



Centrum voor Wiskunde en Informatica

REPORT*RAPPORT*

MAS

Modelling, Analysis and Simulation



Modelling, Analysis and Simulation

Towards an h -adaptive immersed boundary method for
the unsteady incompressible Navier-Stokes equations

J. Naber, B. Koren

REPORT MAS-E0705 FEBRUARY 2007

Centrum voor Wiskunde en Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO). CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Simulation (MAS)

Information Systems (INS)

Copyright © 2007, Stichting Centrum voor Wiskunde en Informatica
P.O. Box 94079, 1090 GB Amsterdam (NL)
Kruislaan 413, 1098 SJ Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

Towards an h -adaptive immersed boundary method for the unsteady incompressible Navier-Stokes equations

ABSTRACT

This paper treats the development and validation of a numerical method for the solution of the unsteady incompressible Navier-Stokes equations in two dimensions. The method of choice is a Godunov-type pressure-free correction method similar to the method developed by Lee and LeVeque. Validation is done by considering a two-dimensional boundary-layer flow along a flat-plate. The resulting method provides a starting point for the development of an h -adaptive immersed boundary method to be applied to the flapping flexible filament problem proposed by Zhu and Peskin.

2000 Mathematics Subject Classification: 65M55, 65M60, 76D05

Keywords and Phrases: unsteady incompressible Navier-Stokes equations; pressure-correction method; boundary-layer flow

Towards an h -Adaptive Immersed Boundary Method for the Unsteady Incompressible Navier-Stokes Equations*

J. Naber[†] and B. Koren[‡]

Abstract

This paper treats the development and validation of a numerical method for the solution of the unsteady incompressible Navier-Stokes equations in two dimensions. The method of choice is a Godunov-type pressure-free correction method similar to the method developed by Lee and LeVeque. Validation is done by considering a two-dimensional boundary-layer flow along a flat-plate. The resulting method provides a starting point for the development of an h -adaptive immersed boundary method to be applied to the flapping flexible filament problem proposed by Zhu and Peskin.

Keywords: unsteady incompressible Navier-Stokes equations, pressure-correction method, boundary-layer flow.

1 Introduction

Immersed boundary methods (IBMs) – introduced by Peskin and McQueen [13, 14, 15] for the simulation of blood flows through human hearts – have been developed as an answer to the limitations encountered by methods that use body-fitted meshes. In the case of complex-shaped and moving bodies the use of body-fitted meshes is rather tedious. The generation of meshes around bodies with complex geometries requires a lot of effort. In case these complex-shaped bodies also move or deform, the restrictions of body-fitted mesh methods become even larger. After each time step a new body-fitted grid has to be generated, which increases the computation time to a large extent. Immersed boundary methods do not suffer from these limitations. Calculations are performed on fixed cartesian grids. Making these grids is trivial. The body of interest is simply immersed in these grids (see figure 1). The challenge of developing an IBM is to impose the boundary conditions on the immersed boundary such that the flow is aware of the boundary's presence to the required accuracy level. Several approaches have already been developed for this. Some of these have proven themselves on a variety of problems.

A disadvantage of IBMs is the possibly large amount of cells required to capture the immersed boundary sufficiently accurate. Especially in the case of high-Reynolds-number flows, which are characterized by a very thin boundary layer (boundary-layer thickness is in general proportional to the inverse of the square root of the Reynolds number, the amount of grid cells can become extremely large. As opposed to that, body-fitted meshes can efficiently capture these boundary layers. To make IBM solutions as accurate as body-fitted grid solutions, it is of prime importance that these boundary layers are captured with sufficient resolution. Given the simple grids used by IBMs the logical step towards increased boundary-layer resolution is the application of local grid-refinement techniques.

The combination of adaptive mesh refinement and an IBM for the unsteady incompressible Navier-Stokes equations allows for the simulation of various interesting flow problems. Especially interesting is the application of these methods to problems with moving flexible bodies, such as

*This research has been funded by the Dutch NWO under project 613-000-105.

[†]CWI, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands. E-mail: j.naber@cwi.nl

[‡]CWI, and TU Delft, Faculty of Aerospace Engineering, P.O. Box 5058, 2600 GB Delft, The Netherlands.

swimming organisms, flapping sails and many other. A representative flow problem is the flow around a flapping flexible filament [20].

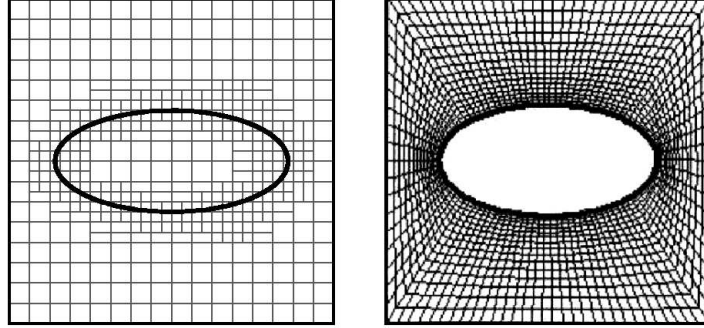


Figure 1: Immersed boundary methods use simple cartesian meshes (*left*) instead of complex body-fitted meshes (*right*). Courtesy: Vander Meulen [12].

The aim of this study is to make a start with the development of an h -adaptive Immersed Boundary Method (IBM) for the simulation of the unsteady incompressible Navier-Stokes equations around moving and flexible bodies. Work has already been performed at CWI on an IBM for steady flows (see Vander Meulen [12]). The step towards unsteady flows is taken here. This paper treats the development of a numerical solver – based on the pressure-projection method – for the unsteady incompressible Navier-Stokes equations.

The pressure-projection method, which was introduced independently by Chorin [3, 4, 5] and Temam [18], is a fractional-step method for the solution of the time-dependent incompressible Navier-Stokes equations. It solves the Navier-Stokes equations using an intermediate velocity, which is calculated without use of the continuity equation, i.e., without the divergence constraint. This intermediate velocity field is then projected onto a space of divergence-free vector fields, which requires the solution of a Poisson equation for the pressure. This projection step is necessary because the incompressible Navier-Stokes equations contain no time derivative for the pressure. Solely solving for the momentum equations under influence of the divergence constraint does not take into account the change of pressure.

Since its introduction the pressure-projection method has seen many different forms. In the excellent overview paper by Brown, Cortez and Minion [2] the most frequently used projection methods are compared and fitted into a general formulation. Here we will adapt this general formulation but restrict ourselves to the pressure-free projection method by Kim and Moin [9], the pressure-correction method by Van Kan [8], by Bell, Colella and Glaz [1], and a method by Lee and LeVeque [11] that combines elements from both former methods.

Elements of all three methods will be combined. An important criterion in selecting these elements is the ease of extending the resulting numerical solver with immersed boundary and mesh-refinement techniques. For the validation a standard two-dimensional test case is considered, a boundary-layer flow along a flat plate. Numerical results are compared to the exact boundary-layer solution by Blasius, and to numerical results obtained using Femlab. The results are convincing and are a good motivation for the extension to an h -adaptive immersed boundary method for the unsteady incompressible Navier-Stokes equations.

In section 2 the Navier-Stokes equations are introduced, followed by a discussion of the general formulation of the class of pressure-projection methods in section 3. Of the large class of available solution methods for the unsteady incompressible Navier-Stokes equations, three commonly used methods are treated in more detail in section 4. By combining elements of these methods, a method of choice is constructed in section 5, which is validated in section 6. The method of choice is chosen such that it is particularly suitable for the development of an immersed boundary method. Conclusions and suggestions for future work are given in sections 7 and 8.

2 The Navier-Stokes equations

The unsteady incompressible Navier-Stokes equations for viscous fluid flow read:

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p = \nu \Delta \mathbf{u} + \mathbf{F}, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

with $\mathbf{u} = (u, v)^T$ the velocity field, p the pressure, ν the kinematic viscosity ($\nu \sim 1/Re$) and \mathbf{F} the external forces. Boundary conditions are only imposed to the velocity \mathbf{u} . For convenience we only consider the Dirichlet case:

$$\mathbf{u} = \mathbf{b} \quad \text{on} \quad \partial\Omega. \quad (3)$$

3 General formulation of projection methods

The pressure-projection method aims at solving (1) under the influence of the divergence constraint (2) and boundary conditions (3). Since this can not be done in one step, a fractional-step approach is used. First an analog of (1) is solved for an intermediate velocity field \mathbf{u}^* without regard of the divergence constraint. Next this velocity field is projected onto a divergence-free vector field resulting in \mathbf{u}^{n+1} , with $n + 1$ the new time level. This projection step requires the solution of a Poisson equation for the pressure. The general procedure for this projection approach, adapting the notation by Brown, Cortez and Minion [2], is the following.

Transport step: Given the velocity \mathbf{u}^n solve for the intermediate velocity field \mathbf{u}^* (with $t^* = t^n + \Delta t$) using the following semi-discrete form of (1):

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^{n+1/2} \cdot \nabla) \mathbf{u}^{n+1/2} + \nabla q = \frac{\nu}{2} \Delta (\mathbf{u}^* + \mathbf{u}^n) + \mathbf{F}, \quad (4)$$

with boundary condition:

$$B(\mathbf{u}^*) = 0 \quad \text{on} \quad \partial\Omega. \quad (5)$$

Equation (4) is a Crank-Nicholson-type discretization of (1), in which the pressure $p^{n+1/2}$ has been replaced by some approximation q . The boundary condition (5) still has to be specified depending on the method used.

Projection step: Advance the intermediate velocity \mathbf{u}^* to the final time level $t^{n+1} = t^* + \Delta t$ using:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}, \quad (6)$$

subject to the divergence constraint:

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \quad (7)$$

Boundary conditions for this projection step depend again on the type of projection method used. For now it can be said that these have to be consistent with $B(\mathbf{u}^*) = 0$ and $\mathbf{u}^{n+1} = \mathbf{b}^{n+1}$ on $\partial\Omega$. From (6) it is obvious that this projection step requires ϕ^{n+1} in order to calculate the new velocity field. Applying the discrete divergence operator to (6) and using the continuity equation (7) leads to a Poisson equation for ϕ :

$$\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \quad (8)$$

Pressure update: Substitution of (6) into (4) and elimination of \mathbf{u}^* gives the following expression for the pressure at the new time level:

$$p^{n+1/2} = q + \phi^{n+1} - \frac{\nu \Delta t}{2} \Delta \phi^{n+1}. \quad (9)$$

After this pressure update one can repeat all steps to march in time.

From the above it is clear that by choosing q and $B(\mathbf{u}^*)$ a large class of projection methods can be constructed. We will strive for a pressure-projection method that is formally second-order accurate in both time and space. Several frequently used methods that satisfy this requirement will be discussed in more detail below. But first a short comment will be made on the spatial discretization of the projection method.

The Crank-Nicholson-type discretization used above results in a second-order accurate discretization in time. For the projection method to be formally second-order in space too, one has to choose a corresponding spatial discretization. The most frequently used approach is the one by Harlow and Welch [6], which uses a staggered grid. This allows for the use of second-order central differences only. A staggered grid (see figure 2) defines the pressures p in the cell centres, the u velocities on the centres of the vertical cell edges and the v velocities in the centres of the horizontal cell edges. It is not hard to see that when applied to equations (4), (6) and (8), this results in a compact scheme (covering no more than three cells), which is fully second-order accurate.

Unfortunately, a staggered grid approach can lead to difficulties near the boundaries. Interpolation is required when a velocity cell does not exactly coincide with the boundary. It is less suited for higher-order discretizations and – last but not least – does not allow for an easy implementation of h -adaptivity and immersed boundary techniques. The more familiar collocated grid approach does not suffer from these restrictions. However, application of second-order central differences on such a collocated grid results in the so-called odd-even decoupling or checkerboard pattern. A second-order central difference scheme decouples the velocity and pressure fields and allows for oscillations. Solutions to this problem have been found but generally result in less elegant methods. See appendix A for a more detailed discussion of the staggered and collocated approach.

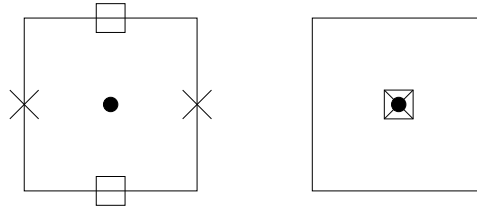


Figure 2: Pressure-projection methods often use staggered grids (*left*) instead of the more familiar collocated grids (*right*). Pressures p indicated by \bullet , velocity components u by \times and velocity components v by \square .

4 Several projection methods

Given the general formulation, a large class of projection methods for the unsteady incompressible Navier-Stokes equations can be constructed. A few of the most commonly used methods will be discussed here, after which a method of choice is determined. For a more extensive overview of the methods discussed below see the excellent overview paper by Brown, Cortez and Minion [2].

4.1 Pressure-free projection method

The first method that will be considered has been developed by Kim and Moin [9]. Using the terminology from [2] it can be referred to as the pressure-free projection method. As the name already suggests this projection method is characterized by the choice of $q = 0$, which corresponds to a system of equations that contains no pressure terms. According to [2] this specific choice of q has two consequences: first, it implies that time-accumulating errors in the pressure distribution do not contribute to errors in the momentum equations. Second, Brown, Cortez and Minion state that \mathbf{u}^* is no longer within $\mathcal{O}(\Delta t^2)$ of \mathbf{u}^{n+1} since the projection step implies larger corrections. Finally

the boundary condition (5) becomes nontrivial since it should take into account the absence of the pressure term in the momentum equations.

Following the work by Kim and Moin the abovementioned problems have been solved with a smart choice for the boundary condition (5). The resulting method can be summarized as follows (using again the general formulation of projection methods):

Transport step: Advance \mathbf{u}^n to \mathbf{u}^* (with $t^* = t^n + \Delta t$) using:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^{n+1/2} \cdot \nabla) \mathbf{u}^{n+1/2} = \frac{\nu}{2} \Delta (\mathbf{u}^* + \mathbf{u}^n) + \mathbf{F}. \quad (10)$$

The corresponding boundary condition can now be obtained in a straightforward manner by using a Taylor-series expansion of $\mathbf{u}^*(\mathbf{x}, t^*)$ around the point t^n (see [9]), resulting in:

$$\mathbf{u}^* = \mathbf{b} + \Delta t \nabla \phi^n \quad \text{on} \quad \partial\Omega. \quad (11)$$

Note that an extra term is added to the original boundary condition $\mathbf{u}^* = \mathbf{b}$ in order to assure second-order accuracy in time. In fact, one actually wants to use $\mathbf{u}^* = \mathbf{b} + \Delta t \nabla \phi^{n+1}$ but ϕ^{n+1} is not known until the projection step. It can be shown that there is no reduction in order when using this lagged potential. In [2] the boundary condition (11) is only used in tangential direction, while in normal direction $\mathbf{u}^* = \mathbf{b}$ is used.

Projection step: Advance the intermediate velocity \mathbf{u}^* to \mathbf{u}^{n+1} (with $t^{n+1} = t^* + \Delta t$) by solving the Poisson equation for ϕ^{n+1} :

$$\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (12)$$

with boundary condition:

$$\mathbf{n} \cdot \nabla \phi^{n+1} = 0 \quad \text{on} \quad \partial\Omega, \quad (13)$$

and setting:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}. \quad (14)$$

Here \mathbf{n} is the unit normal vector. Note that the original boundary condition for (14) subject to the divergence constraint $\nabla \cdot \mathbf{u}^{n+1} = 0$ reads $\mathbf{n} \cdot \mathbf{u}^{n+1} = \mathbf{n} \cdot \mathbf{b}$ on $\partial\Omega$. Taking the normal component of (14) and applying the divergence constraint and original boundary conditions leads to the boundary conditions given in (13).

Pressure update: Although this method is called a pressure-free projection method, it is possible to obtain an expression for the pressure as a function of the potential ϕ . Setting $q = 0$ in (9) gives for this post-processing step:

$$p^{n+1/2} = \phi^{n+1} - \frac{\nu \Delta t}{2} \Delta \phi^{n+1}. \quad (15)$$

This expression assures that also the pressure is updated with second-order accuracy.

This concludes the discussion of the pressure-free projection method. In [9] the governing equations are solved using second-order central differences (finite-differences) on a staggered grid. Solving the semi-discrete advection-diffusion equation (10) is done using standard techniques. Kim and Moin used an Adams-Bashfort scheme to approximate the nonlinear convection terms, i.e., $(\mathbf{u}^{n+1/2} \cdot \nabla) \mathbf{u}^{n+1/2} \sim \frac{3}{2}(\mathbf{u}^n \cdot \nabla) \mathbf{u}^n - \frac{1}{2}(\mathbf{u}^{n-1} \cdot \nabla) \mathbf{u}^{n-1}$. Crank-Nicholson for the diffusion terms removes the severe numerical viscous stability restriction. A fast solver is used for the Poisson equation.

4.2 Pressure-correction method

The second method that is considered is the one by Van Kan [8] and a slightly adapted version by Bell, Colella and Glaz [1]. This projection method, which is often referred to as the pressure-correction method, is characterized by the choice of $q = p^{n-1/2}$ and $B(\mathbf{u}^*) = \mathbf{u}^* - \mathbf{b} = 0$ on $\partial\Omega$. As such the projection step consists of solving the Poisson equation with boundary condition $\mathbf{n} \cdot \nabla \phi^{n+1} = 0$ on $\partial\Omega$. According to [2] this implies that \mathbf{u}^* differs at most $\mathcal{O}(\Delta t^2)$ from \mathbf{u}^{n+1} such that the velocity boundary condition for \mathbf{u}^* is justified. The resulting velocity field converges at a second-order rate (in the maximum norm). In the original method the pressure converged at a first-order rate only, but a slight improvement discussed below improves this to second-order. The pressure-correction method can be summarized as follows:

Transport step: Advance \mathbf{u}^n to \mathbf{u}^* (with $t^* = t^n + \Delta t$) using:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + (\mathbf{u}^{n+1/2} \cdot \nabla) \mathbf{u}^{n+1/2} + \nabla p^{n-1/2} = \frac{\nu}{2} \Delta (\mathbf{u}^* + \mathbf{u}^n) + \mathbf{F}, \quad (16)$$

with boundary condition:

$$\mathbf{u}^* = \mathbf{b} \quad \text{on} \quad \partial\Omega. \quad (17)$$

Projection step: Advance the intermediate velocity \mathbf{u}^* to \mathbf{u}^{n+1} (with $t^{n+1} = t^* + \Delta t$) by solving the Poisson equation for ϕ^{n+1} :

$$\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (18)$$

with boundary condition:

$$\mathbf{n} \cdot \nabla \phi^{n+1} = 0 \quad \text{on} \quad \partial\Omega, \quad (19)$$

and setting:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}. \quad (20)$$

Again \mathbf{n} is the unit normal vector. Substitution of (20) into (16) gives:

$$\nabla \phi^{n+1} = \nabla p^{n+1/2} - \nabla p^{n-1/2}. \quad (21)$$

This implies that $\mathbf{u}^{n+1} \approx \mathbf{u}^* - \Delta t^2 \frac{\partial \Delta p}{\partial t}$. And thus $\mathbf{u}^* = \mathbf{b} + \mathcal{O}(\Delta t^2)$ on $\partial\Omega$, which justifies the use of (17).

Pressure update: In the original method by Bell, Colella and Glaz the pressure was updated using the following gradient equation:

$$\nabla p^{n+1/2} = \nabla p^{n-1/2} + \nabla \phi^{n+1}. \quad (22)$$

However, from the boundary condition for the Poisson equation $\mathbf{n} \cdot \nabla \phi^{n+1}$ it follows that $\mathbf{n} \cdot \nabla p^{n+1/2} = \mathbf{n} \cdot \nabla p^{n-1/2}$ on $\partial\Omega$, which is in general not correct on the boundary, resulting in an artificial boundary layer. Simply using the expression in (9) gives the correct update for the pressure:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} - \frac{\nu \Delta t}{2} \Delta \phi^{n+1}. \quad (23)$$

This expression assures that also the pressure is updated with second-order accuracy.

The method proposed by Van Kan uses an Adams-Bashfort discretization of the nonlinear convection terms in the momentum equations, while Bell, Colella and Glaz use a Godunov-type approach. Especially the latter method is suitable for the implementation into a collocated finite-volume setting.

4.3 Godunov-type pressure-free projection method

It has already been mentioned that the two methods described above are only two of many different pressure-projection methods. Before making a decision between the methods available we shall first consider a third method developed by Lee and LeVeque [11]. This Godunov-type projection method combines the pressure-free projection method by Kim and Moin [9] with the Godunov techniques from the pressure-correction method by Bell et al. [1].

The basic difference between this method and the former two is that instead of only a staggered velocity field also a cell-centered velocity field is used. We shall define the cell-centered velocities in cell (i, j) as $\mathbf{U}^n = (U_{ij}^n, V_{ij}^n)^T$ (to be interpreted as cell averages). Similar to the staggered approach a second set of velocities \mathbf{u}^n is defined at the cell edges, i.e., $u_{i-1/2,j}^n$ and $u_{i+1/2,j}^n$ at the left and right cell edge respectively and $v_{i,j-1/2}^n$ and $v_{i,j+1/2}^n$ at the bottom and top cell edges respectively. Lee and LeVeque now assume that at the beginning of each time step the edge velocity field is discrete divergence-free: $\nabla_h \cdot \mathbf{u}^n = 0$. Here the discrete divergence operator is defined as:

$$(\nabla_h \cdot \mathbf{u}^n)_{ij} = \frac{u_{i+1/2,j}^n - u_{i-1/2,j}^n}{\Delta x} + \frac{v_{i,j+1/2}^n - v_{i,j-1/2}^n}{\Delta y}. \quad (24)$$

With this their algorithm is then defined as:

Transport step: Advance \mathbf{U}^n to \mathbf{U}^* (with $t^* = t^n + \Delta t$) by solving the following pressure-free advection-diffusion equation:

$$\mathbf{U}_t + (\mathbf{u}^n \cdot \nabla) \mathbf{U} = \nu \Delta \mathbf{U} + \mathbf{F}, \quad (25)$$

with boundary condition:

$$\mathbf{U}^* = \mathbf{b} + \Delta t \nabla \phi^n \quad \text{on} \quad \partial \Omega. \quad (26)$$

Note that the nonlinear convection terms are linearized by introducing the edge velocities at the old time level \mathbf{u}^n . Instead of solving the advection-diffusion equation (25) in one step, Lee and LeVeque choose a fractional step approach. This procedure allows them to solve the advective part using their CLAWPACK higher-order finite-volume code separate from the diffusive part. The procedure they follow is based on the so-called Godunov splitting method, which for the semi-discrete advection-diffusion equation consists of two steps. First one solves the advection part, followed by an update of the diffusion part:

$$\frac{\mathbf{U}^\dagger - \mathbf{U}^n}{\Delta t} + (\mathbf{u}^n \cdot \nabla) \mathbf{U}^\dagger = 0, \quad (27)$$

$$\frac{\mathbf{U}^* - \mathbf{U}^\dagger}{\Delta t} = \frac{\nu}{2} \Delta (\mathbf{U}^* + \mathbf{U}^\dagger) + \mathbf{F}. \quad (28)$$

Unfortunately, this approach is only first-order accurate in time. A formally second-order accurate splitting method is the Strang splitting approach:

$$\frac{\mathbf{U}^\dagger - \mathbf{U}^n}{\Delta t/2} = \frac{\nu}{2} \Delta (\mathbf{U}^\dagger + \mathbf{U}^n), \quad (29)$$

$$\frac{\mathbf{U}^\ddagger - \mathbf{U}^\dagger}{\Delta t} + (\mathbf{u}^n \cdot \nabla) \mathbf{U}^\ddagger = 0, \quad (30)$$

$$\frac{\mathbf{U}^* - \mathbf{U}^\ddagger}{\Delta t/2} = \frac{\nu}{2} \Delta (\mathbf{U}^* + \mathbf{U}^\ddagger) + \mathbf{F}. \quad (31)$$

It is not difficult to notice that the first and last step of this Strang splitting method can be combined into one step over Δt instead of over two steps of $\Delta t/2$. Only at the first and last time step one has to use the halved time step then. Further note that since an implicit method is used for the diffusion equation, no viscous stability restriction is present.

Conversion step: Since two velocity fields are used in this projection method, and the transport step only calculates intermediate cell-centered velocities, a conversion step is required to obtain intermediate cell-edge velocities. Lee and LeVeque simply average the cell-centered velocities \mathbf{U}^* :

$$u_{i-1/2,j}^* = \frac{1}{2}(\mathbf{U}_{i-1,j}^* + \mathbf{U}_{i,j}^*), \quad (32)$$

$$v_{i,j-1/2}^* = \frac{1}{2}(\mathbf{V}_{i,j-1}^* + \mathbf{V}_{i,j}^*). \quad (33)$$

Although this seems to be a rather coarse procedure, it can be shown that this is indeed consistent with the manner in which the staggered approaches work. Further it can be noted that the intermediate edge velocity field is in general not divergence-free, i.e., $\nabla_h \cdot \mathbf{u}^* \neq 0$.

Projection step 1: Advance the intermediate velocity \mathbf{u}^* to \mathbf{u}^{n+1} (with $t^{n+1} = t^* + \Delta t$) by solving the Poisson equation for ϕ^{n+1} :

$$\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (34)$$

with boundary condition:

$$\mathbf{n} \cdot \nabla \phi^{n+1} = 0 \quad \text{on} \quad \partial\Omega. \quad (35)$$

Here \mathbf{n} is the unit normal vector. As usual the velocities at the final time level can now be obtained using the expression:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \phi^{n+1}, \quad (36)$$

where the gradient of ϕ^{n+1} is discretized using second-order central differences over the cell edges. Note the use of the ∇_h symbol for this compact discrete gradient operator. For the left cell edge the gradient is then discretized by $(\frac{\partial \phi}{\partial x})_{i-1/2,j}^{n+1} = \frac{\phi_{i,j}^{n+1} - \phi_{i-1,j}^{n+1}}{\Delta x}$. Similar expressions hold for the other edges. This projection step results in an update of the edge velocities only. Since a new loop requires the cell-centered velocities at the new time level a separate projection step has to be taken for the cell-centered velocities.

Projection step 2: Advance the intermediate cell-centered velocity \mathbf{U}^* to \mathbf{U}^{n+1} (with $t^{n+1} = t^* + \Delta t$) using:

$$\mathbf{U}^{n+1} = \mathbf{U}^* - \Delta t \nabla_{2h} \phi^{n+1}. \quad (37)$$

The difference between this update and the one in (36) is in the discrete gradient operator ∇_{2h} .

In (37) this gradient is taken over a broader stencil: $(\frac{\partial \phi}{\partial x})_{ij}^{n+1} = \frac{\phi_{i+1,j}^{n+1} - \phi_{i-1,j}^{n+1}}{2\Delta x}$. For the method given above Lee and LeVeque state that the edge velocities calculated with (36) are exactly divergence-free. The cell-centered velocities are therefore divergence-free up to second order, which is as desired.

Pressure update: If required, the pressure at the new time level can be obtained using the following expression:

$$\nabla p^{n+1/2} = \phi^{n+1} - \frac{\nu \Delta t}{2} \Delta \phi^{n+1}. \quad (38)$$

This expression assures that also the pressure is updated with second-order accuracy.

5 The method of choice

A method is constructed which uses elements from the methods discussed above. This method will have to be suited for extension to an h -adaptive immersed boundary method. For this we follow the recommendations by Vander Meulen [12]. In this work it was concluded that of all available IBMs,

the so-called ghost-cell method is most promising for the application to unsteady high-Reynolds-number flows around moving and flexible bodies. The ghost-cell method is an immersed boundary method based on a finite-volume discretization. It introduces the presence of the immersed boundary by locally adapting the numerical fluxes using various interpolation techniques. For this purpose virtual or 'ghost' cells with an interpolated flow state are introduced.

The ghost-cell method can be easily implemented into existing finite-volume solvers, as shown by Vander Meulen [12]. A disadvantage of the ghost-cell approach is its use of interpolation techniques, which can become rather tedious in case of higher dimensions combined with higher orders of accuracy.

The ghost-cell method is based on a finite-volume discretization with numerical fluxes. Although most incompressible Navier-Stokes solvers are based on a staggered finite-difference approach, several versions have also been developed that are of the finite-volume or Godunov-type. Of these, the previously discussed method by Lee and LeVeque [11] has shown its competence and has already proven itself on immersed boundary problems (although an immersed boundary method other than the ghost-cell approach was used). Especially the well-known odd-even decoupling problem has been solved in an elegant and successful manner in this method. With the method by Lee and LeVeque as an example a novel Godunov-type pressure-free projection method is developed. It will be explained in more detail in the following.

5.1 A novel Godunov-type pressure-free projection method

It has been explained that the basic idea of the method by Lee and LeVeque is that besides a staggered velocity field also a cell-centered velocity field is used, for which the cell-centered velocities in cell (i, j) at time level n are defined as $\mathbf{U}^n = (U_{ij}^n, V_{ij}^n)^T$. A second set of staggered velocities \mathbf{u}^n is defined at the cell edges, i.e., $u_{i-1/2,j}^n$ and $u_{i+1/2,j}^n$ at the left and right cell edge respectively and $v_{i,j-1/2}^n$ and $v_{i,j+1/2}^n$ at the bottom and top cell edges, respectively. Assuming that at the beginning of each time step the edge velocity field is discrete divergence-free: $\nabla_h \cdot \mathbf{u}^n = 0$, the method of choice reads:

Transport step: Advance \mathbf{U}^n to \mathbf{U}^* (with $t^* = t^n + \Delta t$) by solving the following pressure-free advection-diffusion equation:

$$\mathbf{U}_t + (\mathbf{u}^n \cdot \nabla) \mathbf{U} = \nu \Delta \mathbf{U} + \mathbf{F}, \quad (39)$$

with boundary condition:

$$\mathbf{U}^* = \mathbf{b} + \Delta t \nabla \phi^n \quad \text{on} \quad \partial \Omega. \quad (40)$$

This linearized advection-diffusion equation is discretized using a finite-volume method based on numerical fluxes, i.e., a Godunov approach. For now body forces are neglected, i.e., $\mathbf{F} = 0$. Our two-dimensional numerical domain is discretized using rectangular cells $\Omega_{i,j}$ having sides $m = 1, 2, 3, 4$ with length l_m being either Δx or Δy , and area $A_{i,j}$. Integration of (39) over a cell $\Omega_{i,j}$ and introducing $\bar{\mathbf{U}}_{i,j} = \frac{1}{A_{i,j}} \int_{\Omega_{i,j}} \mathbf{U} d\Omega$ as the cell mean, results in the following semi-discrete (discretized in space, not in time) momentum equation:

$$\frac{d\bar{\mathbf{U}}_{i,j}}{dt} = -\frac{1}{A_{i,j}} \sum_{m=1}^4 (\mathbf{F}_{i,j} n_x + \mathbf{G}_{i,j} n_y)_m l_{i,j,m}, \quad (41)$$

where n_x and n_y are the outward normal vectors on the cell faces. The functions $\mathbf{F}_{i,j}$ and $\mathbf{G}_{i,j}$ are known as the numerical flux functions and contain the advective and diffusive flux terms:

$$\mathbf{F} = \begin{pmatrix} u^n U - \nu \frac{\partial U}{\partial x} \\ u^n V - \nu \frac{\partial V}{\partial x} \end{pmatrix} \quad \text{and} \quad \mathbf{G} = \begin{pmatrix} v^n U - \nu \frac{\partial U}{\partial y} \\ v^n V - \nu \frac{\partial V}{\partial y} \end{pmatrix}. \quad (42)$$

Since the cell-centered velocity field $\mathbf{U}^n = (U_{ij}^n, V_{ij}^n)^T$ is not defined at the cell faces (the solution is assumed to be piecewise continuous), these numerical fluxes can not be calculated in

a straightforward manner. To decide which velocity should be used for the flux calculation an upwinding technique is applied. For higher-order methods this upwinding is combined with a slope limiter that stabilizes the numerical solution. Here we shall not go that far since only a first-order method is constructed, but the extension to a higher-order scheme is a straightforward procedure for which a large variety of limiters is available. For now the first-order upwind scheme satisfies. It chooses between one of the two velocities in the cells adjacent to the cell face under consideration, i.e., for cell-face m one can use either \mathbf{U}_m^+ or \mathbf{U}_m^- , with the $+$ indicating the cell 'outside' (in outward normal direction) the cell face and the $-$ the cell 'inside'. The velocity is chosen as follows:

$$\mathbf{U}_m = \begin{cases} \mathbf{U}_m^+ & \text{if } \mathbf{n} \cdot \mathbf{u}_m^n \geq 0, \\ \mathbf{U}_m^- & \text{if } \mathbf{n} \cdot \mathbf{u}_m^n < 0. \end{cases} \quad (43)$$

The gradients in (42) are calculated using central differences. Note that since here only a first-order accurate method is considered, the boundary condition (40) reduces to $\mathbf{U}^* = \mathbf{b}$. For higher orders the original expression (40) has to be used.

What results is the time discretization of equation (41). For this the first-order accurate forward Euler scheme is used. Extension to higher orders can for example be done by using an explicit Runge-Kutta scheme. Applying the forward Euler scheme results in the following expression for the velocity $\bar{\mathbf{U}}_{i,j}$ at the new time level $n + 1$:

$$\bar{\mathbf{U}}_{i,j}^{n+1} = \bar{\mathbf{U}}_{i,j}^n - \frac{\Delta t}{A_{i,j}} \sum_{m=1}^4 (\mathbf{F}_{i,j} n_x + \mathbf{G}_{i,j} n_y)_m l_{i,j,m}. \quad (44)$$

Following the work by Hindmarsh et al. [7], two time-step restrictions hold for the resulting scheme; both an advective and a diffusive restriction. They read:

$$\frac{|\mathbf{U}| \Delta t_{\text{adv}}}{\Delta x} < 1 \quad \text{and} \quad \frac{2\nu \Delta t_{\text{diff}}}{(\Delta x)^2} < 1. \quad (45)$$

For the method to be stable the time step has to satisfy both restrictions. In practice this means that the restriction that requires the smallest time step is the one that is used.

Conversion step: The conversion from cell-centered to cell-edge velocities is simply done by averaging:

$$u_{i-1/2,j}^* = \frac{1}{2}(\mathbf{U}_{i-1,j}^* + \mathbf{U}_{i,j}^*), \quad (46)$$

$$v_{i,j-1/2}^* = \frac{1}{2}(\mathbf{V}_{i,j-1}^* + \mathbf{V}_{i,j}^*). \quad (47)$$

Although this seems to be a rather coarse procedure, it can be shown that this is indeed consistent with the manner in which the staggered approaches work. It can further be noted that the intermediate edge velocity field is in general not divergence-free, i.e., $\nabla_h \cdot \mathbf{u}^* \neq 0$.

Projection step 1: The intermediate velocity \mathbf{u}^* is advanced to \mathbf{u}^{n+1} (with $t^{n+1} = t^* + \Delta t$) by solving the Poisson equation for ϕ^{n+1} :

$$\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*, \quad (48)$$

with boundary condition:

$$\mathbf{n} \cdot \nabla \phi^{n+1} = 0 \quad \text{on} \quad \partial \Omega. \quad (49)$$

Here \mathbf{n} is the unit normal vector. For solving the Poisson equation for the pressure-like function ϕ a large variety of Poisson solvers – amongst which several standardized solution packages – are available. Since this solution step needs to be taken within every time step, it is required that the Poisson solver is sufficiently fast. Multigrid methods have proven themselves

as fast Poisson solvers. For the method of choice a solver based on an algebraic multigrid method (AMG), developed by Stüben [17] is chosen. An excellent overview of this and other multigrid methods can be found in [19].

It is well known that the Poisson equation (48) with Neumann boundary conditions (49) does not have a unique solution – any constant value can be added to the solution – making it unable to be solved with the AMG solver by Stüben [17]. The system can be made solvable by removing the constant mode from the solution. This can be done by introducing a Dirichlet boundary condition for the pressure. Often an outflow boundary is used for this purpose.

Note that although this projection method is called pressure-free, since it does not need a physical pressure to solve for the velocity field, the constant mode problem of the Poisson equation requires that a pressure or value for ϕ is prescribed. Using a prescribed ‘real’ pressure for this is the most common approach. So in fact this pressure-free projection method is not as pressure-free as the name suggests.

Given the new value ϕ^{n+1} the velocities at the final time level can be obtained using the expression:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla_h \phi^{n+1}, \quad (50)$$

where the gradient of ϕ^{n+1} is discretized using second-order central differences over the cell edges. Note the use of the ∇_h symbol for this compact discrete gradient operator. For the left cell edge the gradient is then discretized by $(\frac{\partial \phi}{\partial x})_{i-1/2,j}^{n+1} = \frac{\phi_{i,j}^{n+1} - \phi_{i-1,j}^{n+1}}{\Delta x}$. Similar expressions hold for the other edges. This projection step results in an update of the edge velocities only. Since a new loop requires the cell-centered velocities at the new time level a separate projection step has to be taken for the cell-centered velocities.

Projection step 2: The intermediate cell-centered velocity \mathbf{U}^* is advanced to \mathbf{U}^{n+1} (with $t^{n+1} = t^* + \Delta t$) using:

$$\mathbf{U}^{n+1} = \mathbf{U}^* - \Delta t \nabla_{2h} \phi^{n+1}. \quad (51)$$

Pressure update: If required, the pressure at the new time level can be obtained using the following expression:

$$\nabla p^{n+1/2} = \phi^{n+1} - \frac{\nu \Delta t}{2} \Delta \phi^{n+1}. \quad (52)$$

This expression assures that also the pressure is updated with second-order accuracy if necessary. For the first-order method developed here this expression can be reduced to $\nabla p^{n+1/2} = \phi^{n+1}$.

6 Numerical results

For the validation of the numerical method developed here, the two-dimensional flow around a half-infinite flat-plate is considered. This test problem is taken from the work by Koren [10] and of specific interest because of the availability of a reference solution, being the Blasius boundary-layer solution. Besides this exact Blasius solution also a numerical solution, obtained with the commercial finite-element package COMSOL, is used. The Blasius solution provides a reference solution in the boundary layer, while the COMSOL solution allows for comparison of the numerical solution in the entire domain. The solution of this flat-plate problem is steady. It is found by time-stepping to steady state, using the unsteady solver developed here. It should be noted that this is in no way the most efficient way of obtaining steady solutions.

The geometry and boundary conditions for this numerical test case are given in figure 3. Uniform inflow boundary conditions are prescribed on the left boundary. On the top and right boundaries only the pressure is prescribed, resulting in outflow-type boundary conditions. On the flat plate, which starts at $x = 0.5$, to avoid singularities near the left inflow boundary, no-slip conditions hold. On the boundary part left of the flat-plate symmetry-boundary conditions are prescribed.

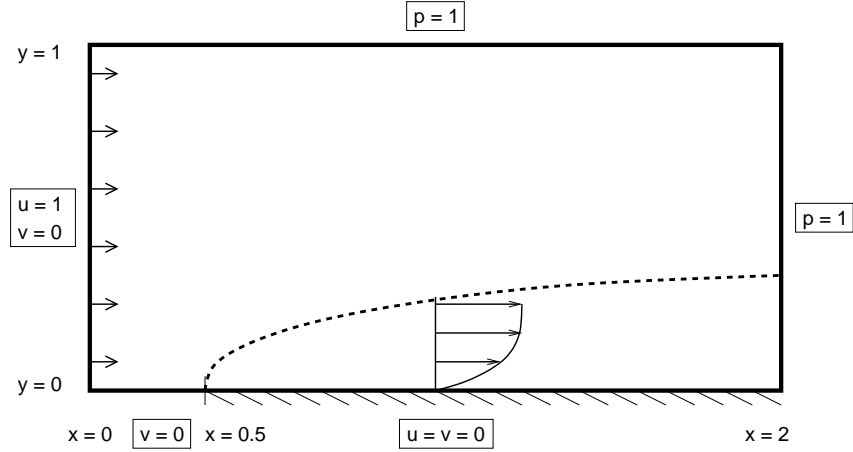


Figure 3: Geometry and boundary conditions for the boundary-layer flow along a half-infinite flat plate.

Two studies have been performed to validate the numerical method; a grid study and a Reynolds-number study. All results are compared to the Blasius solution taken from Schlichting [16] and numerical solutions obtained with COMSOL. For the COMSOL solution a third-order finite-element method on an unstructured grid consisting of triangular cells is used.

6.1 Grid study

In the grid study several numerical solutions on different grids are obtained. The Reynolds number is taken constant as $Re = 150$ (based on the plate length $L = 1.5$ and inflow velocity $U = 1$). Velocity profiles on the three grids with cell sizes $h = 1/8$, $h = 1/16$ and $h = 1/32$ are given in figure 4.

Our results comply well with both reference solutions (Blasius and COMSOL). Even on the coarsest grid a reasonable boundary-layer velocity-profile is obtained. The finest solution is almost identical to the (higher-order) solution obtained with the COMSOL solver on an approximately equally fine grid.

6.2 Reynolds-number study

For the Reynolds-number study a constant cell size of $h = 1/32$ is chosen. Results for different Reynolds numbers are shown in figure 5. With increasing Reynolds number, the boundary layer clearly shrinks. A four times larger Reynolds number, results in a boundary-layer thickness which is two times smaller. This is in agreement with theory.

With decreasing boundary-layer thickness, the amount of grid cells present in the boundary layer becomes smaller. This makes that the numerical solution in the high-Reynolds-number case becomes coarser and thus less accurate.

In figures 6, 7 and 8 contour plots of the velocity and pressure fields at different Reynolds numbers are given. Clearly visible is the stagnation point at the front of the flat plate. Note again the boundary layer getting two times smaller with a four times larger Reynolds number.

6.3 Further observations

It has been noticed that the solution becomes unstable when a relatively low time step is taken. Numerical experiments with $\Delta t \sim 10^{-6}$ result in unstable solutions that blow up in time. The origin

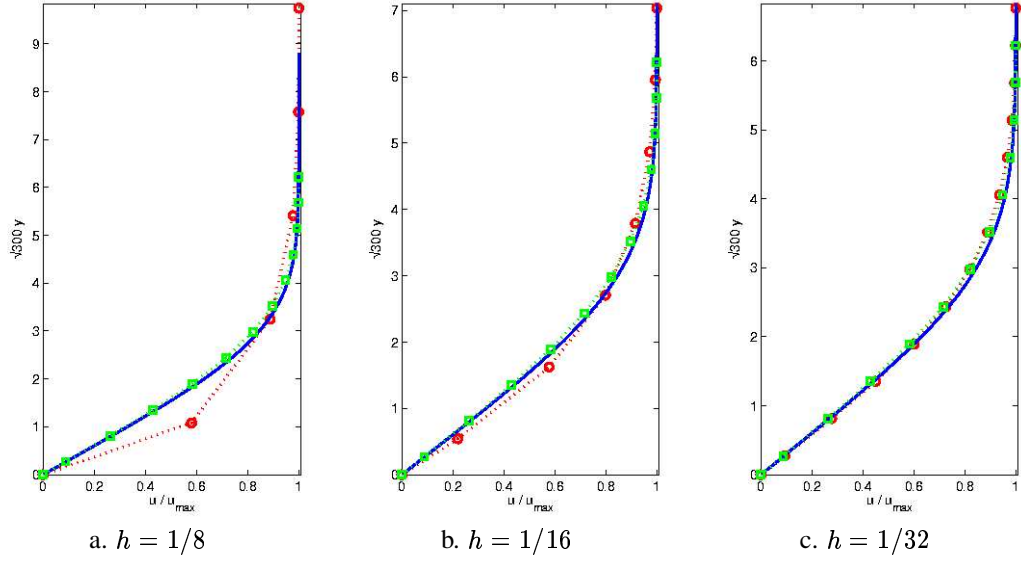


Figure 4: Boundary-layer velocity-profiles at $x = 1$ for the flat-plate flow: grid study (\circ : present numerical solution, \square : COMSOL solution, — : Blasius solution).

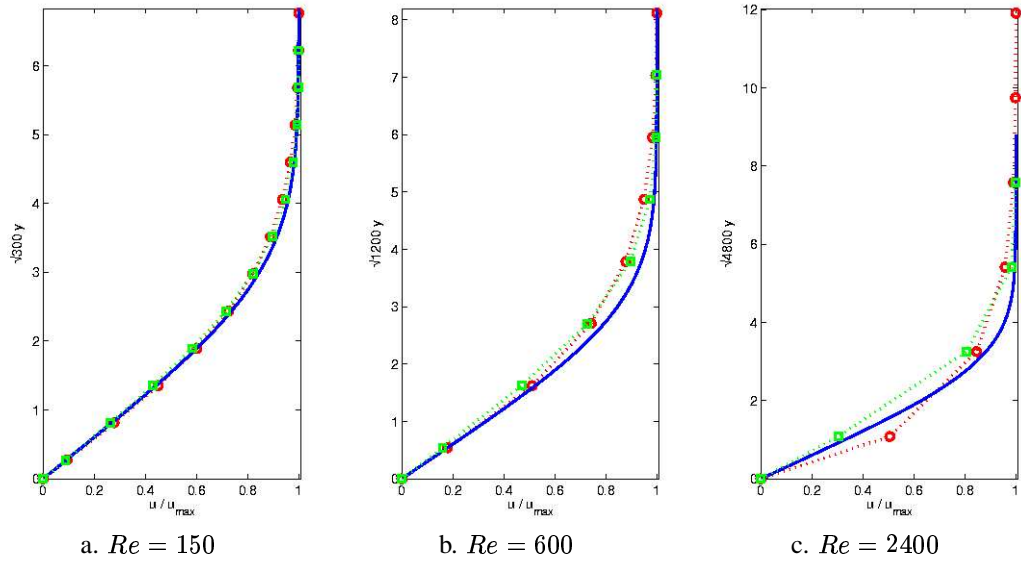


Figure 5: Boundary-layer velocity-profiles at $x = 1$ for the flat-plate flow: Reynolds-number study (\circ : present numerical solution, \square : COMSOL solution, — : Blasius solution).

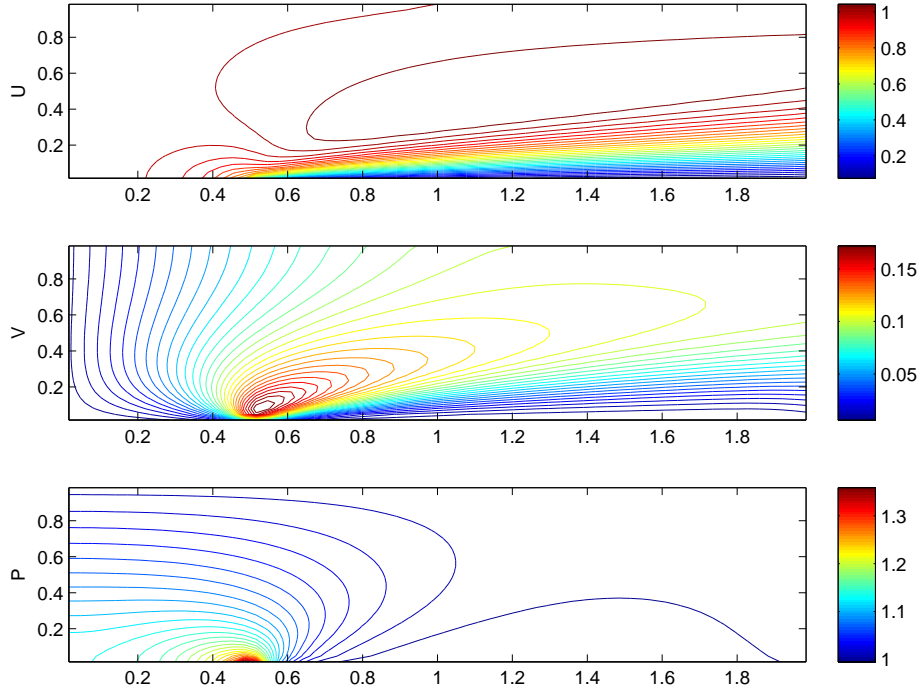


Figure 6: Contour plots at steady state for the fat-plate fbw: $Re = 150$ (*top*: u -component of velocity, *middle*: v -component of velocity, *bottom*: pressure).

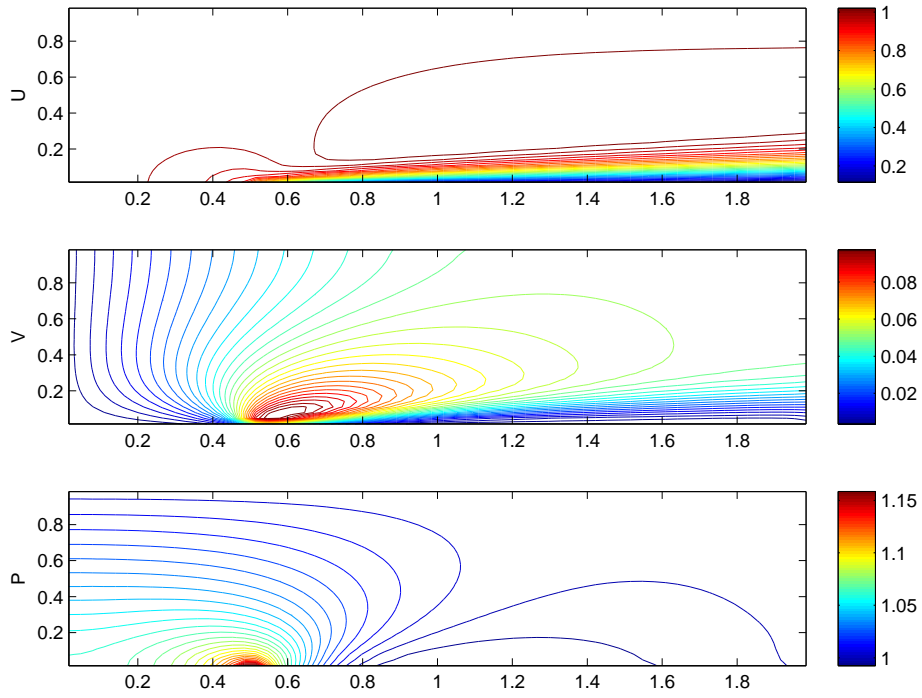


Figure 7: Contour plots at steady state for the fat-plate fbw: $Re = 600$ (*top*: u -component of velocity, *middle*: v -component of velocity, *bottom*: pressure).

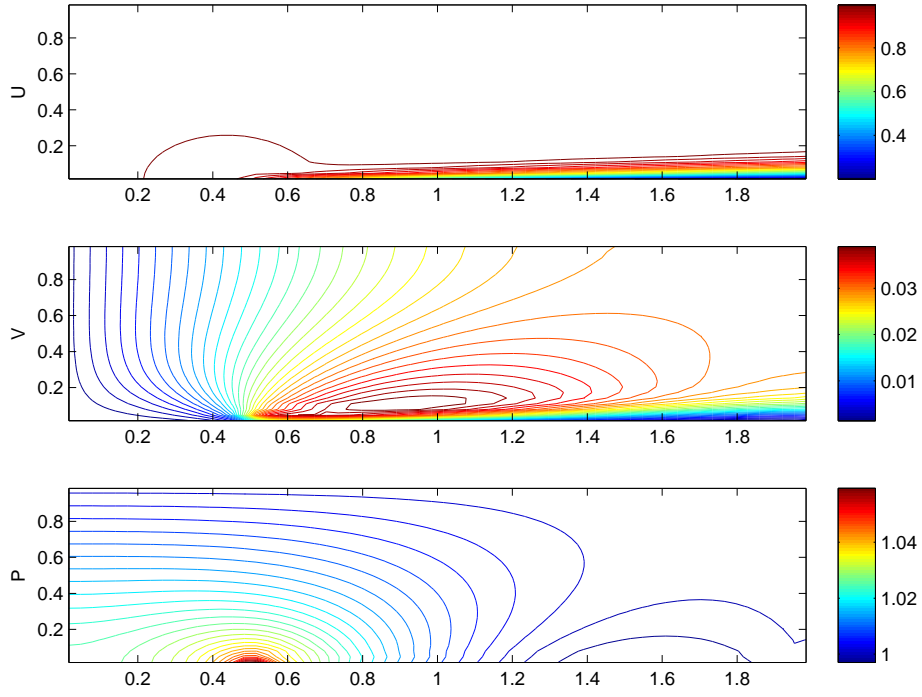


Figure 8: Contour plots at steady state for the flat-plate flow: $Re = 2400$ (*top*: u -component of velocity, *middle*: v -component of velocity, *bottom*: pressure).

of this instability is not yet clear. The unstable behavior seems to resemble that of Lax-Wendroff-type schemes.

Further it should be stated that besides the pressure-free projection method also a version has been developed similar to the pressure-correction method by Van Kan. This method, which makes use of a momentum equation containing a pressure gradient, was unable to produce stable solutions when used in the Godunov-format used for the pressure-free method. This behavior can be fully subscribed to the odd-even decoupling. Since the pressure-free method produces satisfying results there is no immediate need to further develop the pressure-correction variant of this method.

7 Conclusions

The aim of this report was the development of a novel Godunov-type pressure-free projection method able to solve two-dimensional unsteady incompressible flows described by the Navier-Stokes equations. This solution method acts as the starting point for further research towards the development of an h -adaptive immersed boundary method suitable for flexible and moving bodies. The resulting Navier-Stokes solver is based on the Godunov-type pressure-free projection method by Lee and LeVeque [11]. It makes use of the fast and robust AMG solver by Stüben [17], which makes that the total solution time is reasonably short. The Godunov-type approach can be extended relatively easy to higher orders of accuracy. The method is validated for the flow around a half-infinite flat plate. Numerical results comply well with expectations, thereby validating the numerical method.

8 Future work

The aim of future research is threefold: *(i)* extension to higher orders of accuracy, *(ii)* implementation of immersed boundary techniques, and *(iii)* adaptive grid-refinement methods.

The present method is only first-order accurate in time and space. The possibility for extension to higher orders of accuracy has already been incorporated in the method proposed in this report. Realization of a method with a second or higher order of accuracy is within reach.

Extension of the present method with immersed boundary techniques allows the solution of problems involving moving and flexible bodies. Work by Vander Meulen [12] has set a clear path towards such an immersed boundary method. The ghost-cell method is most promising for application to unsteady high-Reynolds-number flows. Extension of the current Navier-Stokes solver with IBM techniques requires the adaption of the flux functions. An additional problem that needs to be solved for is the occurrence of freshly cleared cells; cells that were inside the body at one stage, but are in the fluid at another. Several solution techniques are available for this.

One of the major drawbacks of IBMs is that they have difficulties in capturing sharp boundary layers since the grid cells are – opposed to those in body-fitted grids – in general not aligned with the boundary layer. Implementation of h -adaptivity or mesh-refinement techniques are therefore necessary for the application to real-life problems. The data structure used in the present solver is such that extension with local grid-refinement techniques is rather straightforward.

References

- [1] J.B. BELL, P. COLELLA AND H.M. GLAZ, A second-order projection method for the incompressible Navier-Stokes equations, *J. Comp. Phys.* **85**, pages 257–283, 1989.
- [2] D.L. BROWN, R. CORTEZ AND M.L. MINION, Accurate projection methods for the incompressible Navier-Stokes equations, *J. Comp. Phys.* **168**, pages 464–499, 2001.
- [3] A.J. CHORIN, Numerical solution of the Navier-Stokes equations, *Math. Comp.* **22**, pages 742–762, 1968.
- [4] A.J. CHORIN, Numerical solution of incompressible flow problems, *Studies in Numerical Analysis* **2**, pages 64–71, 1968.
- [5] A.J. CHORIN, On the convergence of discrete approximations to the Navier-Stokes equations, *Math. Comp.* **23**, pages 341–353, 1969.
- [6] F.H. HARLOW AND J.E. WELCH, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fl.* **8**, pages 2182–2189, 1965.
- [7] A.C. HINDMARSH, P.M. GRESHO AND D.F. GRIFFITH, The stability of explicit Euler time-integration for certain finite difference approximations of the multi-dimensional advection-diffusion equation, *Int. J. Num. Meth. Fluids* **4**, pages 853–897, 1984.
- [8] J. VAN KAN, A second-order accurate pressure-correction scheme for viscous incompressible flow, *SIAM J. Sci. Comp.* **7**, pages 870–891, 1986.
- [9] J. KIM AND P. MOIN, Application of a fractional-step method to incompressible Navier-Stokes equations, *J. Comp. Phys.* **59**, pages 308–323, 1985.
- [10] B. KOREN, Multigrid and Defect Correction for the Steady Navier-Stokes Equations, Application to Aerodynamics, *CWI Tracts* **74**, CWI, Amsterdam, 1991.
- [11] L. LEE AND R.J. LEVEQUE, An immersed interface method for incompressible Navier-Stokes equations, *SIAM J. Sci. Comp.* **25**, pages 832–856, 2002.
- [12] R.J.R. VANDER MEÛLEN, The immersed boundary method for the (2D) incompressible Navier-Stokes equations, Report MAS-E0607, CWI, Amsterdam, 2006.
<http://ftp.cwi.nl/CWIreports/MAS/MAS-E0607.pdf>
- [13] C.S. PESKIN, Numerical analysis of blood flow in the heart, *J. Comp. Phys.* **25**, pages 220–252, 1977.
- [14] C.S. PESKIN, Lectures on mathematical aspects of physiology, *Lect. Appl. Math.* **19**, AMS, Providence, RI, 1981.
- [15] C.S. PESKIN AND D.M. MCQUEEN, Modeling prosthetic heart valves for the numerical analysis of blood flow in the heart, *J. Comp. Phys.* **105**, pages 113–132, 1980.
- [16] H. SCHLICHTING, *Boundary-Layer Theory*, McGraw-Hill series in mechanical engineering, 1979.
- [17] K. STÜBEN, An introduction to algebraic multigrid, in *Multigrid*, pages 413–532, Academic Press, 2001.
- [18] R. TEMAM, Sur l’approximation de la solution des équations de Navier-Stokes par la méthode des fractionnaires II, *Arc. Rational Mech. Anal.* **33**, pages 377–385, 1969.
- [19] U. TROTTEBERG, C. OOSTERLEE AND A. SCHÜLLER, *Multigrid*, Academic Press, 2001.
- [20] L. ZHU AND C.S. PESKIN, Simulation of a flexible filament in a flowing soap film by the immersed boundary method, *J. Comp. Phys.* **179**, pages 452–468, 2002.

A Staggered versus collocated grids

In this appendix we shall clarify the staggered and collocated grid approach for the unsteady incompressible Navier-Stokes equations. The second-order accurate pressure-correction method by Van Kan [8] is used for the discretization of the governing equations since it allows for a clear explanation. First the staggered approach will be considered. It is standard in pressure-correction methods. It results in a second-order accurate discretization in both space and time. It is followed by a discussion of the same pressure-correction method, but now with a discretization on a collocated grid. Problems connected to this collocated approach are discussed.

A.1 Van Kan's pressure-correction method

The pressure-correction method proposed by Van Kan aims at solving the unsteady incompressible Navier-Stokes equations in two separate steps. The first step solves the momentum equations for an intermediate velocity field, which is then in the correction step projected onto a space of divergence-free vector fields. This second correction or projection step requires the solution of a Poisson equation for the pressure. The explanation of this pressure-correction method starts with the semi-discrete (discretized in time, not in space) Navier-Stokes equations. As such the two steps of the pressure-correction method read:

Transport step: Advance \mathbf{u}^n to the intermediate velocity \mathbf{u}^* (with $t^* = t^n + \Delta t$) using:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{1}{2} [(\mathbf{u}^* \cdot \nabla) \mathbf{u}^* + (\mathbf{u}^n \cdot \nabla) \mathbf{u}^n] + \nabla p^n = \frac{\nu}{2} \Delta (\mathbf{u}^* + \mathbf{u}^n), \quad (53)$$

which is an implicit second-order Crank-Nicholson scheme.

Projection step: Advance \mathbf{u}^* to the final time level $t^{n+1} = t^* + \Delta t$ using:

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t \nabla \phi^{n+1}, \quad (54)$$

subject to the divergence-constraint:

$$\nabla \cdot \mathbf{u}^{n+1} = 0. \quad (55)$$

Here the scalar potential is given by $\phi^{n+1} = p^{n+1} - p^n$. This potential can be computed by taking the discrete divergence of (54) and substituting (55). This leads to a Poisson equation for ϕ :

$$\Delta \phi^{n+1} = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^*. \quad (56)$$

A.2 Staggered difference scheme

The original method by Van Kan – and many other pressure-correction methods – uses the staggered grid by Harlow and Welch [6] for the spatial discretization of the equations given above. Such a staggered grid cell is depicted in figure 9.

For the spatial discretization of equations (53), (54), (55) and (56) several discrete spatial operators are defined. For the x -direction these operators read:

$$\nabla_x^+ \psi_{i,j} = \frac{\psi_{i+1,j} - \psi_{i,j}}{\Delta x}, \quad (57)$$

$$\nabla_x^- \psi_{i,j} = \frac{\psi_{i,j} - \psi_{i-1,j}}{\Delta x}, \quad (58)$$

$$\nabla_{2x} \psi_{i,j} = \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta x}, \quad (59)$$

$$\Delta_x \psi_{i,j} = \frac{\psi_{i+1,j} - 2\psi_{i,j} + \psi_{i-1,j}}{\Delta x^2}. \quad (60)$$

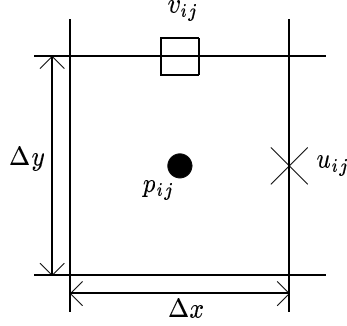


Figure 9: Definition of a staggered grid cell (i, j) . Pressure p indicated by \bullet , velocity components u by \times and velocity components v by \square .

Similar expressions hold for the y -direction. Further we define the following averaged velocities:

$$\bar{u}_{i,j} = \frac{1}{4} (u_{i,j+1} + u_{i,j} + u_{i-1,j} + u_{i-1,j+1}), \quad (61)$$

$$\bar{v}_{i,j} = \frac{1}{4} (v_{i,j} + v_{i,j-1} + v_{i+1,j} + v_{i+1,j-1}), \quad (62)$$

which are necessary since the velocities are only defined at their own (either vertical or horizontal) edges and not on those of the other velocity component.

Using these definitions a discretization of the correction method can be obtained. For the staggered grid this discretization reads:

Transport step: Advance \mathbf{u}^n to the intermediate velocity \mathbf{u}^* (with $t^* = t^n + \Delta t$) using:

$$\begin{aligned} \frac{u_{i,j}^* - u_{i,j}^n}{\Delta t} &+ \frac{1}{2} [u_{i,j}^* \nabla_{2x} u_{i,j}^* + \bar{v}_{i,j}^* \nabla_{2y} u_{i,j}^* \\ &+ u_{i,j}^n \nabla_{2x} u_{i,j}^n + \bar{v}_{i,j}^n \nabla_{2y} u_{i,j}^n] + \nabla_x^+ p^n \\ &= \frac{\nu}{2} \Delta_x (u_{i,j}^* + u_{i,j}^n), \end{aligned} \quad (63)$$

and

$$\begin{aligned} \frac{v_{i,j}^* - v_{i,j}^n}{\Delta t} &+ \frac{1}{2} [\bar{u}_{i,j}^* \nabla_{2x} v_{i,j}^* + v_{i,j}^* \nabla_{2y} v_{i,j}^* \\ &+ \bar{u}_{i,j}^n \nabla_{2x} v_{i,j}^n + v_{i,j}^n \nabla_{2y} v_{i,j}^n] + \nabla_y^+ p^n \\ &= \frac{\nu}{2} \Delta_y (v_{i,j}^* + v_{i,j}^n). \end{aligned} \quad (64)$$

Projection step: Advance \mathbf{u}^* to the final time level $t^{n+1} = t^* + \Delta t$ using:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^*}{\Delta t} = -\nabla_x^+ (\phi_{i,j}^{n+1}), \quad (65)$$

and

$$\frac{v_{i,j}^{n+1} - v_{i,j}^*}{\Delta t} = -\nabla_y^+ (\phi_{i,j}^{n+1}), \quad (66)$$

subject to the divergence constraint:

$$\nabla_x^- \cdot u_{i,j}^{n+1} + \nabla_y^- \cdot v_{i,j}^{n+1} = 0. \quad (67)$$

Note that this divergence constraint is discretized centrally around the pressure point. Taking the discrete divergence ∇^- of (65) and (66) and substituting (67) gives:

$$\frac{1}{\Delta t} [\nabla_x^- u_{i,j}^* + \nabla_y^- v_{i,j}^*] = [\nabla_x^- \nabla_x^+ \phi_{i,j}^{n+1} + \nabla_y^- \nabla_y^+ \phi_{i,j}^{n+1}], \quad (68)$$

which is equivalent to the Poisson equation for ϕ^{n+1} discretized on a compact stencil:

$$\frac{1}{\Delta t} [\nabla_x^- u_{i,j}^* + \nabla_y^- v_{i,j}^*] = [\Delta_x \phi_{i,j}^{n+1} + \Delta_y \phi_{i,j}^{n+1}]. \quad (69)$$

A.3 Collocated difference scheme

On a collocated grid the pressure-correction method by Van Kan will have a slightly different discretization. A collocated grid cell has all its variables (p , u and v) defined in the centre (i, j) . It will become clear that a larger stencil is required for the full method to be second-order accurate in both space and time, and to be stable.

The same discrete spatial operator definitions as used for the staggered grid hold for the collocated grid. It only remains to add one discrete diffusion operator, namely the one on a larger stencil:

$$\Delta_{2x} \psi_{i,j} = \frac{\psi_{i+2,j} - 2\psi_{i,j} + \psi_{i-2,j}}{4\Delta x^2}. \quad (70)$$

Following the same procedure as was done for the staggered approach we get:

Transport step: Advance \mathbf{u}^n to the intermediate velocity \mathbf{u}^* (with $t^* = t^n + \Delta t$) using:

$$\begin{aligned} \frac{u_{i,j}^* - u_{i,j}^n}{\Delta t} &+ \frac{1}{2} [u_{i,j}^* \nabla_{2x} u_{i,j}^* + v_{i,j}^* \nabla_{2y} u_{i,j}^* \\ &+ u_{i,j}^n \nabla_{2x} u_{i,j}^n + \bar{v}_{i,j}^n \nabla_{2y} u_{i,j}^n] + \nabla_{2x} p^n \\ &= \frac{\nu}{2} \Delta_x (u_{i,j}^* + u_{i,j}^n), \end{aligned} \quad (71)$$

and

$$\begin{aligned} \frac{v_{i,j}^* - v_{i,j}^n}{\Delta t} &+ \frac{1}{2} [u_{i,j}^* \nabla_{2x} v_{i,j}^* + v_{i,j}^* \nabla_{2y} v_{i,j}^* \\ &+ u_{i,j}^n \nabla_{2x} v_{i,j}^n + v_{i,j}^n \nabla_{2y} v_{i,j}^n] + \nabla_{2y} p^n \\ &= \frac{\nu}{2} \Delta_y (v_{i,j}^* + v_{i,j}^n), \end{aligned} \quad (72)$$

Projection step: Advance \mathbf{u}^* to the final time level $t^{n+1} = t^* + \Delta t$ using:

$$\frac{u_{i,j}^{n+1} - u_{i,j}^*}{\Delta t} = -\nabla_{2x} (\phi_{i,j}^{n+1}), \quad (73)$$

and

$$\frac{v_{i,j}^{n+1} - v_{i,j}^*}{\Delta t} = -\nabla_{2y} (\phi_{i,j}^{n+1}), \quad (74)$$

subject to the divergence constraint:

$$\nabla_{2x} \cdot u_{i,j}^{n+1} + \nabla_{2y} \cdot v_{i,j}^{n+1} = 0. \quad (75)$$

Note that this divergence constraint is discretized centrally around the pressure point. Taking the discrete divergence ∇^- of (73) and (74) and substituting (75) gives:

$$\frac{1}{\Delta t} [\nabla_{2x} u_{i,j}^* + \nabla_{2y} v_{i,j}^*] = [\nabla_{2x} \nabla_{2x} \phi_{i,j}^{n+1} + \nabla_{2y} \nabla_{2y} \phi_{i,j}^{n+1}], \quad (76)$$

which is equivalent to the Poisson equation for ϕ^{n+1} discretized on the larger stencil:

$$\frac{1}{\Delta t} [\nabla_{2x} u_{i,j}^* + \nabla_{2y} v_{i,j}^*] = [\Delta_{2x} \phi_{i,j}^{n+1} + \Delta_{2y} \phi_{i,j}^{n+1}]. \quad (77)$$

It is especially the Δ_{2x} and Δ_{2y} in this Poisson equation that makes the collocated grid approach unsuitable for the pressure-projection method. This broader stencil makes that the velocity and the pressure are decoupled, allowing for oscillations to occur. This phenomenon is also referred to as odd-even decoupling or the checkerboard pattern.

A possible fix would be to simply take the compact diffusion operators Δ_x and Δ_y . This however results in an effect sometimes referred to as dilation. The compact stencil makes that the discrete divergence of the velocity field will not be exactly equal to zero.