Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 285 (2015) 659–693
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Spurious transients of projection methods in microflow simulations

Fabricio S. Sousa[a,*], Cassio M. Oishi[b], Gustavo C. Buscaglia[a]

[a] *Departamento de Matemática Aplicada e Estatística, Instituto de Ciências Matemáticas e de Computação - ICMC, Universidade de São Paulo - Campus de São Carlos, Caixa Postal 668, 13560-970 São Carlos, SP, Brazil*

[b] *Departamento de Matemática e Computação, Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista "Júlio de Mesquita Filho", 19060-900 Presidente Prudente, SP, Brazil*

Received 23 June 2014; received in revised form 19 November 2014; accepted 24 November 2014
Available online 3 December 2014

Highlights

- Most popular fractional step methods are analyzed for highly viscous flows.
- Most incremental methods develop a spurious transient, independent of mesh size.
- Non-incremental methods do not develop spurious transients, but require small time steps.
- Time step bounds are proposed as to avoid artifacts produced by spurious transients.

Abstract

The temporal behavior of projection methods for viscous incompressible low-Reynolds-number flows is addressed. The methods considered result from algebraically splitting the linear system corresponding to each time step, in such a way that the computation of velocity is segregated from that of pressure. Each method is characterized by two (possibly equal) approximate inverses (B_1 and B_2) of the momentum-equation velocity matrix, plus a parameter γ which renders the method non-incremental (if $\gamma = 0$) or incremental (if $\gamma = 1$). The classical first-order projection method, together with more sophisticated methods (Perot's second-order method, Yosida method, pseudo-exact factorization method) and their incremental variants are put into the same algebraic form and their accuracy numerically tested. Splitting errors of first, second and third order in the time step size δt are obtained, depending on the method. The methods are then discussed in terms of their ability and efficiency to compute steady states. Non-incremental methods are impractical because extremely small time steps are required for the steady state, which depends on δt, to be reasonably accurate. Incremental methods, on the other hand, either become unstable as δt is increased or develop a remarkable *spurious transient* which may last an extremely long time (much longer than any physical time scale involved). These transients have serious practical consequences on the simulation of steady (or slowly varying), low-inertia flows. From the physical viewpoint, the spurious transients may interfere with true slow processes of the system, such as heat transfer or species transport, without showing any obvious symptoms (wiggly behavior in space or time, for example, do not occur). From the computational viewpoint, the limitation in time step imposed by the spurious transient phenomenon weighs against choosing projection schemes for microflow applications, despite the low cost of each time step.

Keywords: Projection method; Algebraic splitting; Pressure segregation; Navier–Stokes equations; Incompressible flow; Microfluidics

* Corresponding author.
E-mail addresses: fsimeoni@icmc.usp.br (F.S. Sousa), oishi@fct.unesp.br (C.M. Oishi), gustavo.buscaglia@icmc.usp.br (G.C. Buscaglia).

http://dx.doi.org/10.1016/j.cma.2014.11.039
0045-7825/

1. Introduction

All along the history of the numerical simulation of incompressible flows, much attention has been devoted to *pressure-segregation* methods, which are methods that avoid the solution of linear systems in which the velocity unknowns are coupled to the pressure unknowns. Many pressure-segregation methods have been developed since the early work of Harlow and Welch [1]. We consider here a class of pressure-segregation methods known as *projection methods* [2–7], *pressure-correction methods* [8,9] or *fractional step methods* [10,11]. They have in common that the update of velocity unknowns is obtained from the momentum equation, while the update of pressure unknowns results from solving a Poisson-like equation.

Let us recall the incompressible Navier–Stokes equations, which can be written as

$$\partial_t \mathbf{u} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nu \nabla^2 \mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } [0, T] \times \Omega, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } [0, T] \times \Omega, \tag{2}$$

where t is time, $\mathbf{u}$ is the velocity vector field, p is the pressure, ν is the kinematic viscosity of the fluid, $\mathbf{f}$ is a body-force, all defined in a domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) for $t \in [0, T]$.

Formally, the above equations are equivalent to

$$\partial_t \mathbf{u} = \Pi_Z \left(\mathbf{f} - \nabla \cdot (\mathbf{u}\mathbf{u}) + \nu \nabla^2 \mathbf{u} \right) \tag{3}$$

where Π_Z is the projection operator onto the subspace Z of divergence-free vector fields in Ω. Notice that p does not appear in (3). The early works of A. Chorin [3] and R. Temam [4] exploited the projection structure of the problem to define *projection methods* which successfully accomplish pressure segregation. The underlying mathematics is rooted on the Helmholtz–Hodge theorem (see [12]) about the decomposition of a vector field into gradient and solenoidal components, which allows for the computation of the projection operator by solving a *Poisson equation* for the pressure.

Over the years, many authors [5,13–17] studied the effects of different update schemes for the pressure to obtain higher-order (in time) methods. The treatment of boundary conditions also focused research efforts, as they impact on the time accuracy [15,18,19,7,20–22].

An interesting viewpoint of projection methods is that of *approximate factorization*, or *algebraic splitting* [11,23–25]. In this approach, the decoupling is performed on the spatially and temporally discretized equations by replacing the system matrix of the monolithic method (also called all-at-once method) by an approximate factorization of it. This approach leaves the treatment of the boundary conditions implicit in the approximate matrix factors, much simplifying the analysis. Interesting applications of the approximate factorization approach can be found in the studies by Quarteroni et al. [26], Badia and Codina [27], Lee et al. [20], Chang et al. [28], Griffith [29], among others.

From the cited references it becomes clear that most of the existing projection methods are in fact equivalent to some algebraic splitting method. We have thus chosen to focus this article on algebraic splitting methods, considering them as representative of most projection-like methods.

Section 2 of this article contains an overview of several algebraic splitting methods for viscous incompressible flows, similar to that performed in [25]. The methods are described within a uniform algebraic setting and numerically tested in the benchmark problem of decaying vortices in a periodic domain at $Re = 10^{-2}$. A detailed study of the convergence of the time discretization is conducted, which serves both as verification of the implementation and as direct comparison of the different variants in a low-inertia flow. A similar study involving monolithic and segregated methods for incompressible fluid flows was presented by Elman [30], though in that paper the goal was to investigate the performance of preconditioning strategies (see also [31]).

The convenience of segregating the velocity from the pressure unknowns without sacrificing temporal accuracy is quite significant, as in a mesh with N cells in 3D one solves four $N \times N$ matrices instead of one large matrix of dimensions $4N \times 4N$. Unfortunately, there exist serious time-step restrictions on the applicability of projection methods to low-inertia incompressible flows.

Section 3 discusses these restrictions, which arise from either accuracy or stability considerations. In particular, it is shown that incremental projection methods suffer from severe spurious transients in the computation of pressure-driven viscous-dominated flows. These spurious transients may easily last thousands of time steps. They may become larger than the physical time scale of the process being simulated and completely pollute the numerical results.

An explanation of these spurious transients is provided both by an algebraic argument and by analyzing an analogous thin-film problem. Then the various implemented methods are numerically assessed in what regards the spurious-transient phenomenon. The assessment reveals the importance of the phenomenon, of which CFD practitioners should be aware at the time of selecting simulation methods and parameters (time step in particular) for microfluidic applications. The issue is further discussed, drawing practical conclusions, in Section 4.

2. Overview of discretizations

Our objective here is to describe pressure-segregation techniques for the numerical approximation of (1)–(2). These techniques are obtained by algebraically splitting the monolithic discretization, which in matrix form reads

$$\begin{bmatrix} \mathsf{A} & \mathsf{G} \\ \mathsf{D} & 0 \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix}, \tag{4}$$

where

$$\mathsf{A} = \frac{1}{\delta t}\mathsf{I_u} - \theta\, \nu\, \mathsf{L} \tag{5}$$

$$r^n = \mathsf{N}U^n - \mathsf{T}(U^n) + b^n \tag{6}$$

and

$$\mathsf{N} = \frac{1}{\delta t}\mathsf{I_u} + (1-\theta)\, \nu\, \mathsf{L}. \tag{7}$$

Above, $U^n \in \mathbb{R}^{N+M}$ is the vector of velocity unknowns, assumed to be N for x-velocity components and M for y-velocity components, and $P^{n+\theta} \in \mathbb{R}^Q$ is the vector of pressure unknowns. Although the time location of the algebraic unknown P is irrelevant (P being a Lagrange multiplier), we kept the superscript for consistency with the employed time discretization, as in [25]. In fact, for a fixed spatial discretization, as $\delta t \to 0$, the sequence of vectors P converges to a smooth time-dependent vector $P_{\text{ref}}(t)$, which will be further discussed later on (smooth forcings are of course assumed). Taking $\theta = \frac{1}{2}$ (and assuming negligible inertia for simplicity), the unknown $P^{n+\frac{1}{2}}$ that is computed from (4) converges to $P_{\text{ref}}((n+\frac{1}{2})\delta t)$, as $\delta t \to 0$, with order $\mathcal{O}(\delta t^2)$. It converges with lower order $\mathcal{O}(\delta t)$ to either $P_{\text{ref}}(n\,\delta t)$ or $P_{\text{ref}}((n+1)\delta t)$, thus justifying the preference of the notation $P^{n+\frac{1}{2}}$ over P^n or P^{n+1}.

Notice that the presentation assumes the problem to be 2D for simplicity. The parameter θ is positive and not greater than one. On the right side of (6), $\mathsf{T}(U^n)$ represents the discrete operator of the nonlinear terms. Since we focus on low Reynolds numbers in this work, this term is explicitly approximated (and sometimes neglected altogether).

Note that $\mathsf{A} \in \mathbb{R}^{(M+N)\times(M+N)}$, while $\mathsf{I_u}$ is the identity matrix. If finite elements were used instead of finite differences, one would have $\mathsf{A} = \frac{1}{\delta t}\mathsf{M} - \theta\nu\mathsf{L}$, where M stands for the mass matrix.

Moreover, $\mathsf{L} \in \mathbb{R}^{(M+N)\times(M+N)}$ is the sub-matrix corresponding to the discretization of the viscous operator, $\mathsf{G} \in \mathbb{R}^{(M+N)\times Q}$ corresponds to the discrete gradient operator and $\mathsf{D} \in \mathbb{R}^{Q\times(M+N)}$ is the discrete divergence operator. The vector $r^n \in \mathbb{R}^{(M+N)}$ in (6) contains all the quantities known at the current time level n, and additional information regarding the discretization of forces and boundary conditions, assumed contained in vector $b^n \in \mathbb{R}^{(M+N)}$.

In all the numerical examples the spatial discretization is performed with a Marker-and-Cell scheme [1], in which velocities (in x- and y-direction) are stored at the cell edges (positions $(i+\frac{1}{2}, j)$ and $(i, j+\frac{1}{2})$, respectively) while the pressure is stored at cell centers (positions (i, j)). This scheme is known to be div-stable, in the sense that spatial convergence is attained without the appearance of spurious pressure modes. It is widely used and has been extensively documented in the literature.

2.1. Algebraic splitting

A numerical approach that solves system (4) is said to be *coupled* or *monolithic*. The large associated computational cost motivates the algebraic splitting of the system into subsystems of smaller size. Typically, matrix A is symmetric

and positive, since $-\mathsf{L}$ (upon enforcing the boundary conditions) and I_u are as well. The point of departure of algebraic splitting methods is an exact factorization of the matrix

$$\mathsf{H} = \begin{bmatrix} \mathsf{A} & \mathsf{G} \\ \mathsf{D} & 0 \end{bmatrix} \tag{8}$$

into two, three or four factors [20]. Considering the standard block LU factorization (see [23]) of H, we have

$$\mathsf{H} = \begin{bmatrix} \mathsf{A} & 0 \\ \mathsf{D} & -\mathsf{D}\mathsf{A}^{-1}\mathsf{G} \end{bmatrix} \begin{bmatrix} \mathsf{I}_\mathsf{u} & \mathsf{A}^{-1}\mathsf{G} \\ 0 & \mathsf{I}_\mathsf{p} \end{bmatrix}, \tag{9}$$

where $\mathsf{I}_\mathsf{p} \in \mathbb{R}^{Q\times Q}$ is the identity matrix corresponding to the pressure unknowns. Thus the system (4) can be rewritten as

$$\begin{bmatrix} \mathsf{A} & 0 \\ \mathsf{D} & -\mathsf{D}\mathsf{A}^{-1}\mathsf{G} \end{bmatrix} \begin{bmatrix} \mathsf{I}_\mathsf{u} & \mathsf{A}^{-1}\mathsf{G} \\ 0 & \mathsf{I}_\mathsf{p} \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix}. \tag{10}$$

Note that the system (10) can be solved in two stages:

$$\begin{bmatrix} \mathsf{A} & 0 \\ \mathsf{D} & -\mathsf{D}\mathsf{A}^{-1}\mathsf{G} \end{bmatrix} \begin{bmatrix} \widetilde{U} \\ \widetilde{P} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix}, \tag{11}$$

and

$$\begin{bmatrix} \mathsf{I}_\mathsf{u} & \mathsf{A}^{-1}\mathsf{G} \\ 0 & \mathsf{I}_\mathsf{p} \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} \widetilde{U} \\ \widetilde{P} \end{bmatrix}, \tag{12}$$

where the vectors $\widetilde{U}$ and $\widetilde{P}$ are often regarded as intermediate velocity and pressure, respectively.

Systems (11) and (12) constitute an exact algebraic splitting method for the numerical solution of the Navier–Stokes equations. Unfortunately, it involves the matrix A^{-1} which is unaffordable to compute.

Algebraic splitting methods are approximations of the exact splitting above. These methods define matrices B_1 and B_2 which will approximate A^{-1} in its first and second occurrences, respectively. The numerical method is then defined by

$$\begin{bmatrix} \mathsf{A} & 0 \\ \mathsf{D} & -\mathsf{D}\mathsf{B}_1\mathsf{G} \end{bmatrix} \begin{bmatrix} \widetilde{U} \\ \widetilde{P} \end{bmatrix} = \begin{bmatrix} r^n - \gamma \mathsf{G} P^{n+\theta-1} \\ 0 \end{bmatrix}, \tag{13}$$

and

$$\begin{bmatrix} \mathsf{I}_\mathsf{u} & \mathsf{B}_2\mathsf{G} \\ 0 & \mathsf{I}_\mathsf{p} \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} \widetilde{U} + \gamma\, \mathsf{B}_2\mathsf{G}\, P^{n+\theta-1} \\ \widetilde{P} + \gamma\, P^{n+\theta-1} \end{bmatrix}, \tag{14}$$

where $\gamma \in \{0, 1\}$ is a parameter introduced to make $\widetilde{P}$ "incremental". If $\gamma = 0$ then $\widetilde{P} = P^{n+\theta}$ and the method is said to be "non-incremental". If $\gamma = 1$ then $\widetilde{P} = P^{n+\theta} - P^{n+\theta-1}$ and the method is said to be "incremental".

Notice that, abusing notation, we kept in (13)–(14) the same letters for the velocity and pressure unknowns as that used for the discrete solution of (11)–(12), which is nothing but the solution of the monolithic system (4). From now on the solution of the coupled system will be denoted by U^n_{coupled} and $P^{n+\theta}_{\text{coupled}}$. The sequence of velocity and pressure solutions generated by the different algebraic splitting schemes will only coincide with the monolithic solution if $\mathsf{B}_1 = \mathsf{B}_2 = \mathsf{A}^{-1}$ (interestingly, the value of γ is irrelevant because it cancels out). In all other cases there appears a *splitting error*, both in velocity (i.e.; $U^n - U^n_{\text{coupled}}$) and in pressure (i.e.; $P^{n+\theta} - P^{n+\theta}_{\text{coupled}}$).

Remark 1. Applying the discrete divergence D to the first equation of (14), one can deduce that

$$\mathsf{D}U^{n+1} = \mathsf{D}(\widetilde{U} - \mathsf{B}_2\mathsf{G}\widetilde{P})$$

and combine with the second equation in (13) to get

$$\mathsf{D}U^{n+1} = \mathsf{D}\,(\mathsf{B}_1 - \mathsf{B}_2)\,\mathsf{G}\widetilde{P}$$

which is exactly zero if both approximations to the inverse of A are the same ($\mathsf{B}_1 = \mathsf{B}_2$). This makes the scheme to exactly enforce the incompressibility constraint after each time step. Whenever $\mathsf{B}_1 = \mathsf{B}_2$ the approximate inverse will be denoted by B.

The system of algebraic equations to be solved at each time step for the algebraic splitting methods considered here can thus be summarized as

$$\mathsf{A}\widetilde{U} = r^n - \gamma \mathsf{G} P^{n+\theta-1}, \tag{15}$$

$$\mathsf{D}\mathsf{B}_1\mathsf{G}\left(P^{n+\theta} - \gamma P^{n+\theta-1}\right) = \mathsf{D}\widetilde{U}, \tag{16}$$

$$U^{n+1} = \widetilde{U} - \mathsf{B}_2\mathsf{G}\left(P^{n+\theta} - \gamma P^{n+\theta-1}\right). \tag{17}$$

It is worth looking at the evolution matrix of system (15)–(17) when inertia and forcing terms are neglected, i.e.; when

$$r^n = \mathsf{N}U^n. \tag{18}$$

Eliminating $\widetilde{U}$ one arrives at the equivalent system

$$\begin{bmatrix} \mathsf{A} & \mathsf{A}\mathsf{B}_{2\mathsf{G}} \\ \mathsf{D} & \mathsf{D}(\mathsf{B}_2 - \mathsf{B}_1)\mathsf{G} \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} \mathsf{N}U^n - \gamma(\mathsf{I}_\mathsf{u} - \mathsf{A}\mathsf{B}_2)\mathsf{G}\, P^{n+\theta-1} \\ \gamma \mathsf{D}(\mathsf{B}_2 - \mathsf{B}_1)\mathsf{G}\, P^{n+\theta-1} \end{bmatrix} \tag{19}$$

and inverting the matrix in the left hand side, one finally gets the matrix evolution equation

$$\begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathsf{A} & \mathsf{A}\mathsf{B}_{2\mathsf{G}} \\ \mathsf{D} & \mathsf{D}(\mathsf{B}_2 - \mathsf{B}_1)\mathsf{G} \end{bmatrix}^{-1} \begin{bmatrix} \mathsf{N} & -\gamma(\mathsf{I}_\mathsf{u} - \mathsf{A}\mathsf{B}_2)\mathsf{G} \\ 0 & \gamma \mathsf{D}(\mathsf{B}_2 - \mathsf{B}_1)\mathsf{G} \end{bmatrix}}_{\mathsf{S}} \begin{bmatrix} U^n \\ P^{n+\theta-1} \end{bmatrix} \tag{20}$$

with S given by

$$\mathsf{S} = \begin{bmatrix} \left[\mathsf{I}_\mathsf{u} - \mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{N} & -\gamma\left[\mathsf{I}_\mathsf{u} - \mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{G} \\ (\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{N} & -\gamma\left[(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{G} - \mathsf{I}_\mathsf{p}\right] \end{bmatrix}. \tag{21}$$

The evolution equations of the algebraically split method should be contrasted with those of the monolithic method, which in this (linearized and homogeneous) case reads

$$\begin{bmatrix} \mathsf{A} & \mathsf{G} \\ \mathsf{D} & 0 \end{bmatrix} \begin{bmatrix} U^{n+1}_{\text{coupled}} \\ P^{n+\theta}_{\text{coupled}} \end{bmatrix} = \begin{bmatrix} \mathsf{N}U^n_{\text{coupled}} \\ 0 \end{bmatrix}. \tag{22}$$

Its matrix evolution equation is thus given by

$$\begin{bmatrix} U^{n+1}_{\text{coupled}} \\ P^{n+\theta}_{\text{coupled}} \end{bmatrix} = \underbrace{\begin{bmatrix} \mathsf{A} & \mathsf{G} \\ \mathsf{D} & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathsf{N} & 0 \\ 0 & 0 \end{bmatrix}}_{\mathsf{S}_{\text{coupled}}} \begin{bmatrix} U^n_{\text{coupled}} \\ P^{n+\theta-1}_{\text{coupled}} \end{bmatrix} \tag{23}$$

with

$$\mathsf{S}_{\text{coupled}} = \begin{bmatrix} \left[\mathsf{I}_\mathsf{u} - \mathsf{A}^{-1}\mathsf{G}(\mathsf{D}\mathsf{A}^{-1}\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{N} & 0 \\ (\mathsf{D}\mathsf{A}^{-1}\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{N} & 0 \end{bmatrix}. \tag{24}$$

The splitting error can be estimated by assuming $U^n = U^n_{\text{coupled}}$ and $P^{n+\theta-1} = P^{n+\theta-1}_{\text{coupled}}$ and subtracting (23) from (20), but this is quite involved and will not be detailed here.

In the next sections we recall several algebraic splitting methods of the type discussed above. Analyses of their temporal accuracy can be found in the literature [17,22,25,26,32,33].

2.2. First-order projection scheme

From Eq. (5), one can expand A^{-1} as

$$\mathsf{A}^{-1} = \delta t \mathsf{I_u} + \delta t^2 \theta\, \nu\, \mathsf{L} + \cdots = \delta t \mathsf{I_u} + \sum_{i=1}^{\infty} \delta t^{i+1} (\theta\, \nu\, \mathsf{L})^i\,, \tag{25}$$

a series that is known to be convergent if $\delta t < \|\theta\, \nu\, \mathsf{L}\|_2^{-1}$ (see [34]).

The first-order projection method is defined by the choices $\gamma = 0$ and $\mathsf{B}_1 = \mathsf{B}_2 = \mathsf{B}$ as the first term of the expansion ($\mathsf{B} = \delta t \mathsf{I_u}$), which corresponds to the algebraic system

$$\mathsf{A}\widetilde{U} = r^n \tag{26}$$

$$\delta t \mathsf{D}\mathsf{G} P^{n+\theta} = \mathsf{D}\widetilde{U} \tag{27}$$

$$U^{n+1} = \widetilde{U} - \delta t \mathsf{G} P^{n+\theta} \tag{28}$$

and, eliminating $\widetilde{U}$,

$$\mathsf{A}U^{n+1} + \delta t\, \mathsf{A}\mathsf{G} P^{n+\theta} = \mathsf{N}U^n \tag{29}$$

$$\mathsf{D}U^{n+1} = 0. \tag{30}$$

As its name suggests, this method has splitting error of order $\mathcal{O}(\delta t)$.

2.3. Incremental projection scheme

This method corresponds to taking $\gamma = 1$ and $\mathsf{B} = \delta t\, \mathsf{I_u}$, and is thus the incremental version of the previous method. The equations that define the method are

$$\mathsf{A}\widetilde{U} = r^n - \mathsf{G} P^{n+\theta-1} \tag{31}$$

$$\delta t \mathsf{D}\mathsf{G}\widetilde{P} = \mathsf{D}\widetilde{U} \tag{32}$$

$$U^{n+1} = \widetilde{U} - \delta t \mathsf{G}\widetilde{P} \tag{33}$$

$$P^{n+\theta} = P^{n+\theta-1} + \widetilde{P}. \tag{34}$$

Its splitting error is of order $\mathcal{O}(\delta t^2)$, so that the scheme is of second order in time when $\theta = \frac{1}{2}$ [7,21].

2.4. Perot's second order approximation to A^{-1}

Another way to improve the order of accuracy in time of the first-order projection scheme [24] is to consider one more term in the expansion (25), i.e.,

$$\mathsf{B} = \delta t \mathsf{I_u} + \theta\, \nu\, \delta t^2\, \mathsf{L}. \tag{35}$$

One arrives at the following method (non-incremental version)

$$\mathsf{A}\widetilde{U} = r^n \tag{36}$$

$$\mathsf{D}\left(\delta t \mathsf{I_u} + \theta\, \nu\, \delta t^2\, \mathsf{L}\right) \mathsf{G} P^{n+\theta} = \mathsf{D}\widetilde{U} \tag{37}$$

$$U^{n+1} = \widetilde{U} - \left(\delta t \mathsf{I_u} + \theta\, \nu\, \delta t^2\, \mathsf{L}\right) \mathsf{G} P^{n+\theta}. \tag{38}$$

The splitting error of this method is $\mathcal{O}(\delta t^2)$, so that when the time discretization employed for the coupled problem is second order in time, Perot's method exhibits second order time accuracy [24].

2.5. Yosida method

Yosida method was first introduced by Quarteroni and co-workers [26,35] and can be seen as the algebraic counterpart of the pseudo-compressibility method.

This method consists, in our framework, of taking $\mathsf{B}_1 = \delta t \mathsf{I}_\mathsf{u}$ and $\mathsf{B}_2 = \mathsf{A}^{-1}$. Its non-incremental version ($\gamma = 0$) is then written as

$$\mathsf{A}\widetilde{U} = r^n \tag{39}$$

$$\delta t \mathsf{D}\mathsf{G} P^{n+\theta} = \mathsf{D}\widetilde{U} \tag{40}$$

$$\mathsf{A}U^{n+1} = \mathsf{A}\widetilde{U} - \mathsf{G}P^{n+\theta}, \tag{41}$$

where the last equation was pre-multiplied by A to avoid the direct computation of A^{-1}. As well discussed and analyzed in [26,35], this method is $\mathcal{O}(\delta t^2)$. Interestingly, its incremental version has splitting error of order $\mathcal{O}(\delta t^3)$, as discussed later on.

2.6. Zhang's pseudo-exact factorization

Zhang et al. [36] introduced a pseudo-exact factorization technique which is based on a modified version of the system (4), given by

$$\begin{bmatrix} \mathsf{A} & \mathsf{A}\mathsf{G} \\ \mathsf{D} & 0 \end{bmatrix} \begin{bmatrix} U^{n+1} \\ \Phi^{n+\theta} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix}, \tag{42}$$

where $\Phi^{n+\theta}$ is defined as a "numerical pressure" or "gauge pressure". In this case, systems (4) and (42) are equivalent if

$$\mathsf{A}\mathsf{G}\Phi^{n+\theta} = \mathsf{G}P^{n+\theta}. \tag{43}$$

The matrix in (42) can be exactly decomposed as

$$\begin{bmatrix} \mathsf{A} & \mathsf{A}\mathsf{G} \\ \mathsf{D} & 0 \end{bmatrix} = \begin{bmatrix} \mathsf{A} & 0 \\ \mathsf{D} & -\mathsf{D}\mathsf{G} \end{bmatrix} \begin{bmatrix} \mathsf{I}_\mathsf{u} & \mathsf{G} \\ 0 & \mathsf{I}_\mathsf{p} \end{bmatrix}, \tag{44}$$

resulting in the following scheme

$$\begin{bmatrix} \mathsf{A} & 0 \\ \mathsf{D} & -\mathsf{D}\mathsf{G} \end{bmatrix} \begin{bmatrix} \widetilde{U} \\ \Phi^{n+\theta} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix}, \tag{45}$$

$$\begin{bmatrix} \mathsf{I}_\mathsf{u} & \mathsf{G} \\ 0 & \mathsf{I}_\mathsf{p} \end{bmatrix} \begin{bmatrix} U^{n+1} \\ \Phi^{n+\theta} \end{bmatrix} = \begin{bmatrix} \widetilde{U} \\ \Phi^{n+\theta} \end{bmatrix}. \tag{46}$$

Thus, the resulting algorithm can be written as

$$\mathsf{A}\widetilde{U} = r^n \tag{47}$$

$$\mathsf{D}\mathsf{G}\Phi^{n+\theta} = \mathsf{D}\widetilde{U} \tag{48}$$

$$U^{n+1} = \widetilde{U} - \mathsf{G}\Phi^{n+\theta}. \tag{49}$$

Notice that system (47)–(49) is equivalent to the one obtained for the first order projection scheme Eqs. (26)–(28), if one identifies the gauge pressure $\Phi^{n+\theta}$ as being $\delta t\, P^{n+\theta}$. Zhang et al. [36], however, propose the pressure to be computed (when necessary) from the (discrete) divergence of Eq. (43), resulting in the following Poisson-like equation

$$\mathsf{D}\mathsf{G}P^{n+\theta} = \mathsf{D}\mathsf{A}\mathsf{G}\Phi^{n+\theta}. \tag{50}$$

The splitting error implicit in this method is obtained by writing

$$\Phi^{n+\theta} = (\mathsf{DAG})^{-1}\mathsf{DG}P^{n+\theta}, \tag{51}$$

and substituting into Eqs. (47)–(49) to arrive at the equivalent system

$$\begin{bmatrix} \mathsf{A} & \mathsf{AG}(\mathsf{DAG})^{-1}\mathsf{DG} \\ \mathsf{D} & 0 \end{bmatrix} \begin{bmatrix} U^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} r^n \\ 0 \end{bmatrix}, \tag{52}$$

which can be compared to the original coupled system (4). The splitting error of this method turns out to be first order in time.

By comparing to the generic algebraic splitting evolution equation (19), on the other hand, one further concludes that this pseudo-exact factorization method is a non-incremental scheme that corresponds to choosing

$$\mathsf{B}_1 = \mathsf{B}_2 = \mathsf{G}(\mathsf{DAG})^{-1}\mathsf{D}. \tag{53}$$

An incremental version is readily obtained by taking $\gamma = 1$ in (19) with these choices of B_1 and B_2.

2.7. Exact factorization

In [28], an exact fractional step method for incompressible fluid flows is proposed. The scheme can be interpreted as a discrete stream function method; however the disadvantages of the stream function formulation are eliminated when the change of variables is applied to the already discrete problem.

This method relies on the construction of null spaces for the discrete operators D and G, i.e., the computation of matrices R and C, such that $\mathsf{RG} = 0$ and $\mathsf{DC} = 0$, respectively. Let $V = M + N$ be the total number of velocity unknowns and Q the number of pressure unknowns. The null space matrices will have dimensions $\mathsf{R} \in \mathbb{R}^{(V-Q)\times V}$ and $\mathsf{C} \in \mathbb{R}^{V\times(V-Q)}$.

The coupled system (4) is re-organized as follows

$$\begin{bmatrix} \mathsf{RA} & \mathsf{RG} \\ \mathsf{D} & 0 \end{bmatrix} \begin{bmatrix} \mathsf{C}S^{n+1} \\ P^{n+\theta} \end{bmatrix} = \begin{bmatrix} \mathsf{R}r^n \\ 0 \end{bmatrix}, \tag{54}$$

which is obtained by pre-multiplying the first row by R and defining the variable $S^{n+1} \in \mathbb{R}^{(V-Q)}$ such that $U^{n+1} = \mathsf{C}S^{n+1}$. Since RG is zero, we arrive at the following system of equations, which can be solved sequentially:

$$\mathsf{RAC}S^{n+1} = \mathsf{R}r^n \tag{55}$$

$$U^{n+1} = \mathsf{C}S^{n+1} \tag{56}$$

$$\mathsf{DG}P^{n+\theta} = \mathsf{D}r^n - \mathsf{DA}U^{n+1}. \tag{57}$$

The last equation is obtained by applying the divergence operator to the first row of (4). Note that this method requires the solution of one linear system for velocity, plus an optional Poisson-like problem to evaluate pressure at new time step $t_{n+\theta}$, if needed.

This method is a truly exact factorization scheme: the original system is not modified by any approximation and the splitting error is in fact zero, as illustrated numerically later in this work.

Both null space matrices can be viewed as discrete curl operators: the curl of the gradient is zero ($\mathsf{RG} = 0$) and the divergence of the curl is zero ($\mathsf{DC} = 0$). Therefore S^{n+1} can be viewed as a discrete stream function associated with the velocity field U^{n+1} (the curl of the velocity). Details on how to assemble matrices R and C can be found in [28].

2.8. Convergence tests of algebraic splitting methods

In the previous sections several methods were described, which are listed in Table 1 together with their formal orders of accuracy with $\theta = 0.5$ (second order time discretization). To establish the actual performance of each method, let us test them in the well-known benchmark of freely decaying square vortices [10].

Table 1
Methods used in the numerical tests in this work, and their expected temporal order of accuracy when a second order time discretization is used.

Ref.	Method	Order
M01	Monolithic method	$\mathcal{O}(\delta t^2)$
M02	First order projection method	$\mathcal{O}(\delta t)$
M03	Incremental projection method	$\mathcal{O}(\delta t^2)$
M04	Perot's second order method	$\mathcal{O}(\delta t^2)$
M05	Incremental Perot's second order method	$\mathcal{O}(\delta t^2)$
M06	Yosida method	$\mathcal{O}(\delta t^2)$
M07	Incremental Yosida method	$\mathcal{O}(\delta t^2)$
M08	Pseudo-exact factorization method	$\mathcal{O}(\delta t)$
M09	Incremental Pseudo-exact factorization method	$\mathcal{O}(\delta t^2)$
M10	Exact factorization method	$\mathcal{O}(\delta t^2)$

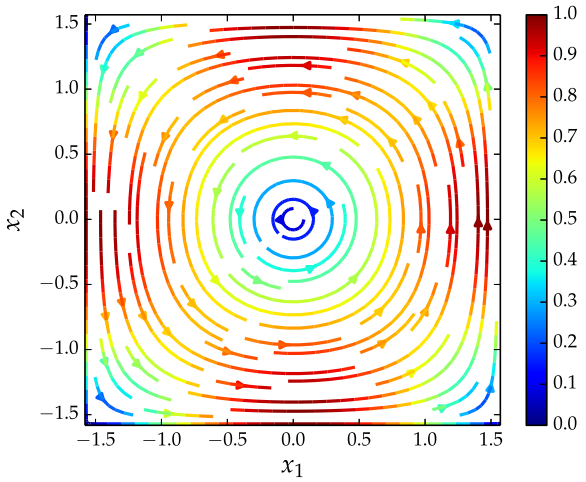


Fig. 1. Streamlines for the freely decaying vortices benchmark at $t = 0$. Colors represent the magnitude of the non-dimensional velocity field. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The domain considered is the square $\Omega = \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \times \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ such that the exact solution of the Navier–Stokes equations is given by

$$\mathbf{u}(x_1, x_2, t) = e^{-2\nu t}\,(-\cos(x_1)\sin(x_2),\ \ \sin(x_1)\cos(x_2)) \tag{58}$$

$$p(x_1, x_2, t) = -0.25\, e^{-4\nu t}\,(\cos(2x_1) + \cos(2x_2))\,. \tag{59}$$

For this problem we choose $\nu = 100$ so that the Reynolds number is small ($Re = 10^{-2}$). The velocity is initialized with the analytical solution computed at time $t = 0$. The boundary conditions used are non-penetrating for normal components ($u_n = 0$) and total slip for tangential components ($\partial_n u_t = 0$), in all sides of the square Ω. Fig. 1 illustrates the flow streamlines at $t = 0$.

In what follows, we fix the spatial discretization at 40×40 cells and focus on the error with respect to the *exact solution of the semi-discrete problem* (discretized in space but not in time). To do this, the reference for comparison, denoted by $(U_{\text{ref}}(t),\, P_{\text{ref}}(t)) \in \mathbb{R}^V \times \mathbb{R}^Q$, is defined as the exact solution of

$$\frac{dU}{dt} + \mathsf{T}(U) - \nu\,\mathsf{L}\,U + \mathsf{G}\,P = 0 \tag{60}$$

$$\mathsf{D}\,U = 0. \tag{61}$$

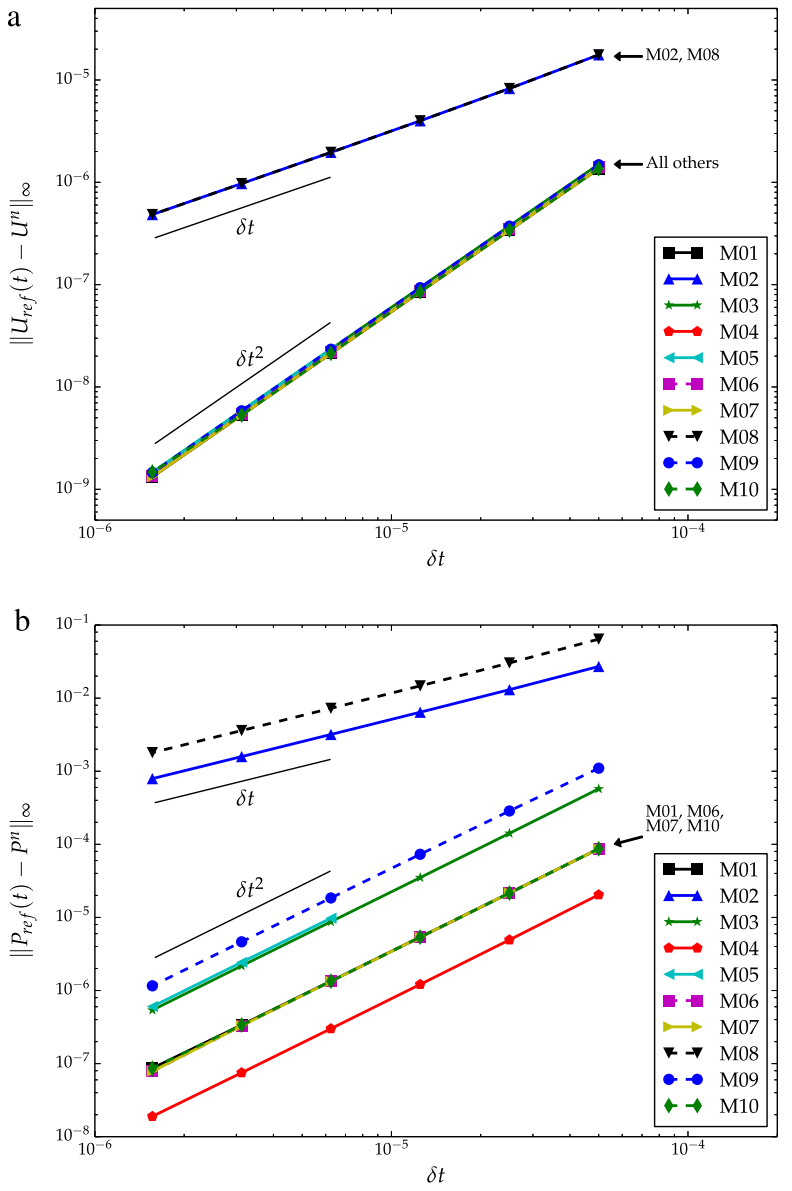


Fig. 2. Temporal convergence analysis for methods M01–M10, compared with a reference solution: (a) Velocity errors $\|U_{\text{ref}}(t) - U^n\|_\infty$. (b) Pressure errors $\|P_{\text{ref}}(t) - P^n\|_\infty$.

Remark 2. By computing errors with respect to $(U_{\text{ref}}(t), P_{\text{ref}}(t))$ and not to $(\mathbf{u}(\cdot, t), p(\cdot, t))$ we explicitly avoid addressing the difficult problem of the convergence of the *fully discrete* solution towards the *exact* solution, of which several aspects remain open and for which the choice of norms is crucial [17].

Since the reference solution $(U_{\text{ref}}(t), P_{\text{ref}}(t))$ is not analytically available, it is computed with the monolithic method (with interpolated initial condition) with a very small time step $\delta t = 10^{-6}$. The problem was computed with all methods listed in Table 1, for time steps $\delta t_k = 2^{-k} \cdot 10^{-3}$, for $k = 1, 2, \ldots, 8$. To ensure second order time accuracy, the Crank–Nicolson scheme was employed ($\theta = 0.5$). When an initial condition for pressure is required by the incremental versions of the tested methods, it is interpolated from the analytical solution. Furthermore, the convective term is linearized by Adams–Bashforth second order extrapolation $\mathsf{T}(U^{n+\frac{1}{2}}) = \frac{3}{2}\mathsf{T}(U^n) - \frac{1}{2}\mathsf{T}(U^{n-1})$.

In Fig. 2 the errors $\|U_{\text{ref}}(t) - U^n\|_\infty$ (in subplot (a)) and $\|P_{\text{ref}}(t) - P^n\|_\infty$ (in subplot (b)) at $t = 10^{-2}$ (with $n\delta t = t$) are plotted as functions of the non-dimensional δt. From this figure, it is clear that all methods attain

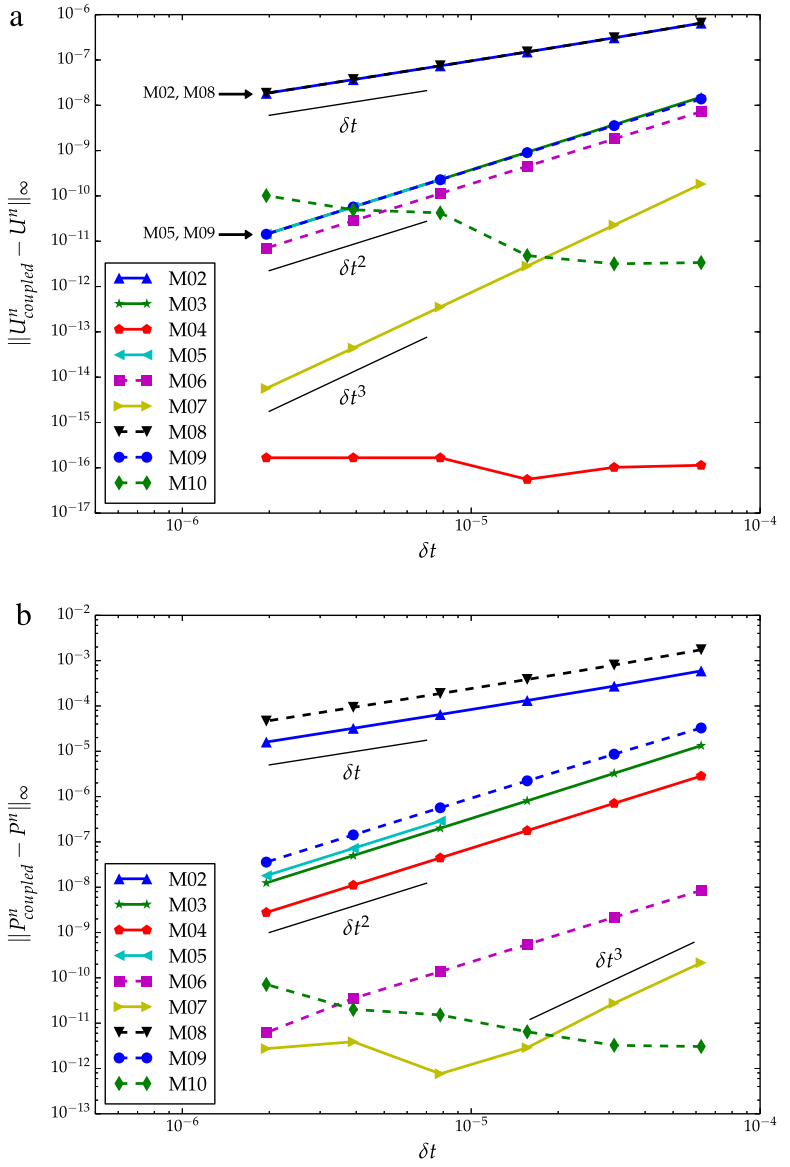


Fig. 3. Convergence analysis for the splitting error of methods M02–M10: (a) Velocity errors $\|U^n_{\text{coupled}} - U^n\|_\infty$. (b) Pressure errors $\|P^n_{\text{coupled}} - P^n\|_\infty$.

their expected temporal order of accuracy, for both velocity and pressure. In particular, methods M02 and M08 are confirmed first order methods, while all others are second order accurate in time, when combined with second order time discretization, as previously stated.

This previous result does not quantify, however, the accuracy of the splitting error. To estimate it, we plot in Fig. 3 the errors $\|U^n_{\text{coupled}} - U^n\|_\infty$ and $\|P^n_{\text{coupled}} - P^n\|_\infty$, which depict the difference between the solutions of methods M02–M10 and the result from the monolithic method (M01) for the same δt. As can be seen in the figures, the first order projection and the pseudo-exact factorization (M02 and M08, both non-incremental) have indeed splitting errors of order $\mathcal{O}(\delta t)$. The rest of the methods yield second order splitting errors, with the exception of M07, the Yosida incremental method, which yields third order splitting error, as predicted in [25]. For a thorough analysis of third order accurate splitting methods the reader is referred to the works of Shen [37], Badia and Codina [38], Owen and Codina [39] and references therein. Interestingly, we implemented the Yosida incremental method together with

Backward Differentiation Formula (BDF3) time discretization, a variant suggested by Badia and Codina [25], and confirmed full third-order convergence to the reference solution.

As expected, the algebraic splitting error of M07 is essentially zero, though rounding errors preclude it from going below 10^{-12}. The behavior of Perot's second order method, in both its non-incremental (M04) and incremental (M05) variants, deserves some comments. In what concerns M04, the velocity yielded by this method agrees to machine precision with that of the monolithic method, while the splitting error for pressure converges with $\mathcal{O}(\delta t^2)$. This velocity superconvergence is quite certainly an artifact of the specific problem and discretization. Also note that the incremental variant (M05) does not attain greater splitting accuracy than that of the non-incremental one. Further, the incremental variant behaves unstably for $\delta t > 10^{-5}$.

Finally, the exact factorization method (M10) agrees with the monolithic method, as expected, with errors always between 10^{-10} and 10^{-12}. The method is however sensitive to roundoff errors, showing a slight increase in the splitting error as δt becomes smaller, which is associated with the computation of the null spaces matrices. In our case, we used MATLAB's `null` command, that finds a rational basis for the null space by a method based on the reduced row echelon form of matrices G and D. This method is known to be sensitive to roundoff errors [40].

3. Spurious transients in algebraic splitting methods

Projection-based methods, such as the algebraic splitting ones discussed in the previous sections, are the basic tool behind the *transient* Navier–Stokes solvers in most of the commercial/open source CFD packages.

In some technologies, notably in microfluidics, the inertial effects are very small. The time scale in which mechanical equilibrium is established can thus be much smaller than the time scale of other intervening processes, such as the transport of chemical species. Consider fluid properties (e.g., viscosity) that depend, for example, on some concentration field $C(\mathbf{x}, t)$ which is convected by the fluid (with some molecular diffusion) along a microfluidic channel Ω of length ℓ and width w. Changes in the value of the inflow concentration lead in this situation to a transient process that last for at least the time $T_{\text{proc}} = \ell/U$ required for the concentration change to reach the outflow of the domain, U being the velocity scale. This induces, through the viscosity dependence on C, an equally long transient in $\mathbf{u}$ and p, so that any simulation of the process must last at least from $t = 0$ to $t = T_{\text{proc}}$.

Since the time scale of momentum diffusion is roughly $T_\nu = w^2/(4\nu)$, if

$$T_\nu \ll T_{\text{proc}} \quad \left(\text{i.e.;}\ \frac{w^2}{4\nu} \ll \frac{\ell}{U}\right) \tag{62}$$

the time-derivative term $\partial_t \mathbf{u}$ in the Navier–Stokes equations is expected to be negligible, making the dynamical variables $\mathbf{u}$ and p to evolve as a sequence of *quasi-steady* (or *quasi-static*) states, dependent on t only because the viscosity depends on t.

The condition (62) can be rewritten as

$$\frac{U\, w/\nu}{\ell/w} \ll 4 \tag{63}$$

which predicts that the quasi-steady behavior holds for small values of $U\, w/\nu$ (small Reynolds number) and/or large values of the length-to-width ratio ℓ/w, both typical of microfluidic applications.

The relevant physics of the system occurs at the time scale T_{proc}, and although this is a *slow* time scale the customary practice is to numerically approximate the flow with a *transient* solver. We will make the case below that a remarkable phenomenon takes place when the aforementioned transient solver is based on an incremental *projection-based method* (such as M03, M05 and M07 of the previous sections), even if θ is taken as 1. Specifically, a *spurious transient* develops that pollutes the numerical velocity and pressure fields. The time extent T_{num} of this spurious transient depends on δt, but *it is rarely smaller than* $\sim$100 δt *and can be as big as* 10 000 δt *or even more*. In particular, T_{num} can exceed T_{proc}, and thus deteriorate the accuracy of the solution globally in time, *even if the time step satisfies* $\delta t \ll T_{\text{proc}}$.

To justify this claim, and for the sake of simplicity, we will mainly consider the extreme case $T_{\text{proc}} \to +\infty$, which automatically satisfies $T_\nu \ll T_{\text{proc}}$ without requiring an explicit estimate for T_ν. In other words, we will consider the numerical transient that takes place when calculating a steady state, and experimentally compute its timespan T_{num} as a function of δt. Later on we provide an example of the numerical pollution that takes place if δt is chosen such that $T_{\text{num}}(\delta t)$ is comparable to or greater than T_{proc}.

In what follows, we first provide an algebraic argument (Section 3.1) and then an analytic argument arising from a thin-film analogy (Section 3.2) to explain the spurious-transient phenomenon. Finally, we report on some numerical experiments, so as to provide the user with some quantitative examples of what to expect in actual simulations.

3.1. Algebraic estimates of spurious transients

Let us consider a problem with negligible inertia effects ($\mathsf{T}(U^n) \simeq 0$), of which the steady state solution is sought. Without loss of generality, we will assume $r^n = 0$ so that the steady solution consists of zero velocities and pressures. Further, let us adopt a *fully implicit* scheme ($\theta = 1$), so that

$$\mathsf{A} = \frac{1}{\delta t}\mathsf{I_u} - \nu\,\mathsf{L} \quad \text{and} \quad \mathsf{N} = \frac{1}{\delta t}\mathsf{I_u}.$$

The evolution matrix $\mathsf{S}_{\text{coupled}}$ for the monolithic method, taken from (24), reads

$$\mathsf{S}_{\text{coupled}} = \begin{bmatrix} \left[\mathsf{I_u} - \mathsf{A}^{-1}\mathsf{G}(\mathsf{D}\mathsf{A}^{-1}\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{N} & \mathsf{0} \\ (\mathsf{D}\mathsf{A}^{-1}\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{N} & \mathsf{0} \end{bmatrix}.$$

Taking an arbitrarily large δt the monolithic method provides the steady state in just one time step, because

$$\mathsf{N} \overset{\delta t\to+\infty}{\longrightarrow} 0, \qquad \mathsf{A}^{-1} \overset{\delta t\to+\infty}{\longrightarrow} -\frac{1}{\nu}\mathsf{L}^{-1} \quad \text{and thus} \quad \mathsf{S}_{\text{coupled}} \overset{\delta t\to+\infty}{\longrightarrow} 0.$$

The behavior of the algebraic splitting methods is quite different. Consider their evolution matrix extracted from (21), that is

$$\mathsf{S} = \begin{bmatrix} \left[\mathsf{I_u} - \mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{N} & -\gamma\left[\mathsf{I_u} - \mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{G} \\ (\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{N} & -\gamma\left[(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{G} - \mathsf{I_p}\right] \end{bmatrix}.$$

The first conclusion that can be drawn from (21) is

Proposition 1. *If the approximate inverses* B_1 *and* B_2 *of the algebraic split method are such that*

$$\frac{1}{\delta t}\mathsf{B}_2(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D} \overset{\delta t\to\infty}{\longrightarrow} 0 \quad \textit{and} \quad \frac{1}{\delta t}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1} \overset{\delta t\to\infty}{\longrightarrow} 0$$

then the ***non-incremental version*** *($\gamma = 0$) of the method yields the steady state solution in just one (sufficiently large) time step.*

Proof. Since $\gamma = 0$, only blocks S_{11} and S_{21} need to be analyzed. Moreover, since B_1 and B_2 are matrices satisfying the hypothesis, it is easy to note that

$$\mathsf{S}_{11} = \left[\mathsf{I_u} - \mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1}\mathsf{N} = \frac{1}{\delta t}\left[\mathsf{I_u} - \mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\right]\mathsf{A}^{-1} \overset{\delta t\to\infty}{\longrightarrow} 0,$$

since A^{-1} is bounded. In a similar way,

$$\mathsf{S}_{21} = (\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{N} = \frac{1}{\delta t}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1} \overset{\delta t\to\infty}{\longrightarrow} 0.$$

Therefore $\mathsf{S} \overset{\delta t\to\infty}{\longrightarrow} 0$, which concludes the proof. □

Before applying this proposition to the methods seen here, we state the following lemma:

Lemma 1. *If the approximate inverses* B_1 *and* B_2 *satisfy one of the following hypotheses:*

(i) B_1 *is independent of* δt*, with* $(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}$ *and* B_2 *bounded as* $\delta t \to \infty$*;*
(ii) $\mathsf{B}_1 = \delta t^{q_1}\mathsf{M}_1$*, with* M_1 *and* B_2 *bounded as* $\delta t \to \infty$*;*
(iii) $\mathsf{B}_1 = \delta t^{q_1}\mathsf{M}_1$ *and* $\mathsf{B}_2 = \delta t^{q_2}\mathsf{M}_2$*, for* $q_2 \le q_1$*, with* M_1 *and* M_2 *bounded as* $\delta t \to \infty$*;*

Then the following holds:

$$\frac{1}{\delta t}\mathsf{B}_2(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D} \overset{\delta t\to\infty}{\longrightarrow} 0 \quad \textit{and} \quad \frac{1}{\delta t}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1} \overset{\delta t\to\infty}{\longrightarrow} 0. \tag{64}$$

Proof. Indeed, if hypothesis (i) holds, the conclusion is straightforward. If hypothesis (ii) holds, then

$$(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1} = \frac{1}{\delta t^{q_1}}(\mathsf{D}\mathsf{M}_1\mathsf{G})^{-1} \overset{\delta t\to\infty}{\longrightarrow} 0,$$

and since B_2 is bounded, we can easily conclude the validity of (64). Finally, in case (iii) holds, then

$$\mathsf{B}_2\mathsf{G}(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D} = \delta t^{(q_2-q_1)}\mathsf{M}_2\mathsf{G}(\mathsf{D}\mathsf{M}_1\mathsf{G})^{-1}\mathsf{D}$$

that either converges to the null matrix or is bounded, since the exponent $(q_2 - q_1) \leq 0$. □

Corollary 1. *The non-incremental versions of the first order projection method* (M02)*, of Perot's second order projection method* (M04)*, of Yosida method* (M06) *and of the pseudo-exact factorization method* (M08) *yield the steady state solution in just one (sufficiently large) time step.*

Proof. Recalling that the approximation matrices in methods M02 and M04 are based on the inverse expansion (25), we have

$$\begin{aligned}\mathsf{B}_1 &= \mathsf{B}_2 = \delta t\ \mathsf{I}_\mathsf{u} + \sum_{i=1}^{q} \delta t^{i+1}\ (\theta\nu\mathsf{L})^i \\ &= \delta t^{q+1}\left[\delta t^{-q}\ \mathsf{I}_\mathsf{u} + \sum_{i=1}^{q} \delta t^{i-q}\ (\theta\nu\mathsf{L})^i\right] = \delta t^{q+1}\ \mathsf{M}\end{aligned} \tag{65}$$

where $\mathsf{M} \overset{\delta t\to\infty}{\longrightarrow} (\theta\nu\mathsf{L})^q$, for some $q \in \mathbb{N}$. Therefore, these methods satisfy hypothesis (iii) of Lemma 1, and consequently Proposition 1 applies. It is also easy to verify that the non-incremental version of Yosida method (M06) satisfies hypothesis (ii), with $\mathsf{B}_1 = \delta t\mathsf{I}_\mathsf{u}$ and $\mathsf{B}_2 = \mathsf{A}^{-1}$ bounded. For the non-incremental version of the pseudo-exact factorization method (M08), we observe that B_1 and B_2 are both given by Eq. (53), and thus

$$\left(\mathsf{D}\mathsf{B}_1\mathsf{G}\right)^{-1} = \left(\mathsf{D}\mathsf{G}(\mathsf{D}\mathsf{A}\mathsf{G})^{-1}\mathsf{D}\mathsf{G}\right)^{-1} = (\mathsf{D}\mathsf{G})^{-1}(\mathsf{D}\mathsf{A}\mathsf{G})(\mathsf{D}\mathsf{G})^{-1},$$

from which it is possible to conclude that matrices $(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}$ and B_2 are both bounded as $\delta t \to \infty$, satisfying hypothesis (i) of Lemma 1, and therefore concluding the proof. □

Remark 3. It has thus been proved that non-incremental algebraic splitting methods do not exhibit spurious transients and that they arrive at their steady state solution in just one iteration if inertia effects are negligible. A most important caveat regarding these methods, however, is that *their steady state solutions depend on the time step size*. This fact is well known for the first-order projection method [41]. We later illustrate through numerical examples that a reasonable accuracy in the steady state velocity and pressure is only attained when δt is comparable to the stability limit of *explicit* time-integration methods. This restriction in the choice of δt holds not just for the first-order projection method, but for all non-incremental splitting techniques considered in this article. As expected, such a small δt is not practical for computing steady states or, more generally, slowly-evolving flows because *millions* of time steps are required.

Consider now *incremental* schemes ($\gamma = 1$), which by construction yield *the same steady state solution as the coupled method*. From Proposition 1, we know that the first column of submatrices of S tends to zero as $\delta t \to +\infty$, so that the evolution of the discrete system depends on the spectral properties of submatrix

$$\mathsf{S}_{22} = -\gamma\left[(\mathsf{D}\mathsf{B}_1\mathsf{G})^{-1}\mathsf{D}\mathsf{A}^{-1}\mathsf{G} - \mathsf{I}_\mathsf{p}\right].$$

We thus can infer the following:

Proposition 2. *The incremental projection method* (M03)*, as well as the incremental versions of Perot's second order method* (M05) *and of Yosida method* (M07)*, exhibit* ***spurious transients*** *when δt is large, in the sense that Q eigenvalues of the evolution matrix S tend to* 1 *as $\delta t \to +\infty$.*

Proof. A common property shared by methods M03, M05 and M07, is that $\left(\mathsf{DB}_1\mathsf{G}\right)^{-1} \overset{\delta t\to\infty}{\longrightarrow} 0$, while $\mathsf{A}^{-1} \overset{\delta t\to\infty}{\longrightarrow} -(\nu\mathsf{L})^{-1}$, resulting in

$$\mathsf{S}_{22} \overset{\delta t\to\infty}{\longrightarrow} \gamma\mathsf{I}_\mathsf{p}.$$

Therefore, since $\mathsf{I}_\mathsf{p} \in \mathbb{R}^{Q\times Q}$ and $\gamma = 1$, at least Q eigenvalues of S tend to 1 as $\delta t \to \infty$. □

Remark 4. In the incremental version of the pseudo-exact factorization method, on the other hand, one has

$$\mathsf{S}_{22} \overset{\delta t\to+\infty}{\longrightarrow} -\gamma\left[(\mathsf{DG})^{-1}\mathsf{DLG}(\mathsf{DG})^{-1}\mathsf{DL}^{-1}\mathsf{G} - \mathsf{I}_\mathsf{p}\right] \tag{66}$$

making it difficult to estimate the spectrum in general cases, so that its actual behavior is left to be assessed numerically.

The consequence of Proposition 2 is that incremental algebraic splitting methods, in particular the popular incremental projection method (M03), cannot be used to compute steady-state or slowly-evolving solutions by simply choosing δt to be sufficiently large. In much the same way as the Crank–Nicolson method when applied to the heat equation, exceedingly large values of δt lead to eigenvalues of the evolution matrix with modulus practically equal to one. This gives rise to *spurious transients*; i.e., numerical transients with timespan much bigger than the physical transient of the simulated system.

3.2. Spurious transients in a thin-film analogy

It is possible to analytically visualize the spurious transients generated by incremental projection-based methods by considering a thin-film analogy. This will provide us with estimates of the solution behavior that are independent of the spatial discretization. The argument is given below in two dimensions, but it is readily extendable to 3D.

In a narrow channel of length ℓ and width w, with $w \ll \ell$, it is well-known that neglecting inertia effects the velocity profile at each location along the channel can be accurately approximated by a parabolic function. Plugging such profile into the incompressible Navier–Stokes equations (1)–(2) one arrives at

$$\partial_t v + \sigma\, v + \partial_x p = f \tag{67}$$

$$\partial_x v = 0 \tag{68}$$

where

$$\sigma = \frac{12\,\nu}{w^2},$$

$v(x,t)$ is the cross-section-averaged velocity, and the force f (along the flow direction x) is assumed to depend only on x.

Let us assume *periodic* conditions to hold at $x = 0$ and $x = \ell$, so that the narrow channel under consideration becomes the *simplest* hydraulic circuit: A closed loop. Since this is a toy problem that will be used just to reason by analogy, let us assume that σ may depend on x and denote by overlines the domain averages, such as

$$\overline{f} = \frac{1}{\ell}\int_0^\ell f(x)\,dx \quad \text{and} \quad \overline{\sigma} = \frac{1}{\ell}\int_0^\ell \sigma(x)\,dx.$$

The exact steady state of (67)–(68) is given by

$$v^\infty = \frac{\overline{f}}{\overline{\sigma}}, \qquad \partial_x p^\infty = f - \frac{\overline{f}}{\overline{\sigma}}\,\sigma. \tag{69}$$

Consider now a fractional step scheme that mimics the projection schemes M02 and M03 (with $\theta = 1$), i.e.,

$$\frac{1}{\delta t}(\widetilde{V} - V^n) + \sigma\,\widetilde{V} = f - \gamma\,\partial_x P^n \tag{70}$$

$$\partial^2_{x^2}(P^{n+1} - \gamma\,P^n) = \frac{1}{\delta t}\,\partial_x \widetilde{V} \tag{71}$$

$$V^{n+1} = \widetilde{V} - \delta t\,\partial_x(P^{n+1} - \gamma\,P^n) \tag{72}$$

to be solved with periodic boundary conditions in x and initial conditions V^0 and P^0.

Eqs. (70)–(72) can be solved analytically, yielding

$$V^{n+1} - v^\infty + \delta t\,\partial_x(P^{n+1} - p^\infty) = \frac{1}{1+\sigma\,\delta t}(V^n - v^\infty) + \frac{\sigma\,\delta t}{1+\sigma\,\delta t}\,\delta t\,\partial_x(\gamma\,P^n - p^\infty). \tag{73}$$

Notice that this last equation determines both V^{n+1} and $\partial_x P^{n+1}$, because

$$\overline{V^{n+1}} = V^{n+1} \in \mathbb{R} \quad \text{and} \quad \overline{\partial_x P^{n+1}} = 0.$$

3.2.1. Steady-state error of the non-incremental scheme

Consider first the steady state obtained from (70)–(72) when $\gamma = 0$. Taking $V^{n+1} = V^n = V^*$ and $P^{n+1} = P^n = P^*$ we get

$$V^* - v^\infty = \frac{\overline{\left(\frac{\delta t\,\partial_x p^\infty}{1+\sigma\,\delta t}\right)}}{\overline{\left(\frac{\sigma\,\delta t}{1+\sigma\,\delta t}\right)}} \tag{74}$$

$$\partial_x(P^* - p^\infty) = -\frac{\sigma\,\delta t}{1+\sigma\,\delta t}\,\partial_x p^\infty - \frac{\overline{\left(\frac{\partial_x p^\infty}{1+\sigma\,\delta t}\right)}}{\overline{\left(\frac{\sigma\,\delta t}{1+\sigma\,\delta t}\right)}}\,\frac{\sigma\,\delta t}{1+\sigma\,\delta t}. \tag{75}$$

The numerical steady state (V^*, P^*) does not, in general, coincide with the exact steady state (v^∞, p^∞). The error of the steady state can indeed be estimated as $\mathcal{O}(\delta t)$. Interestingly, if the channel is homogeneous (σ independent of x) then from (74)–(75) and $\overline{\partial_x p^\infty} = 0$ we have that

$$V^* = v^\infty \quad \text{and} \quad \partial_x(P^* - p^\infty) = -\frac{\sigma\,\delta t}{1+\sigma\,\delta t}\,\partial_x p^\infty$$

and thus the error of the steady state only affects the pressure (in fact, if $\sigma\,\delta t \gg 1$ one has $P^* \simeq 0$!).

3.2.2. Spurious transient of the incremental scheme

The steady state of the incremental scheme coincides with the exact steady state (v^∞, p^∞). There however appears a spurious transient in the semi-discrete (in time) solution that precludes the method from rapidly achieving the steady state when δt is large.

To see this, let us define the auxiliary variables $\varphi^n \in \mathbb{R}$ and $\psi^n \in L^2(0,\ell)$, with zero mean, by

$$\varphi^n = V^n - v^\infty, \qquad \psi^n = \delta t\,\partial_x(P^n - p^\infty).$$

Substituting into (73) leads to the evolution system

$$\varphi^{n+1} = \overline{\left(\frac{1}{1+\sigma\,\delta t}\right)}\,\varphi^n + \overline{\left(\frac{\sigma\,\delta t\,\psi^n}{1+\sigma\,\delta t}\right)} \tag{76}$$

$$\psi^{n+1} = \left[\frac{1}{1+\sigma\,\delta t} - \overline{\left(\frac{1}{1+\sigma\,\delta t}\right)}\right]\varphi^n + \frac{\sigma\,\delta t}{1+\sigma\,\delta t}\,\psi^n - \overline{\left(\frac{\sigma\,\delta t\,\psi^n}{1+\sigma\,\delta t}\right)}. \tag{77}$$

Notice first that, if the channel is homogeneous (σ independent of x), the system reduces to

$$\varphi^{n+1} = \frac{1}{1+\sigma\,\delta t}\,\varphi^n, \qquad \psi^{n+1} = \frac{\sigma\,\delta t}{1+\sigma\,\delta t}\,\psi^n.$$

For large time step sizes, $\sigma\delta t \gg 1$, the velocity V^n rapidly converges to the steady state v^∞. The pressure, on the other hand, has $\psi^{n+1} \simeq \psi^n$ and thus exhibits a *spurious transient* with slow decay.

When the channel is heterogeneous the situation is more complex and in general both the velocity and the pressure exhibit spurious, slowly-decaying transients before attaining the steady state. An illustrative example can be found in the Appendix.

The previous analysis suggests that spurious transients occur when $\sigma\delta t$ is comparable to or greater than one. This can be rewritten in terms of the momentum-diffusion time to provide a criterion for the selection of δt. Indeed, selecting

$$\delta t \le \delta t^* = \frac{T_\nu}{48} \tag{78}$$

seems to be a useful criterion to avoid the spurious transient phenomenon.

3.3. Spurious transients of low-Reynolds flows: numerical experiments

A crucial point made in this article is that projection-based methods can be quite inefficient in the computation of steady or slowly varying flows. Their non-incremental versions ($\gamma = 0$) yield solutions with unacceptably large errors, unless δt is taken very small. Their incremental versions ($\gamma = 1$), in turn, yield results polluted by long spurious transients.

To show that these difficulties are the rule and not the exception, and illustrate their quantitative significance, let us consider the simplest hydraulic circuit possible, a closed loop, which has been already discussed in the previous section.

Let Ω represent a rectangular channel with dimensions $[0, \ell] \times [0, w]$, discretized by a structured grid of $N_1 \times N_2$ cells. The boundary conditions impose periodicity along x_1 and no-slip along the lower ($x_2 = 0$) and upper ($x_2 = w$) walls.

The force $\mathbf{f}$ driving the flow is not taken as constant to avoid an identically zero pressure gradient. Specifically, we select

$$\mathbf{f}(\mathbf{x}) = (f(x_1), 0), \quad \text{with } f(x_1) = \begin{cases} \dfrac{16W\nu}{w^2} & \text{if } x_1 < \dfrac{\ell}{2} \\ 0 \quad \text{otherwise} \end{cases}$$

where W is an arbitrary velocity scale, so that the exact steady state is

$$\begin{aligned} u_1^\infty &= 4\,W\left(\frac{x_2}{w} - \frac{x_2^2}{w^2}\right) \\ u_2^\infty &= 0 \\ \partial_{x_1} p^\infty &= \begin{cases} \dfrac{8W\nu}{w^2} & \text{if } x_1 < \dfrac{\ell}{2} \\ -\dfrac{8W\nu}{w^2} & \text{if } \dfrac{\ell}{2} \le x_1 \end{cases} \\ \partial_{x_2} p^\infty &= 0 \end{aligned} \tag{79}$$

with maximum velocity $u_1^\infty(x_1, x_2 = w/2) = W$ and $\overline{\partial_{x_1} p^\infty} = 0$ as expected. The Reynolds number of the resulting flow is defined as

$$Re = \frac{W\,w}{\nu}.$$

In what follows, we select $W = 10^{-3}$ m/s, $w = 10^{-5}$ m and $\nu = 10^{-6}$ m^2/s, mimicking water flowing at 1 mm/s along a 10-micron-wide channel, which results in $Re = 10^{-2}$. The time scale for this system to attain its steady state

is the momentum-diffusion time

$$T_\nu \simeq \frac{w^2}{4\nu} = 2.5 \times 10^{-5}\ \text{s}. \tag{80}$$

The aspect ratio of the domain is taken as $\ell/w = 3$.

On any given spatial mesh, there exist discrete steady pressure and velocity fields, denoted by U^∞ and P^∞, which are the unique solutions of the nonlinear system

$$-\nu \mathsf{L}\, U^\infty + T(U^\infty) + \mathsf{G}\, P^\infty = b \tag{81}$$

$$\mathsf{D}\, U^\infty = 0 \tag{82}$$

where b comes from the (time independent) forcing.

To compute the steady state with a transient solver, the customary practice is to take a large time step, so as to bypass the physical transient and hopefully obtain the steady solution after a few time steps. Indeed, on a 60×20 mesh the monolithic method reaches the discrete steady solution (U^∞, P^∞) in seven time steps with $\delta t = 10^{-5}$ s, and in just three time steps with any $\delta t > 10^{-4}$ s. The physical estimate (80) is thus seen to hold for the discrete problem, the steady state not being attained in a single step because $Re > 0$ and the nonlinear term is treated explicitly.

In what follows, we will discuss the outcome of the different algebraic splitting methods in this specific problem by examining the numerical solutions as obtained with different time steps. All the results discussed below were obtained with Backward Euler time stepping ($\theta = 1$).

3.3.1. Non-incremental schemes: steady-state error

Let us begin by the first-order projection scheme (M02) as defined in (26)–(28). It is already known that the solution of (26)–(28) depends on time step size δt [41]. We reproduce below this simple analysis to better illustrate it in the algebraic framework. Particularizing it for $\theta = 1$, at each time step the code computes:

1. The auxiliary field $\widetilde{U}$, which exactly satisfies the momentum equation (26); i.e.,

$$\frac{\widetilde{U} - U^n}{\delta t} - \nu \mathsf{L}\, \widetilde{U} + \mathsf{T}(U^n) = b. \tag{83}$$

2. The updated pressure field P^{n+1} that makes the updated velocity field U^{n+1} to exactly satisfy $\mathsf{D}\, U^{n+1} = 0$, which results from

$$\delta t\, \mathsf{D}\, \mathsf{G}\, P^{n+1} = \mathsf{D}\, \widetilde{U}. \tag{84}$$

3. Finally, the updated velocity field U^{n+1}, from

$$U^{n+1} = \widetilde{U} - \delta t\, \mathsf{G}\, P^{n+1}. \tag{85}$$

The dependency on δt can be seen by eliminating $\widetilde{U}$ to obtain

$$\frac{U^{n+1} - U^n}{\delta t} - \nu \mathsf{L}\, U^{n+1} + \mathsf{T}(U^n) + \mathsf{G}\, P^{n+1} - \nu\, \delta t\, \mathsf{L}\, \mathsf{G}\, P^{n+1} = b \tag{86}$$

$$\mathsf{D}\, U^{n+1} = 0 \tag{87}$$

in which the extra term $-\nu\, \delta t\, \mathsf{L}\, \mathsf{G}\, P^{n+1}$ is of order $\mathcal{O}(\nu \delta t / h^2)$, h being the mesh size. The numerical steady state (U^*, P^*) is defined as the solution of (86)–(87) when $U^{n+1} = U^n = U^*$ and $P^{n+1} = P^n = P^*$, that is

$$-\nu \mathsf{L}\, U^* + \mathsf{T}(U^*) + \mathsf{G}\, P^* - \nu\, \delta t\, \mathsf{L}\, \mathsf{G}\, P^* = b \tag{88}$$

$$\mathsf{D}\, U^* = 0 \tag{89}$$

which shows that the aforementioned extra term is active even at the steady state, so that U^* and P^* depend on δt. To illustrate this in the simple example (given by Eq. (79)) with $Re = 10^{-2}$, we computed this flow for several δt's. The steady state errors $\|U^* - U^\infty\|_2/\|U^\infty\|_2$ and $\|P^* - P^\infty\|_2/\|P^\infty\|_2$ can be found in Table 2.

Notice how inaccurate the numerical steady state is for essentially any δt. To have a relative error below 1% in both velocity and pressure, the time step needs to be chosen smaller than 10^{-8} s. This is in agreement with early

Table 2
Steady state errors in U and P for the first order projection method M02 with several values of δt.

δt (s)	$\|U^* - U^\infty\|_2/\|U^\infty\|_2$	$\|P^* - P^\infty\|_2/\|P^\infty\|_2$
10^{-2}	$4.790352 \cdot 10^{-1}$	1.000029
10^{-3}	$4.787257 \cdot 10^{-1}$	1.000291
10^{-4}	$4.757764 \cdot 10^{-1}$	1.002987
10^{-5}	$4.550128 \cdot 10^{-1}$	1.026256
10^{-6}	$3.781178 \cdot 10^{-1}$	$9.845069 \cdot 10^{-1}$
10^{-7}	$1.919132 \cdot 10^{-1}$	$2.895994 \cdot 10^{-1}$
10^{-8}	$3.644886 \cdot 10^{-2}$	$5.423391 \cdot 10^{-2}$
10^{-9}	$4.063221 \cdot 10^{-3}$	$5.995474 \cdot 10^{-3}$

Table 3
Steady state errors in U and P for Perot's second order projection method M04, with several values of δt.

δt (s)	$\|U^* - U^\infty\|_2/\|U^\infty\|_2$	$\|P^* - P^\infty\|_2/\|P^\infty\|_2$
10^{-2}	∞	∞
10^{-3}	∞	∞
10^{-4}	∞	∞
10^{-5}	∞	∞
10^{-6}	∞	∞
10^{-7}	∞	∞
10^{-8}	$2.045112 \cdot 10^{-3}$	$3.396125 \cdot 10^{-3}$
10^{-9}	$1.607652 \cdot 10^{-5}$	$1.831227 \cdot 10^{-5}$

observations by Rannacher [41] stating that accurate steady states require δt to be smaller than the stability limit δt_{exp} of *explicit* schemes, which in this case takes the value [42]

$$\delta t_{\text{exp}} = \frac{h^2}{4\nu} = 6.1 \times 10^{-8} \text{ s}.$$

In Fig. 4 we investigate the dynamics of this numerical scheme by plotting the evolution of $\|U^n - U^*\|_2/\|U^*\|_2 + \|P^n - P^*\|_2/\|P^*\|_2$ for δt between 10^{-2} and 10^{-9} (remember that (U^*, P^*) depends on δt). In both cases the code, starting from zero initial conditions, reaches the steady state at about $t = 10^{-4}$ s. This coincides with the time needed to reach steady state as predicted by the monolithic method, so that *no spurious transient appears*. Unfortunately, since from Table 2 one knows that only computations with very small δt (smaller than 10^{-8} s) lead to an acceptably accurate steady state, it becomes evident that at least about 10 000 time steps are needed to compute (U^*, P^*). This behavior alone practically rules out the first order projection method for computing steady states or any flow evolving in time with a time scale much longer than T_ν.

To illustrate the behavior of the numerical solution and its dependence on the time step size, we plot profiles of component U_1 of the velocity field and of the pressure P, computed with $\delta t = 10^{-k}$, $k = 4, 5, 6, 7$, which can be seen in Fig. 5. From these results one observes how far the velocity and pressure fields are from the expected steady state solution (U^∞, P^∞) throughout the domain. Decreasing δt, the velocity and pressure profiles slowly converge to the correct ones, but reasonably accurate steady states are only obtained if $\delta t < 10^{-8}$ s.

Similar assessments can be performed on other non-incremental methods. The steady state errors are reported in Table 3 for Perot's second order projection method (M04), in Table 4 for Yosida method (M06), and in Table 5 for the pseudo-exact factorization method (M08), all of them non-incremental.

From Table 3 one can observe that the method M04 diverges for values of $\delta t \geq 10^{-7}$ s. Indeed, this non-incremental scheme has essentially the same stability limit $\delta t < \delta t_{\text{exp}}$, irrespective of θ being 1 (implicit splitting) or 0 (explicit), the latter being obviously cheaper in computer time.

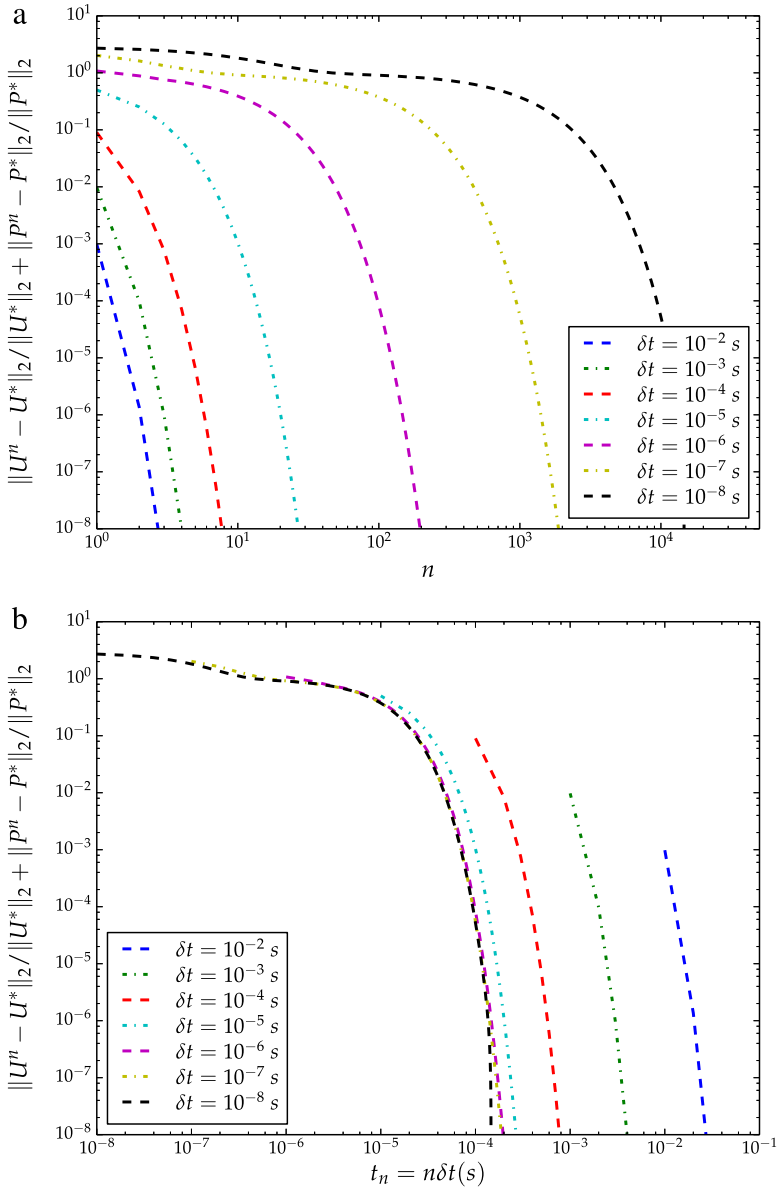
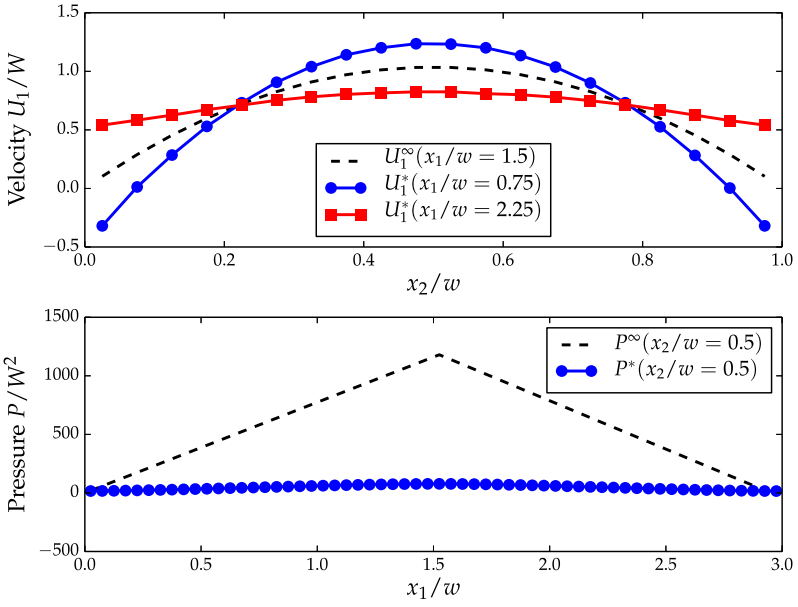


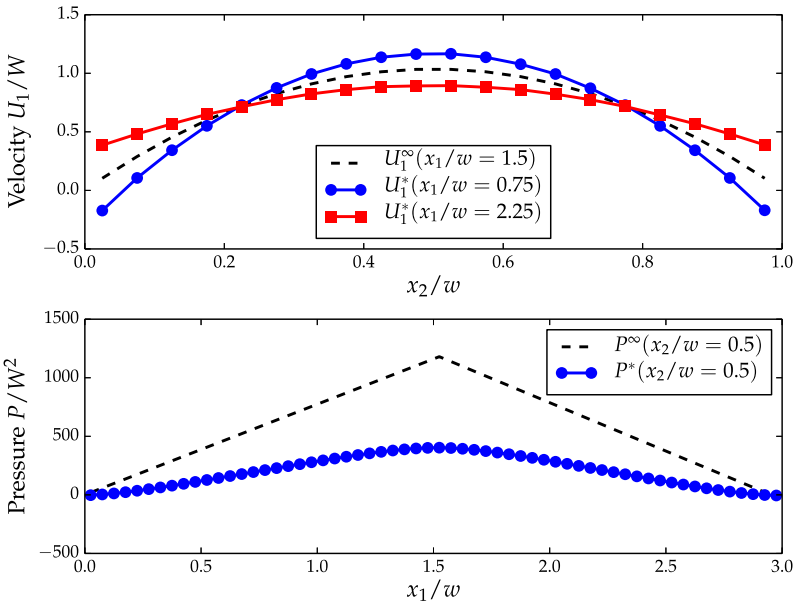
Fig. 4. Evolution of the error between the solution obtained by method M02 and its steady state (U^*, P^*): (a) as a function of iteration number n; (b) as a function of time $t_n = n\,\delta t$ s.

Table 4
Steady state errors in U and P for the Yosida method M06, with several values of δt.

δt (s)	$\|U^* - U^\infty\|_2/\|U^\infty\|_2$	$\|P^* - P^\infty\|_2/\|P^\infty\|_2$
10^{-2}	$5.810048 \cdot 10^{-1}$	1.000031
10^{-3}	$5.799267 \cdot 10^{-1}$	1.000315
10^{-4}	$5.693392 \cdot 10^{-1}$	1.003514
10^{-5}	$4.947052 \cdot 10^{-1}$	1.045886
10^{-6}	$3.673654 \cdot 10^{-1}$	$5.507672 \cdot 10^{-1}$
10^{-7}	$1.284121 \cdot 10^{-1}$	$3.612108 \cdot 10^{-1}$
10^{-8}	$4.415307 \cdot 10^{-3}$	$6.383871 \cdot 10^{-3}$
10^{-9}	$5.736004 \cdot 10^{-5}$	$7.279366 \cdot 10^{-5}$



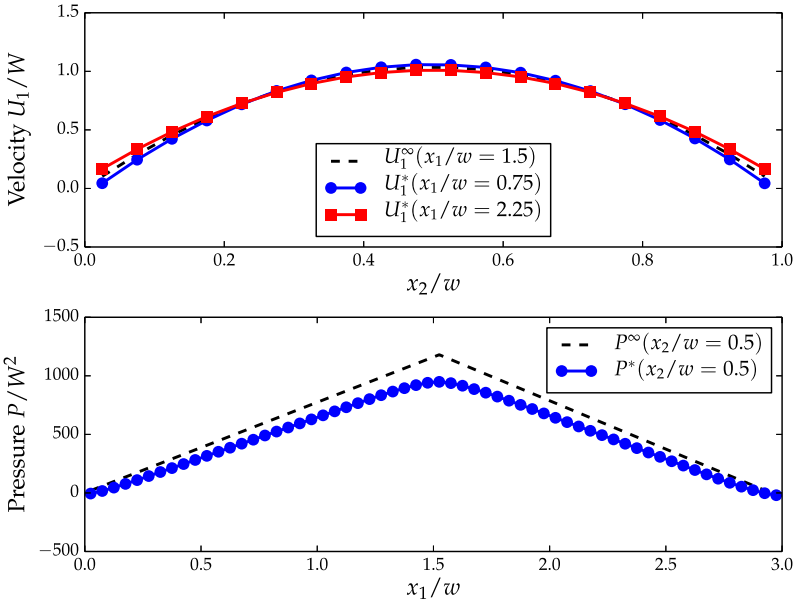
(a) $\delta t = 10^{-4}$ s.



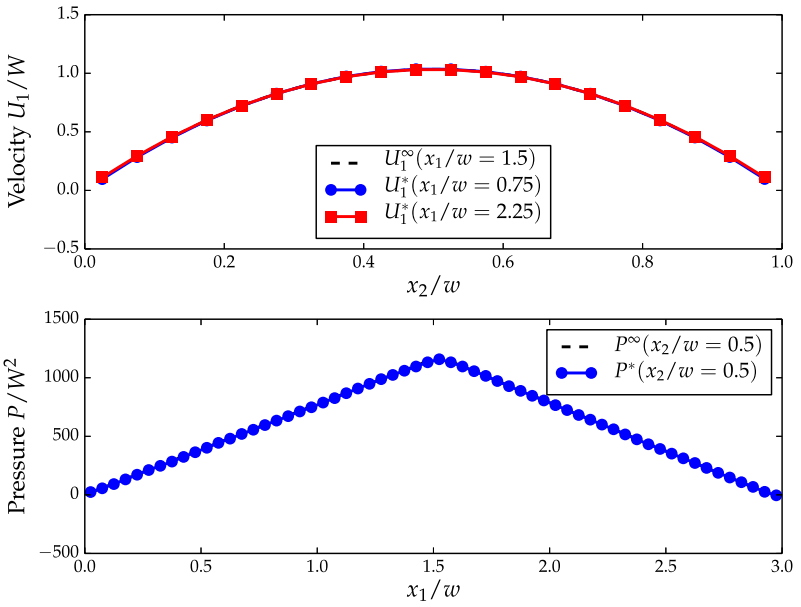
(b) $\delta t = 10^{-5}$ s.

Fig. 5. Stationary velocity (component u_1) and pressure (p) profiles, obtained by the method M02, illustrating how the dependency on δt affects the final solution for this problem. Velocity and pressure values are non-dimensional.

The steady state errors for the non-incremental Yosida method M06 are displayed in Table 4, showing the same behavior as M02. In this case however, the errors for small δt are somewhat smaller than those of Table 2, a consequence of the method's second order accuracy in time. Furthermore, Table 5 displays the steady state errors for the non-incremental version of pseudo-exact factorization method M08. This time, the errors are similar in magnitude to the ones presented in Table 2, confirming its first order accuracy in time. Both M06 and M08 present no spurious transients as predicted by theory, with a behavior very similar to that of M02 (in Fig. 4), therefore the dynamics of the steady state error is omitted here for brevity.



(c) $\delta t = 10^{-6}$ s.



(d) $\delta t = 10^{-7}$ s.

Fig. 5. (*continued*)

3.4. Incremental schemes: spurious numerical transients

The equations defining the incremental projection scheme (method M03) for $\theta = 1$ are (31)–(34), which upon elimination of $\widetilde{U}$ and $\widetilde{P}$ give

$$\frac{U^{n+1} - U^n}{\delta t} - \nu \mathsf{L}\, U^{n+1} + \mathsf{T}(U^n) + \mathsf{G}\, P^{n+1} - \nu\, \delta t\, \mathsf{LG}(P^{n+1} - P^n) = b^n \tag{90}$$

$$\mathsf{D}\, U^{n+1} = 0. \tag{91}$$

Table 5
Steady state errors in U and P for the pseudo-exact factorization method M08, with several values of δt.

δt (s)	$\|U^* - U^\infty\|_2/\|U^\infty\|_2$	$\|P^* - P^\infty\|_2/\|P^\infty\|_2$
10^{-2}	$4.790352 \cdot 10^{-1}$	2.708501
10^{-3}	$4.787257 \cdot 10^{-1}$	2.704751
10^{-4}	$4.757764 \cdot 10^{-1}$	2.669166
10^{-5}	$4.550128 \cdot 10^{-1}$	2.432502
10^{-6}	$3.779709 \cdot 10^{-1}$	$2.608038 \cdot 10^{-1}$
10^{-7}	$1.919132 \cdot 10^{-1}$	$1.088018 \cdot 10^{-1}$
10^{-8}	$3.644851 \cdot 10^{-2}$	$2.058955 \cdot 10^{-2}$
10^{-9}	$4.042745 \cdot 10^{-3}$	$2.289435 \cdot 10^{-3}$

The fixed point of this system (defined as before by $U^{n+1} = U^n = U^*$ and $P^{n+1} = P^n = P^*$) is thus seen to coincide with the discrete steady state (U^∞, P^∞). However, a *spurious transient* appears in the evolution of the discrete system towards the steady state if δt is large.

To avoid an excessive number of plots, let us introduce the following measure of "distance to steady state":

$$D_{ss}^n \stackrel{\text{def}}{=} \frac{\|U^n - U^\infty\|_2}{\|U^\infty\|_2} + \frac{\|P^n - P^\infty\|_2}{\|P^\infty\|_2}. \tag{92}$$

To establish a base case for comparison, we show in Fig. 6 the distance D_{ss}^n for the monolithic method. Notice that no spurious transient is observed: the number of time steps needed to reduce D_{ss}^n below any given tolerance decreases monotonically as the time step is increased. Any simulation with $\delta t \geq 10^{-4}$ s, in particular, quickly reduces the steady state error in just a few time steps.

Results for the evolution of D_{ss}^n for the incremental projection method M03 are presented in Fig. 7. The abscissa in part (a) of the figure is the time index n, while that of part (b) is the physical time t. By arbitrarily defining a tolerance of 10^{-6} and intersecting the line $D_{ss}^n = 10^{-6}$ with the curve corresponding to each δt, one obtains a measure, denoted by T_{num}, of the timespan of the *numerical transient*.

The numerical transients coincide for all $\delta t \leq 10^{-7}$ s, yielding $T_{\text{num}} = 10^{-4}$ s. This is a *physical transient*, as shown by the agreement with results obtained with the monolithic method above and with the estimate $T_\nu = 2.5 \times 10^{-5}$ s.

To reduce the distance to steady state to 10^{-6} with minimal effort the best choice is $\delta t = 10^{-6}$ s, resulting in about two hundred iterations (time steps). These two hundred iterations corresponds to a *numerical transient* that lasts about $T_{\text{num}} = 2 \times 10^{-4}$ s, twice the timespan of the physical transient. In Fig. 7(b) one observes that the time evolution of the system towards steady state with this δt departs from the correct one that is obtained with smaller time steps. If δt is taken greater than 10^{-6} s, this departure grows enormously. The timespan T_{num} of the numerical transient is of 0.02 s for $\delta t = 10^{-5}$ s, and more than 2 s for $\delta t = 10^{-4}$ s. These are obviously *spurious* transients, in the sense that they are purely numerical artifacts with no similarity to the physical transient.

Notice that the estimate obtained from the thin-film analogy yields in this case

$$\delta t^* = \frac{T_\nu}{48} \simeq 5 \times 10^{-7} \text{ s},$$

correctly predicting the appearance of spurious transients for $\delta t > \delta t^*$. The steady state is obtained with minimal computational effort with $\delta t \simeq 2\,\delta t^*$.

Allow us now, for a moment, to consider a transient version of this same problem, in which the initial velocity is zero and the force $\mathbf{f}(\mathbf{x})$ is modulated in time by multiplying it by the function

$$g(t) = \begin{cases} \sin(\pi\, t) & \text{if } 0 \leq t \leq 1 \text{ s} \\ 0 & \text{if } t > 1 \text{ s}. \end{cases}$$

Our goal is to provide a more vivid image of the effect of large spurious transients in the simulation of a truly transient microfluidic-like problem. The typical time of the process is $T_{\text{proc}} = 1$ s, much greater than T_ν.

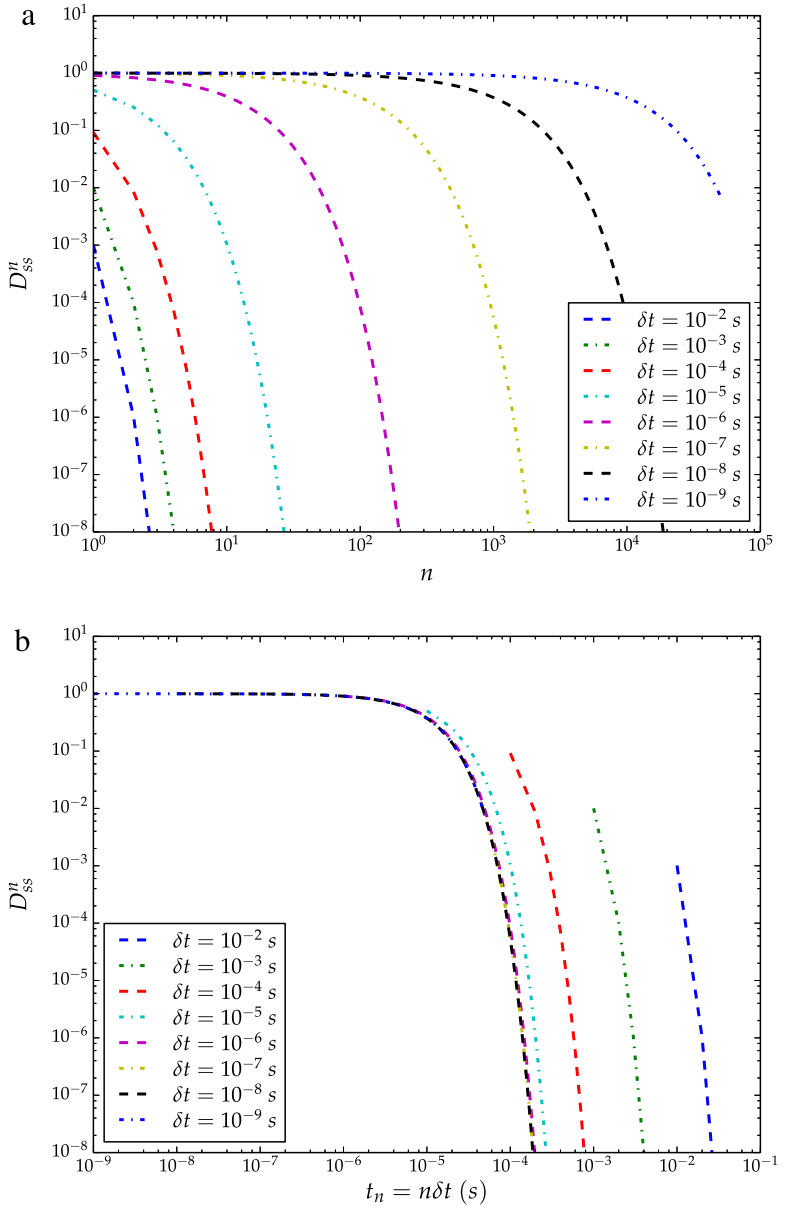


Fig. 6. Evolution of D^n_{ss} for the monolithic method M01, for several δt: (a) as a function of the iteration number n; (b) as a function of time $t_n = n\,\delta t$ s.

We ran the incremental projection method M03 with several choices of δt and examine the velocity at $(x_1, x_2) = (\ell/4, w/2)$ as a function of time for each run, as shown in Fig. 8.

For $\delta t = 10^{-5}$ s or smaller one obtains, essentially, the exact semi-discrete solution (also shown). This is rather expensive, since it requires 10^5 time steps per second of simulated time. The monolithic method M01 computes the semi-discrete solution accurately with $\delta t = 10^{-2}$ s, requiring just 100 time steps per second of simulated time.

However, if δt is increased to 10^{-4} s, for which as discussed above the (spurious) transient time $T_{\text{num}}(\delta t) \simeq 2$ s and thus comparable to the overall time scale of the process, the accuracy of the numerical solution deteriorates. The maximum velocity is underestimated by about 5% and, perhaps more importantly, the flow does not extinguish immediately after $t = 1$ s (as it should) but decays slowly instead, with a decay time of about 0.1 s. Similar unphysical behaviors are observed for $\delta t = 2 \times 10^{-4}$ s, 5×10^{-4} s and 10^{-3} s, as shown in the figure.

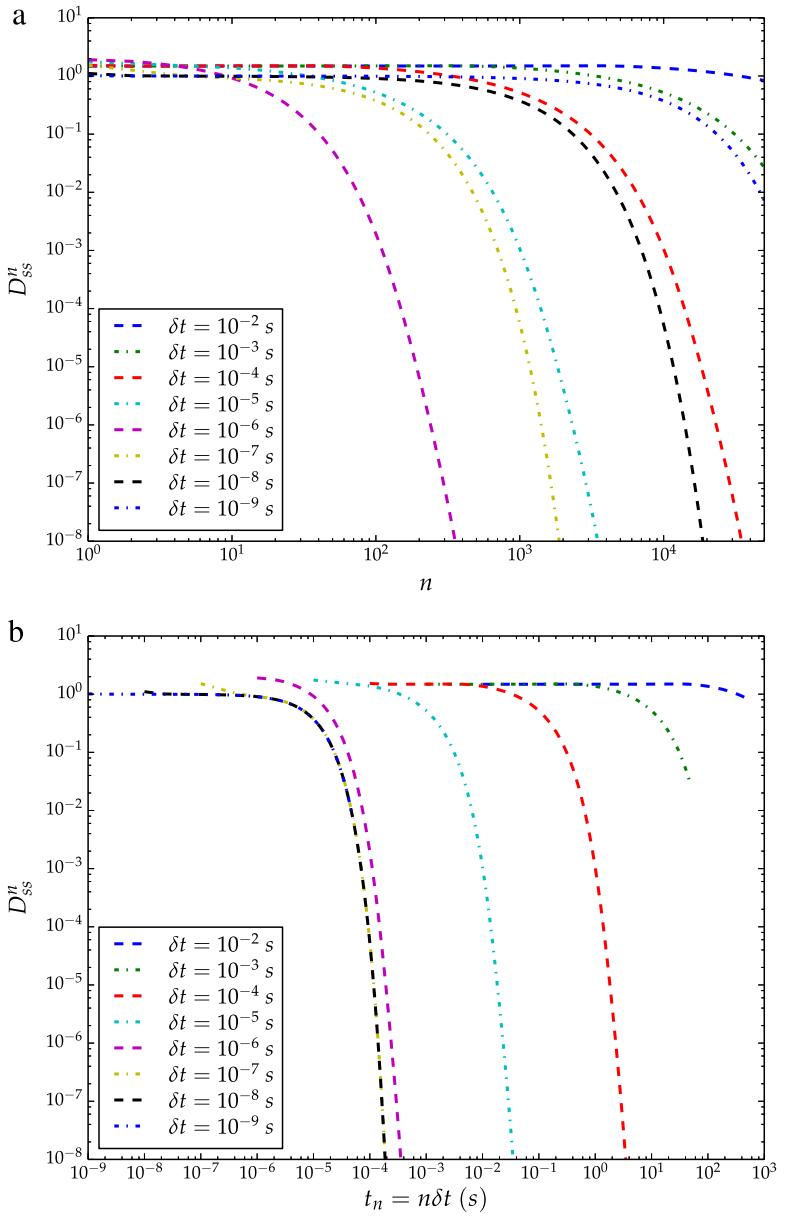


Fig. 7. Evolution of D^n_{ss} for the incremental projection method M03, for several δt: (a) as a function of the iteration number n; (b) as a function of time $t_n = n\,\delta t$ s.

Remarkably, the numerical transient is practically independent of δt for $\delta t \geq 5 \times 10^{-3}$ s. This "large-δt" regime occurs when $T_{\text{num}}(\delta t) \gg T_{\text{proc}}$ and yields a spurious transient solution which differs widely from the exact one (40% error in the maximum velocity, for example).

It is important to point out that the spurious transients being discussed here, which arise from the incremental velocity–pressure segregation, *do not depend on the mesh size*. In Fig. 9 we compare the results to those corresponding to a ten-fold mesh refinement in the y-direction (labeled as "$h/10$" in the figure) to illustrate this.

Remark 5. If this simulation were initially performed with δt chosen as 10^{-2} s or greater, a tricky situation would appear in which the numerical results are independent of δt for *moderate* reductions of it, such as divisions of δt by a factor of two or four. This could trick the user into believing that the numerical results are correct, which is obviously not true. By analyzing a large set of runs, not included here for brevity, we have developed a heuristic criterion to

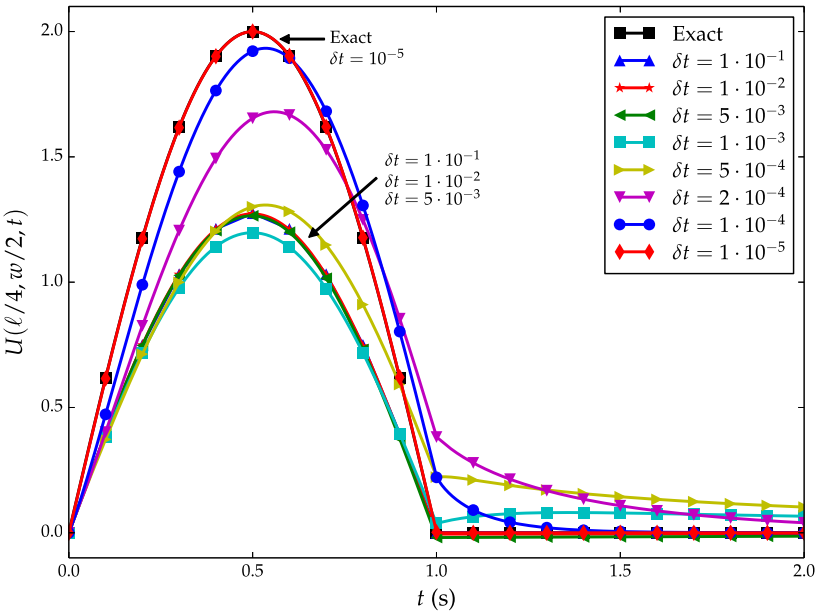


Fig. 8. Time evolution of the numerical solution for the velocity at $(x_1, x_2) = (\ell/4, w/2)$, obtained with method M03, for several δt. The exact semi-discrete solution is also shown as a reference.

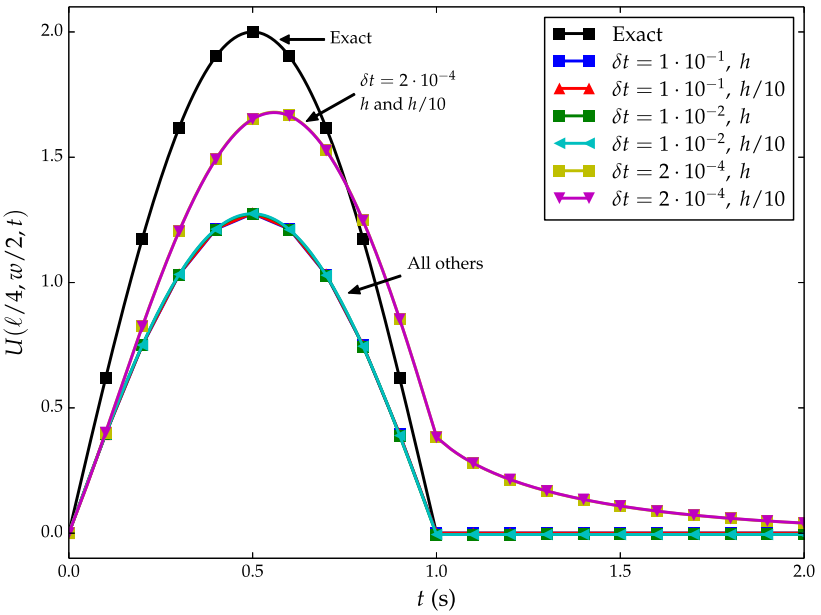


Fig. 9. Time evolution of the numerical solution for the velocity at $(x_1, x_2) = (\ell/4, w/2)$, obtained with method M03, for several δt and vertical mesh sizes h and $h/10$. Notice the invariance of the spurious transient under mesh refinement.

choose δt such that the spurious transient that develops does not significantly affect the evolution of a system with time scale T_{proc} (assumed $\gg T_\nu$). It reads

$$\delta t \leq \delta t^{**} = \frac{\sqrt{T_\nu \, T_{\text{proc}}}}{48} \tag{93}$$

and is less restrictive than (78), since it allows the appearance of spurious transients as long as their timespan is much shorter than T_{proc}. In the example above the heuristic formula gives $\delta t^{**} = 10^{-4}$ s.

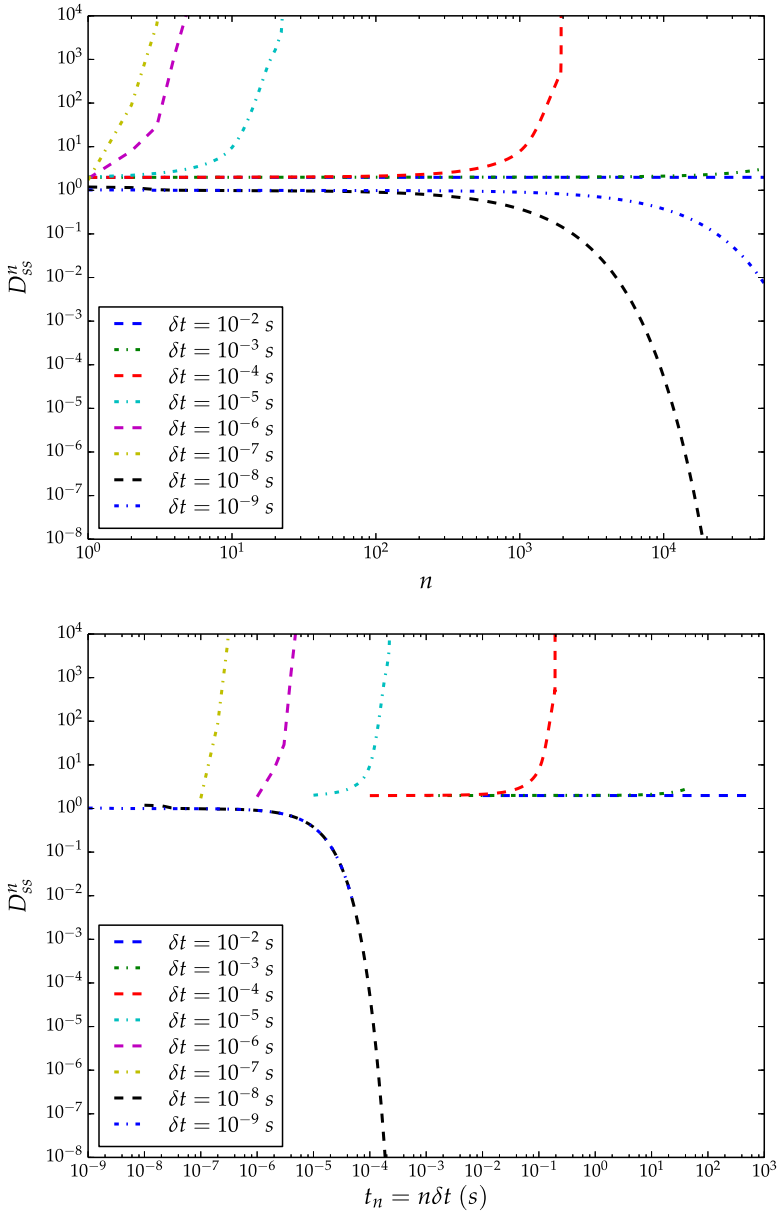


Fig. 10. Evolution of D^n_{ss} for the incremental method M05, for several δt: (a) as a function of the time label number n; (b) as a function of time $t_n = n\,\delta t$.

Going back to the steady-state calculations, let us show in Figs. 10–12 results analogous to those in Fig. 7 for the *incremental versions* of the other methods considered, namely Perot's second order method (M05), Yosida method (M07) and the pseudo-exact factorization method (M09).

The incremental version of Perot's 2nd order method (M05), as shown in Fig. 10, simply diverges for any $\delta t \geq 10^{-7}$. This tells us that this method, in both its non-incremental and incremental versions, suffers from the stability restriction $\delta t \leq \delta t_{\text{exp}}$ for all values of θ.

The results of the incremental Yosida method M07 in Fig. 11 are very similar to those of M03. The same kind of spurious transients appears, with the same timespan and intensity. In fact, when tested on the transient problem of Figs. 8 and 9, indistinguishable results are obtained. These additional tests are not detailed here for the sake of brevity.

The spurious transients exhibited by methods M03 and M07 are, in our opinion, equivalent to those described in the thin-film analogy and thus roughly independent of the mesh and of the splitting method. They seem to be inherent

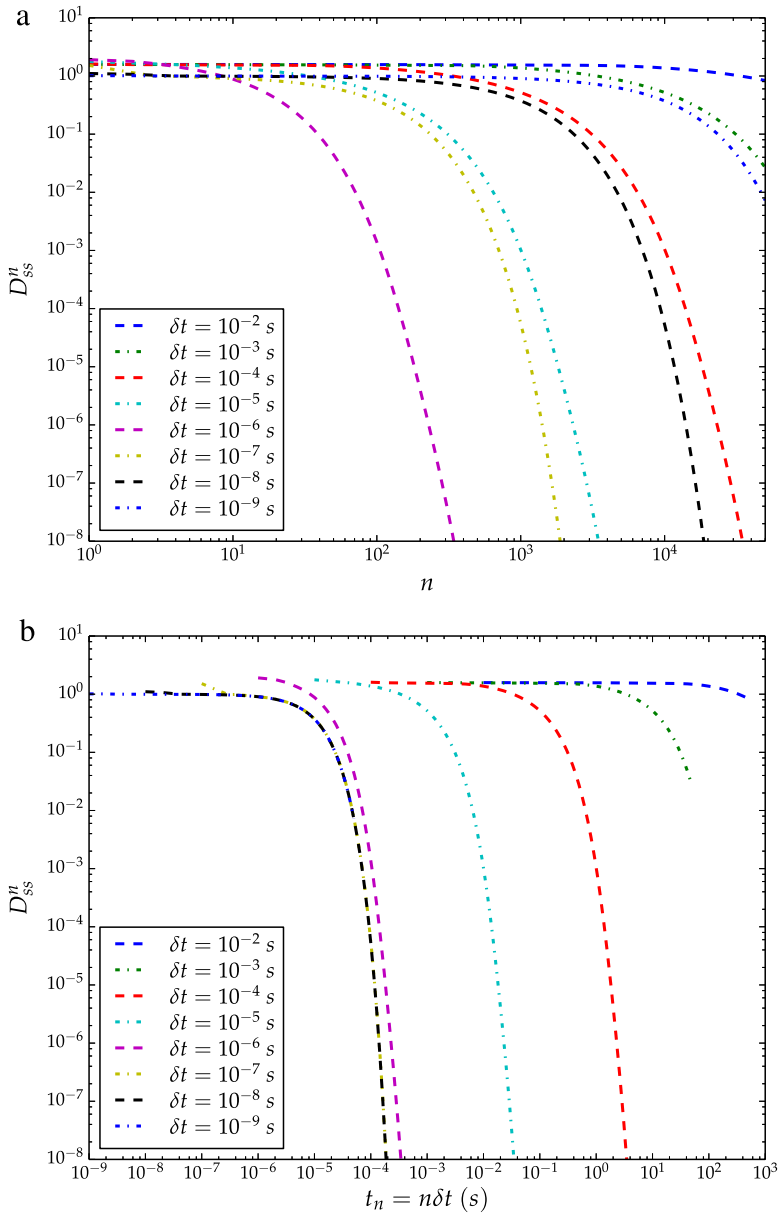


Fig. 11. Evolution of D^n_{ss} for the incremental method M07, for several δt: (a) as a function of the time label number n; (b) as a function of time $t_n = n\,\delta t$.

to the incremental decoupling of pressure and velocity, and they appear in all incremental methods if $\delta t > \delta t^*$, unless the method becomes unstable.

The incremental version of the pseudo-exact factorization method (M09) is another example of instability. It diverges exponentially for values of time step greater than 10^{-6} s. This behavior, similar to that of method M05, is certainly worse than any spurious transient.

Finally, the results of the exact factorization method (M10) are presented in Fig. 13, from which one can notice the similarity with the monolithic results from Fig. 6 and thus the total absence of spurious transients.

3.5. Spurious transient in microflow application

Let us illustrate the phenomenon of spurious transients in the calculation of the steady state of a more realistic, though academic, example. We performed low *Re* number simulations in a domain involving one contraction and one

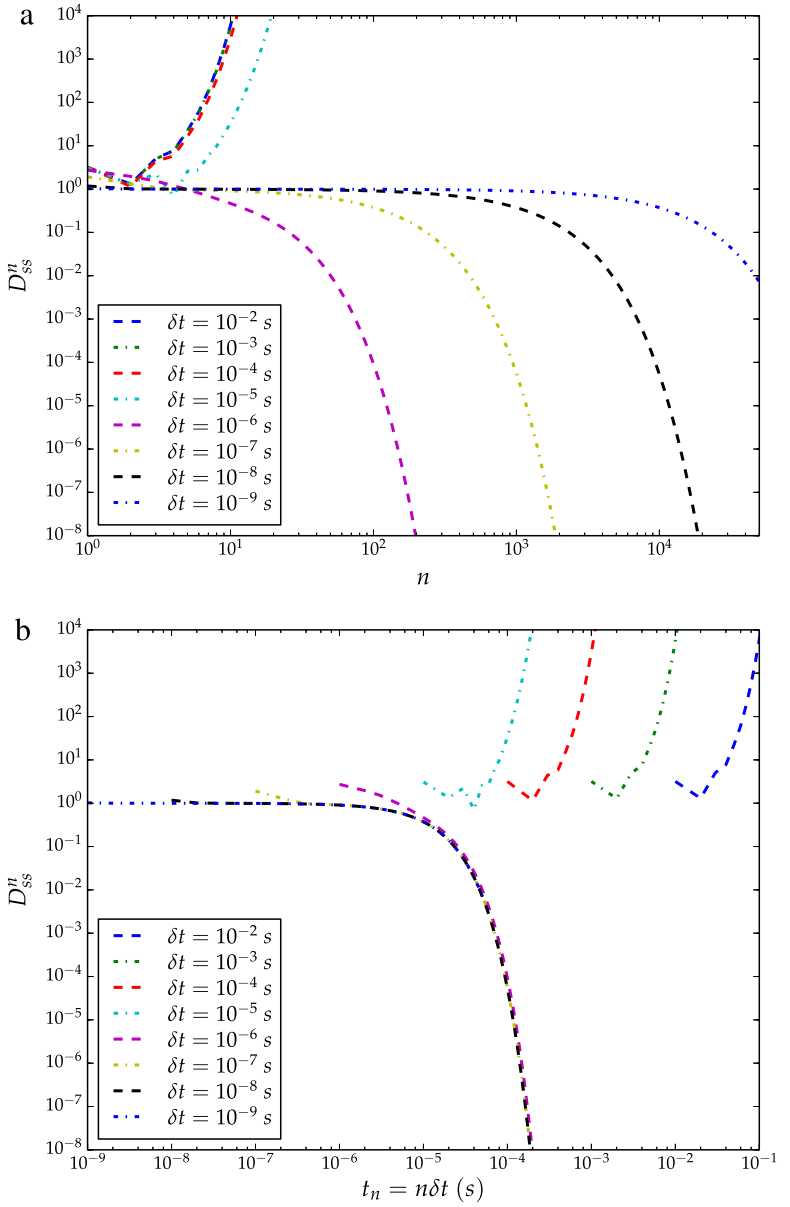


Fig. 12. Evolution of D^n_{ss} for the incremental method M09, for several δt: (a) as a function of the time label number n; (b) as a function of time $t_n = n\,\delta t$.

expansion, as detailed in Fig. 14. Microchannels with corner edges as the one considered have been used in several studies, such as nonlinear dynamical problems of viscoelastic flows [43–46].

The problem consists of a long 2D channel, with a contraction that narrows down the fluid passage, followed by a sudden expansion. This is commonly known as a contraction/expansion geometry. Being L a reference measure of the channel with size $\ell \times w$, we set $w = 4L$ and $\ell = 17L$, with the sudden contraction placed at the middle of the channel (see Fig. 14). As in previous simulations, this problem models water flowing in a microscale channel, by choosing $L = 10^{-5}$ m, with a reference velocity of $W = 10^{-3}$ m/s, resulting in $Re = LW/\nu = 10^{-2}$. Boundary conditions are standard no-slip everywhere, except at inflow, where a parabolic profile is imposed, and at outflow, with a standard outflow boundary condition ($p = 0$ and $\partial_n \mathbf{u}_n = 0$).

To illustrate the appearance of the spurious transient for this problem, we employ the popular incremental projection method in differential form (whose algebraic version is referred previously as M03) [7,32,47].

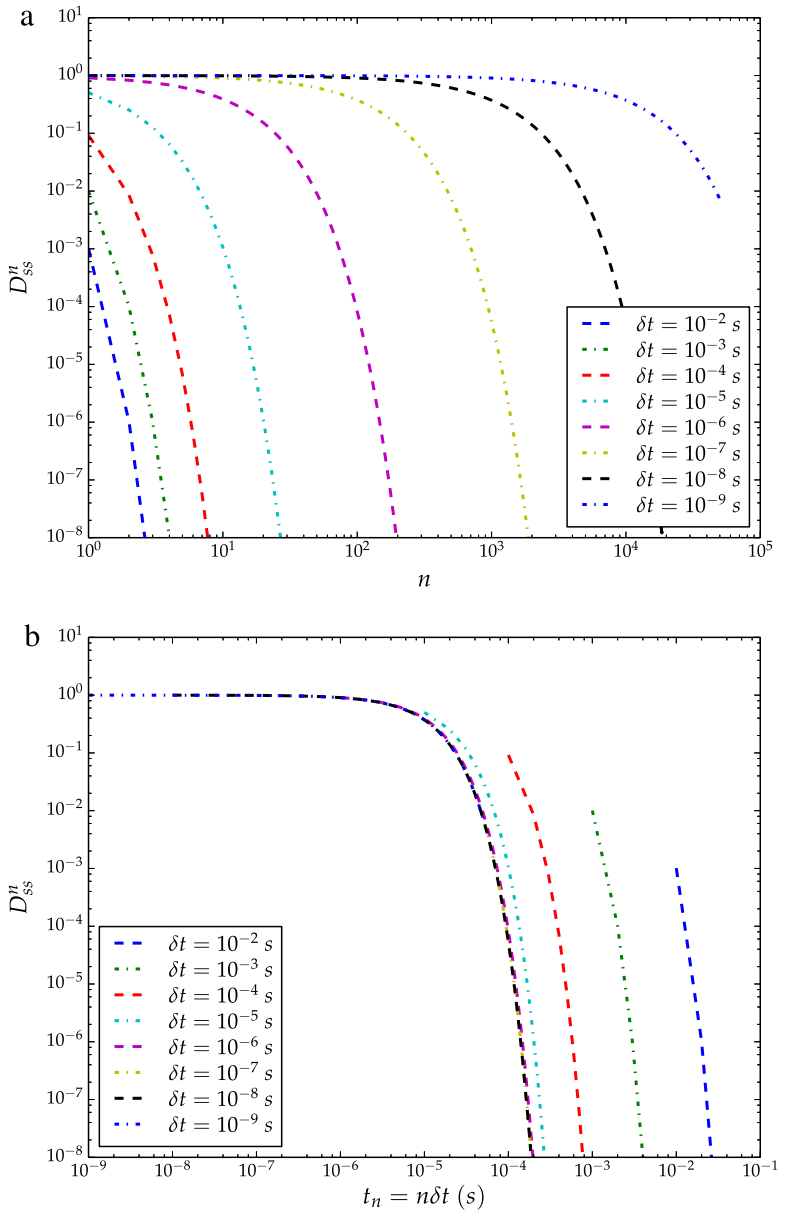


Fig. 13. Evolution of D^n_{ss} for the exact factorization method M10, for several δt: (a) as a function of the time label number n; (b) as a function of time $t_n = n\,\delta t$.

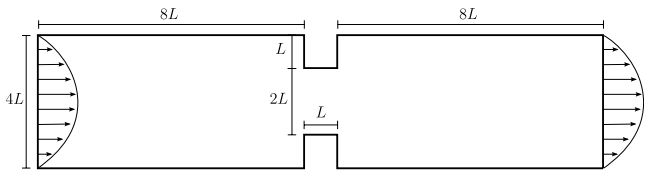


Fig. 14. Domain for the contraction/expansion example. We impose parabolic inflow and standard open outflow boundary conditions.

As before, spatial derivatives are computed by standard staggered finite difference method, and time derivatives discretized with *Backward Euler*. The Poisson equation appearing in the correction step is then solved by the *Conjugate Gradient* method, with a relative residual tolerance of 10^{-10}.

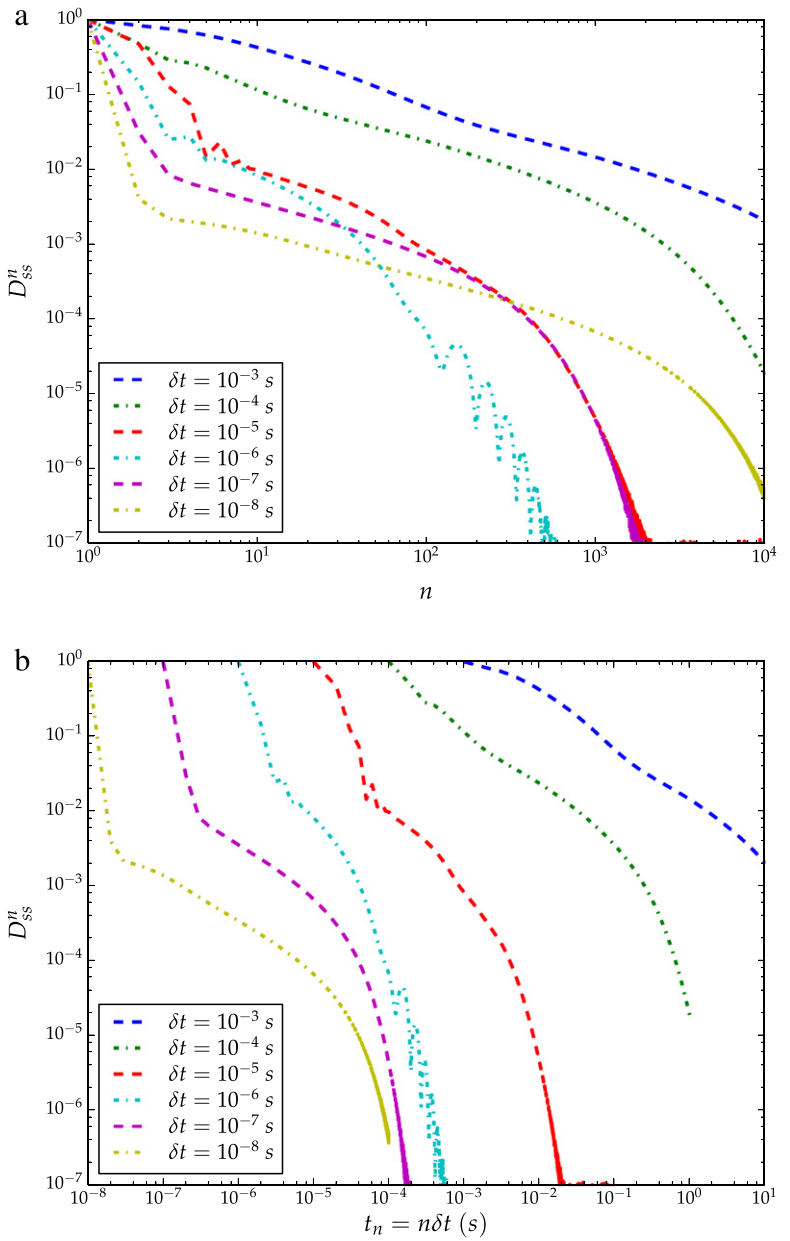


Fig. 15. Evolution of D^n_{ss} for the incremental projection method applied to the microflow obstacle test: (a) as a function of the iteration number n; (b) as a function of time t_n.

The steady state solution (computed to 10^{-10} accuracy in the residuals) was used as reference to compute the distance to steady state D^n_{ss}, defined in Eq. (92). Results are presented in Fig. 15, plotted against the iteration count (part a) and simulation time (part b). One observes the same behavior captured in the closed loop channel presented in previous sections.

For $\delta t \leq 10^{-6}$ s the transient lasts about 10^{-4} s, which is an estimate of the momentum-diffusion time T_ν in this problem. The lack of spurious transients is consistent with the criterion (78), since $\delta t^* = T_\nu/48 \simeq 2 \times 10^{-6}$ s.

For $\delta t > \delta t^*$ the spurious transient phenomenon becomes evident, with $T_{\text{num}} \simeq 10^{-2}$ s for $\delta t = 10^{-5}$ s and $T_{\text{num}} \simeq 1$ s for $\delta t = 10^{-4}$ s. Note from Fig. 15(a) that the number of iterations until steady state is minimal if $\delta t = 10^{-6}$ s, which corresponds to $\delta t \simeq \delta t^*$.

4. Concluding remarks

In this work a wide variety of projection methods for incompressible flows has been discussed and numerically assessed. The methods are obtained by algebraically splitting the monolithic method, and are defined by two approximate inverses, B_1 and B_2, of the velocity matrix, together with a parameter γ which is zero for non-incremental schemes and one for incremental ones. Methods with first, second and third order time accuracy were identified, going from the well-known first-order projection scheme to the more sophisticated incremental Yosida scheme (with BDF3 time integration).

Further, the evolution matrices of all methods were computed. From them, their numerical transients were predicted and later confirmed by numerical assessment. One observes several behaviors:

- Non-incremental methods require extremely small time steps because their results are very inaccurate if δt is greater than the stability limit of explicit methods, $\delta t_{\text{exp}} = h^2/(4\nu)$. A numerical example was used to illustrate the large impact of this fact in steady, low Reynolds number flows. This difficulty was already well-known for the first-order projection scheme [41]. Our tests show that the same happens with all (stable) non-incremental formulations.
- Incremental methods do not introduce splitting errors in the numerical steady states. Their behavior in transient problems is however strongly dependent on δt:
  - Some schemes (such as the incremental version of Perot's second order method M05 and the incremental version of the pseudo-exact factorization method M09) become unstable as δt is increased beyond a certain limit (of the order of the explicit stability limit). This loss of stability takes place for all values of θ, even if the viscous operator is treated implicitly ($\theta = 1$).
  - Some schemes (such as the popular incremental projection method M03 and the incremental version of Yosida scheme M07) remain stable for all δt, but develop a remarkable *spurious transient* which is *independent of the mesh size* and may last much longer than the momentum-diffusion time T_ν. A criterion to select the time step so as to avoid spurious transients was developed, namely
    $$\delta t \leq \delta t^* = \frac{T_\nu}{48}.$$
  - If the simulated process has an intrinsic time scale T_{proc} much greater than T_ν, it is possible to allow for some spurious numerical transients as long as they do not interfere with the overall process evolution. The criterion for selecting δt in these situation has been proposed as
    $$\delta t \leq \delta t^{**} = \frac{\sqrt{T_\nu\, T_{\text{proc}}}}{48}.$$

The spurious transient phenomenon in incremental projection methods has not previously been reported in the literature. For this reason, two additional confirmations were included. The first one showed that the phenomenon is not restricted to spatially-discretized problems, as it also appears when incremental pressure segregation is employed in a continuous model (a thin film analogy). The second confirmation consisted of numerical computations on a more realistic, obstructed channel using the classical incremental projection method at Reynolds number 10^{-2}. Runs with various δt clearly show the development of spurious transients for $\delta t > \delta t^*$.

The overall message for practitioners is that projection methods should only be used in microfluidic simulations in incremental form and choosing the time step according to the recommendations above, which can be quite stringent. The inadequacy of time steps greater than δt^{**} should weigh against choosing projection methods, despite the low cost of each time step.

As a final remark let us first stress that methods such as SIMPLE or PISO [48–50], which iterate between the discrete momentum equation and the discrete incompressibility condition until both are satisfied, are *monolithic* methods in the framework considered in this article. They are thus not expected to suffer from the spurious transient phenomenon, since the algebraic system (4) is satisfied, to within some tolerance, at the end of each time step.

Acknowledgments

The authors gratefully acknowledge Brazilian agencies FAPESP (Fundação de Amparo a Pesquisa do Estado de São Paulo), grants 2011/00538-5, 2009/15892-9, 2012/14481-8; and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), grants 309139/2012-0, 309514/2013-4, 308728/2013-0, 473589/2013-3.

Appendix. A heterogeneous thin-film problem

When the incremental scheme of Section 3.2.2 is particularized to a channel with just two values of σ, namely

$$\sigma(x) = \begin{cases} \sigma_1 & \text{if } 0 < x < \beta\,\ell \\ \sigma_2 = m\,\sigma_1 & \text{if } \beta\,\ell < x < \ell \end{cases}$$

then the function $\psi^n(x)$ takes just two values, depending on whether x is smaller or greater than $\beta\ell$, that will be denoted by ψ_1^n and ψ_2^n, respectively. Notice that m indicates the strength of the heterogeneity, with $m = 1$ corresponding to a homogeneous channel. Let us also introduce the non-dimensional time step

$$\tau = \sigma_1\,\delta t. \tag{94}$$

The system then evolves to the steady state according to the following equation:

$$\begin{pmatrix} \varphi^{n+1} \\ \psi_1^{n+1} \\ \psi_2^{n+1} \end{pmatrix} = \begin{pmatrix} \dfrac{\beta}{1+\tau} + \dfrac{1-\beta}{1+m\tau} & \dfrac{\beta\tau}{1+\tau} & \dfrac{(1-\beta)m\tau}{1+m\tau} \\ \left(\dfrac{1}{1+\tau} - \dfrac{1}{1+m\tau}\right)(1-\beta) & \dfrac{(1-\beta)\tau}{1+\tau} & -\dfrac{(1-\beta)m\tau}{1+m\tau} \\ -\left(\dfrac{1}{1+\tau} - \dfrac{1}{1+m\tau}\right)\beta & -\dfrac{\beta\tau}{1+\tau} & \dfrac{\beta\,m\tau}{1+m\,\tau} \end{pmatrix} \begin{pmatrix} \varphi^n \\ \psi_1^n \\ \psi_2^n \end{pmatrix}. \tag{95}$$

Denoting (again) by S the evolution matrix that appears in (95), the evolution to steady state is dictated by the amplification factor Λ, *defined as the spectral radius of* S. In Fig. 16 we plot Λ as a function of the non-dimensional time step size τ for several values of m and β. We consider a channel with an obstruction ($m \geq 1$) between $x = \beta\ell$ and $x = \ell$. The chosen values for m correspond to obstructions that are "mild" ($m = 1.5$), "moderate" ($m = 5$) and "strong" ($m = 100$). Concerning the values of β, we consider obstructions that are "extended" ($\beta = 0.6$), "localized" ($\beta = 0.9$) and "highly localized" ($\beta = 0.99$).

All the corresponding curves, together with the one corresponding to a homogeneous channel ($m = 1$, β irrelevant), have $\Lambda(\tau) \geq 0.5$, so that convergence to the steady state never occurs at a rate faster than 0.5^n. Considering for example, as is usual in practice, that the steady state has been attained when the residual has been reduced to 10^{-6} its initial value, at least $n = 20$ time steps are needed, and such a low number only occurs if the channel is homogeneous or mildly heterogeneous and the time step is chosen exactly as $\tau = 1$.

If the channel is moderately or strongly heterogeneous the situation worsens, all the more so if the obstruction is highly localized. For the cases with $m = 5$ the minimal values of Λ are 0.54 (if $\beta = 0.6$), 0.65 (if $\beta = 0.9$) and 0.68 (if $\beta = 0.99$). This latter case would require 36 time steps to attain steady state, and this would only happen if τ is taken as approximately 0.5. If τ is taken as 1 as in the homogeneous case, then $\Lambda \simeq 0.83$ implying that *the steady state is only attained after* 75 *time steps*!

The worst of the cases considered corresponds to $m = 100$ and $\beta = 0.99$. In this case the minimum value of Λ is $\simeq 0.88$, so that the minimum required number of time steps is 109, which is only achieved if τ takes one of two values: 0.044 or 0.23. In this case taking $\tau = 1$ yields $\Lambda = 0.98$, with the catastrophic consequence of needing 680 time steps to attain the steady state.

Considering now the actual *simulated time* at which the program reaches steady state, notice that taking the spatial average in (67) and solving yields

$$v(t) - v^\infty = [v(0) - v^\infty]\,e^{-\overline{\sigma}\,t},$$

so that to achieve the required tolerance of 10^{-6} the exact transient lasts, approximately, $13.8/\overline{\sigma}$. The *numerical* transient, on the other hand, can last much longer. For the worst case discussed above ($m = 100$, $\beta = 0.99$, $\tau = 1$), one has $\overline{\sigma} = 1.99\,\sigma_1$ and thus $\delta t = \tau/\sigma_1 = 1.99/\overline{\sigma}$. As already mentioned, steady state is achieved at $t = 680\,\delta t$ which corresponds to $t = 1353.2/\overline{\sigma}$. This implies that the *numerical* transient lasts about 100 times longer than the *exact* transient, justifying the adjective *spurious* that was adopted in this article.

Finally, notice that though the time-integration is implicit ($\theta = 1$) taking $\tau \gg 1$ is never a good idea (in all the considered cases it leads to $\Lambda(\tau) > 0.9$). In fact, from the curves in Fig. 16 one would suggest as general rule to take

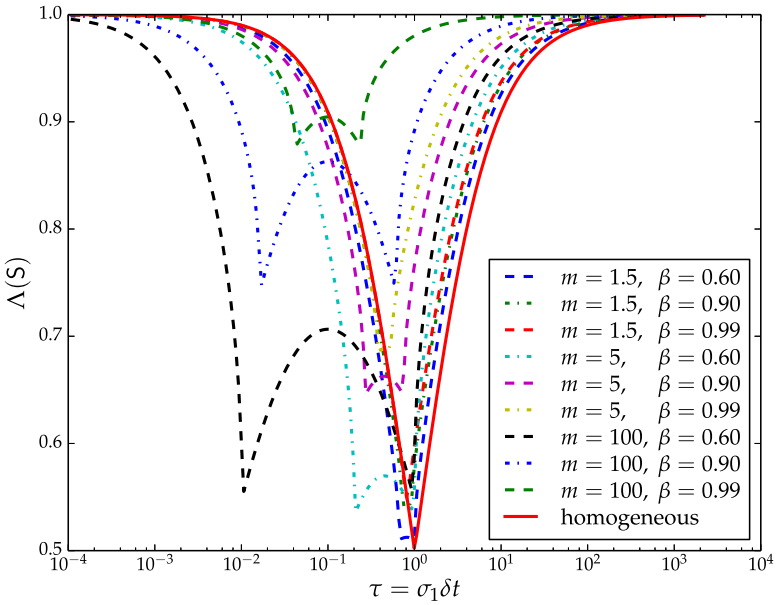


Fig. 16. Spectral radius of S as a function of $\tau = \sigma_1\, \delta t$ for several values of m and β. The case of a homogeneous channel ($m = 1$, β irrelevant) is also plotted.

$\tau \simeq 1$, or perhaps somewhat smaller ($\tau \simeq 1/4$) in channels with highly localized, strong obstructions. Going back to the definition of σ, the suggested choice $\tau \simeq 1/4$ corresponds to

$$\delta t \simeq \frac{1}{4\sigma_1} = \frac{w^2}{48\nu}$$

where w is the channel width in the unobstructed region.

References

[1] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids 8 (1965) 2182–2189.
[2] A. Chorin, A numerical method for solving incompressible viscous flow problems, J. Comput. Phys. 2 (1967) 12–26.
[3] A. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comp. 2 (1968) 745–762.
[4] R. Temam, Sur l'approximation de la solution des equations de Navier–Stokes par la methode de pas fractionnaires (II), Arch. Ration. Mech. Anal. 33 (1969) 377–385.
[5] J.B. Bell, P. Colella, H.M. Glaz, A second order projection method for the incompressible Navier–Stokes equations, J. Comput. Phys. 85 (1989) 257–283.
[6] P.M. Gresho, On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly consistent mass matrix. Part 1: theory, Internat. J. Numer. Methods Fluids 11 (1990) 587–620.
[7] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464–499.
[8] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flow, Int. J. Heat Mass Transfer 15 (1972) 1787–1806.
[9] J. van Kan, A second-order accurate pressure-correction scheme for viscous incompressible flow, SIAM J. Numer. Anal. 7 (1986) 870–891.
[10] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[11] G.E. Karniadakis, M. Israeli, S.A. Orszag, High-order splitting method for the incompressible Navier–Stokes equations, J. Comput. Phys. 97 (1991) 414–443.
[12] A. Chorin, J. Marsden, A Mathematical Introduction to Fluid Mechanics, Springer, 2000.
[13] E.Y. Tau, A second-order projection method for the incompressible Navier–Stokes equations in arbitrary domains, J. Comput. Phys. 115 (1994) 147–152.
[14] J. Shen, On error estimates of the projection methods for the Navier–Stokes equations: second-order schemes, Math. Comp. 65 (1996) 1039–1065.
[15] J.C. Strikwerda, Y.S. Lee, The accuracy of the fractional step method, SIAM J. Numer. Anal. 37 (1999) 37–47.
[16] J.L. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, J. Comput. Phys. 192 (2003) 262–276.

[17] J.L. Guermond, P. Minev, J. Shen, An overview of projection methods for incompressible flows, Comput. Methods Appl. Mech. Engrg. 195 (2006) 6011–6045.
[18] M. Deville, L. Kleiser, F.M. Rannou, Pressure and time treatment for chebyshev spectral solution of a Stokes problem, Internat. J. Numer. Methods Fluids 4 (1984) 1149–1163.
[19] J.B. Perot, Comments on the fractional-step method, J. Comput. Phys. 121 (1995) 179.
[20] M. Lee, D. Oh, Y.B. Kim, Canonical fractional-step methods and consistent boundary conditions for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 73–100.
[21] S. Armfield, R. Street, An analysis and comparison of the time accuracy of fractional-step methods for the Navier–Stokes equations on staggered grids, Internat. J. Numer. Methods Fluids 38 (2002) 255–282.
[22] P. Iannelli, F.M. Denaro, Analysis of the local truncation error in the pressure-free projection method for incompressible flows: a new accurate expression of the intermediate boundary conditions, Internat. J. Numer. Methods Fluids 42 (2003) 399–437.
[23] J.K. Dukowicz, A.S. Dvinsky, Approximate factorization as a high order splitting for the implicit incompressible flow equations, J. Comput. Phys. 102 (1992) 336–347.
[24] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[25] S. Badia, R. Codina, Algebraic pressure segregation methods for the incompressible Navier–Stokes equations, Arch. Comput. Methods Eng. 15 (2008) 343–369.
[26] A. Quarteroni, F. Saleri, A. Veneziani, Factorization methods for the numerical approximation of Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 188 (2000) 505–526.
[27] R. Codina, S. Badia, On some pressure segregation methods of fractional-step type for the finite element approximation of incompressible flow problems, Comput. Methods Appl. Mech. Engrg. 195 (2006) 2900–2918.
[28] W. Chang, F. Giraldo, J.B. Perot, Analysis of an exact fractional step method, J. Comput. Phys. 180 (2002) 183–199.
[29] B.E. Griffith, An accurate and efficient method for the incompressible Navier–Stokes equations using the projection method as a preconditioner, J. Comput. Phys. 228 (2009) 7565–7595.
[30] H.C. Elman, Preconditioning strategies for models of incompressible flow, J. Sci. Comput. 25 (2005) 347–366.
[31] R.F. Hanby, D.J. Sylvester, J.W. Chew, A comparison of coupled and segregated iterative solution techniques for incompressible swirling flows, Internat. J. Numer. Methods Fluids 22 (1996) 353–373.
[32] M. Liu, Y.X. Ren, H. Zhang, A class of fully second order accurate projection methods for solving the incompressible Navier–Stokes equations, J. Comput. Phys. 200 (2004) 325–346.
[33] P. Gervasio, F. Saleri, A. Veneziani, Algebraic fractional-step schemes with spectral methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 214 (2006) 347–365.
[34] M.O. Henriksen, J. Holmen, Algebraic splitting for incompressible Navier–Stokes equations, J. Comput. Phys. 175 (2002) 438–453.
[35] A. Quarteroni, F. Saleri, A. Veneziani, Analysis of the Yosida method for incompressible Navier–Stokes equations, J. Math. Pures Appl. 78 (5) (1999) 473–503.
[36] K.K.Q. Zhang, W.J. Minkowycz, F. Mashayek, Exact factorization technique for numerical simulations of incompressible Navier–Stokes flows, Int. J. Heat Mass Transfer 49 (2006) 535–545.
[37] J. Shen, A remark on the projection-3 method, Internat. J. Numer. Methods Fluids 16 (1993) 249–253.
[38] S. Badia, R. Codina, Pressure segregation methods based on a discrete pressure Poisson equation. An algebraic approach, Internat. J. Numer. Methods Fluids 56 (2008) 351–382.
[39] H. Owen, R. Codina, A third-order velocity correction scheme obtained at the discrete level, Internat. J. Numer. Methods Fluids 69 (2012) 57–72.
[40] C.D. Meyer, Matrix Analysis and Applied Linear Algebra, SIAM, 2000.
[41] R. Rannacher, On Chorin's projection for incompressible Navier–Stokes equations, Lecture Notes Math. 1530 (1992) 167–183.
[42] R.J. LeVeque, Finite Difference Methods for Ordinary and Partial Differential Equations, SIAM, 2007.
[43] J.P. Rothstein, G.H. McKinley, The axisymmetric contraction-expansion: the role of extensional rheology on vortex growth dynamics and the enhanced pressure drop, J. Non-Newton. Fluid Mech. 98 (2001) 33–63.
[44] J.S. Yoon, Y. Kwon, Finite element analysis of viscoelastic flows in a domain with geometric singularities, Korea-Aust. Rheol. J. 17 (2005) 99–110.
[45] Y. Kwon, Numerical description of elastic flow instability and its dependence on liquid viscoelasticity in planar contraction, J. Rheol. 56 (2012) 1335–1362.
[46] A.M. Afonso, P.J. Oliveira, F.T. Pinho, M.A. Alves, Dynamics of high-Deborah-number entry flows: a numerical study, J. Fluid Mech. 677 (2012) 272–304.
[47] C.M. Oishi, F.P. Martins, M.F. Tomé, M.A. Alves, Numerical simulation of drop impact and jet buckling problems using the eXtended Pom-Pom model, J. Non-Newton. Fluid Mech. 169–170 (2012) 91–103.
[48] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, Taylor and Francis, 1980.
[49] C.A.J. Fletcher, Computational Techniques for Fluid Dynamics, Vol. 2: Specific Techniques for Different Flow Categories, second ed., Springer-Verlag, 1991.
[50] J.H. Ferziger, M. Perić, Computational Methods for Fluid Dynamics, third edition, Springer, 2002.