Available online at www.sciencedirect.com
SCIENCE DIRECT®
Comput. Methods Appl. Mech. Engrg. 195 (2006) 3087–3109
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

Improving the speed and accuracy of projection-type incompressible flow solvers

Rainald Löhner [a,*], Chi Yang [a], Juan Cebral [a], Fernando Camelli [a], Orlando Soto [b], Jacob Waltz [c]

[a] School of Computational Sciences M.S. 4C7, George Mason University, Fairfax, VA 22030-4444, USA
[b] Advanced Technology Group SAIC, McLean, VA 22102, USA
[c] Applied Physics Division Los Alamos National Laboratory, Los Alamos, NM 87545, USA

Received 12 January 2004; received in revised form 29 June 2004; accepted 10 July 2004

Abstract

Superseding so-called first-generation incompressible flow solvers of the projection type (based on Taylor–Galerkin advection, second-order pressure damping and element-based data structures), the current, second-generation solvers (based on high-order upwind advection, fourth-order pressure damping and edge-based data structures) have now been in use for half a decade and have proven remarkably robust and efficient for many large-scale problems. In order to achieve higher accuracy and speed, these solvers have recently been enhanced in a variety of ways: (a) substepping for advection, (b) implicit treatment of advective terms via SGS and GMRES-LU-SGS iterative solvers, (c) fully implicit, time-accurate advancement of pressure and velocities, and (d) linelet preconditioning for the pressure-Poisson equation. The combined effect of these third-generation improvements leads to speedups of the order of O(1:5−1:10), with similar or even better temporal accuracy, as demonstrated on a variety of academic and industrial problems.

Keywords: Incompressible flow solvers; Projection schemes; Multistage Runge–Kutta; LU-SGS; Linelet preconditioning; FEM; CFD

1. Introduction

Among the flows that are of importance and interest to mankind, the category of low Mach-number or incompressible flows is by far the largest. Most of the manufactured products we use on a daily basis will start their life as an incompressible flow (polymer extrusion, melts, a large number of food products, etc.). The air which surrounds us can be considered, in almost all instances, as an incompressible fluid (airplanes

* Corresponding author. Tel.: +1 703 993 1990.
E-mail address: rlohner@gmu.edu (R. Löhner).

0045-7825/$ - see front matter
doi:10.1016/j.cma.2004.07.058

flying at low Mach-numbers, flows in and around cars, vans, buses, trains and buildings). The same applies to water (ships, submarines, torpedoes, pipes, etc.) and most biomedical liquids (e.g. blood). Given this large number of possible applications, it is not surprising that numerical methods to simulate incompressible flows have been developed for many years, as evidenced by an abundance of literature [13,64,25,27]. The equations describing incompressible, Newtonian flows may be written as

$$\mathbf{v}_{,t} + \mathbf{v}\nabla\mathbf{v} + \nabla p = \nabla\mu\nabla\mathbf{v}, \tag{1}$$

$$\nabla \cdot \mathbf{v} = 0. \tag{2}$$

Here p denotes the pressure, $\mathbf{v}$ the velocity vector and both the pressure p and the viscosity μ have been normalized by the (constant) density ρ. By taking the divergence of Eq. (1) and using Eq. (2) we can immediately derive the so-called pressure-Poisson equation

$$\nabla^2 p = -\nabla \cdot \mathbf{v}\nabla\mathbf{v}. \tag{3}$$

What sets incompressible flow solvers apart from compressible flow solvers is the fact that the pressure is not obtained from an equation of state of the form $p = p(\rho, T)$, but from the divergence constraint. This implies that the pressure field establishes itself instantaneously (reflecting the infinite speed of sound assumption of incompressible fluids) and must therefore be integrated implicitly in time.

The remainder of the paper is organized as follows: Section 2 describes the basic elements of the solver, in particular the main design criteria. Section 3 treats the spatial discretization, with particular emphasis on advection and divergence terms. This is followed in Section 4 by the temporal discretization chosen. Sections 5 and 6 are devoted to iterative solvers and the acceleration to steady state. Section 7 treats fully implicit timestepping. Several examples are given in Section 8. Finally, in Section 9 some conclusions are drawn and an outlook for future work is summarized.

2. Basic elements of the solver

Since over a decade [42,48,51,43] the numerical schemes chosen to solve the incompressible Navier–Stokes equations given by Eqs. (1) and (2) have been based on the following criteria:

- Spatial discretization using *unstructured grids* (in order to allow for arbitrary geometries and adaptive refinement);
- Spatial approximation of unknowns with *simple finite elements* (in order to have a simple input/output and code structure);
- Temporal approximation using *implicit integration of viscous terms and pressure* (the interesting scales are the ones associated with advection);
- Temporal approximation using *explicit integration of advective terms*;
- *Low-storage, iterative solvers* for the resulting systems of equations (in order to solve large 3-D problems); and
- Steady results that are *independent from the timestep* chosen (in order to have confidence in convergence studies).

3. Spatial discretization

As stated before, we desire a spatial discretization with unstructured grids in order to:

- Approximate arbitrary domains, and
- Perform adaptive refinement in a straightforward manner, i.e. without changes to the solver.

From a numerical point of view, the difficulties in solving Eqs. (1)–(3) are the usual ones. First-order derivatives are problematic (overshoots, oscillations, instabilities), while second-order derivatives can be discretized by a straightforward Galerkin approximation. We will first treat the advection operator and then proceed to the divergence operator. Given that for tetrahedral grids solvers based on edge data structures incur a much lower indirect addressing and CPU overhead than those based on element data structures [44], only these will be considered.

3.1. The advection operator

It is well known that a straightforward Galerkin approximation of the advection terms will lead to an unstable scheme (recall that on a 1-D mesh of elements with constant size, the Galerkin approximation is simply a central difference scheme). Three ways have emerged to modify (or stabilize) the Galerkin discretization of the advection terms:

- Integration along characteristics [22],
- Taylor–Galerkin (or streamline diffusion) [39,6,15] and
- Edge-based upwinding [43].

Of these, we only consider the third option here. The Galerkin approximation for the advection terms yields a right-hand side (RHS) of the form:

$$r^i = D^{ij}\mathscr{F}_{ij} = D^{ij}(\mathbf{f}_i + \mathbf{f}_j), \tag{4}$$

where the $\mathbf{f}_i$ are the 'fluxes along edges'

$$\mathbf{f}_i = S_k^{ij}\mathbf{F}_i^k, \quad S_k^{ij} = \frac{d_k^{ij}}{D^{ij}}, \quad D^{ij} = \sqrt{d_k^{ij} d_k^{ij}}, \tag{5}$$

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j, \tag{6}$$

$$\mathbf{f}_i = (S_k^{ij} v_i^k)\mathbf{v}_i, \quad \mathbf{f}_j = (S_k^{ij} v_j^k)\mathbf{v}_j \tag{7}$$

and the edge-coefficients are based on the shape-functions N^i as follows:

$$d_k^{ij} = \frac{1}{2}\int_\Omega (N^i_{,k} N^j - N^j_{,k} N^i)\,\mathrm{d}\Omega. \tag{8}$$

A consistent numerical flux is given by

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |v^{ij}|(\mathbf{v}_i - \mathbf{v}_j), \tag{9}$$

where

$$v^{ij} = \frac{1}{2} S_k^{ij}(v_i^k + v_j^k). \tag{10}$$

As with all other edge-based upwind fluxes, this first-order scheme can be improved by reducing the difference $\mathbf{v}_i - \mathbf{v}_j$ through (limited) extrapolation to the edge center [44].

3.2. The divergence operator

A persistent difficulty with incompressible flow solvers has been the derivation of a stable scheme for the divergence constraint Eq. (2). The stability criterion for the divergence constraint is also known as the Ladyzenskaya–Babuska–Brezzi or LBB condition [24]. The classic way to satisfy the LBB condition has been to use different functional spaces for the velocity and pressure discretization [18]. Typically, the

velocity space has to be richer, containing more degrees of freedom than the pressure space. Elements belonging to this class are the p1/p1+ bubble mini-element [60], the p1/iso-p1 element [64] and the p1/p2 element [62]. An alternative way to satisfy the LBB condition is through the use of artificial viscosities [42], 'stabilization' [19,63,20] or a 'consistent numerical flux' (more elegant terms for the same thing). The equivalency of these approaches has been repeatedly demonstrated (e.g., [60,42,44]). The approach taken here is based on consistent numerical fluxes, as it fits naturally into the edge-based framework. For the divergence constraint, the Galerkin approximation along edge i, j is given by

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j, \quad \mathbf{f}_i = S_k^{ij} v_i^k, \quad \mathbf{f}_j = S_k^{ij} v_j^k. \tag{11}$$

A consistent numerical flux may be constructed by adding pressure terms of the form:

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}|(p_i - p_j), \tag{12}$$

where the eigenvalue λ^{ij} is given by the ratio of the characteristic advective timestep of the edge Δt and the characteristic advective length of the edge l:

$$\lambda^{ij} = \frac{\Delta t^{ij}}{l^{ij}}. \tag{13}$$

Higher-order schemes can be derived by reconstruction and limiting, or by substituting the first-order differences of the pressure with third-order differences:

$$\mathscr{F}_{ij} = \mathbf{f}_i + \mathbf{f}_j - |\lambda^{ij}|\left(p_i - p_j + \frac{l^{ij}}{2}(\nabla p_i + \nabla p_j)\right). \tag{14}$$

This results in a stable, low-diffusion, fourth-order damping for the divergence constraint.

4. Temporal discretization

As stated before, one is usually interested in physical phenomena that propagate with the advective timescales. Diffusive phenomena typically occur at a much faster rate, and should therefore be integrated implicitly. Given that the pressure establishes itself immediately through the pressure-Poisson equation, an implicit integration of pressure also seems appropriate. The hyperbolic character of the advection operator and the elliptic character of the pressure-Poisson equation have led to a number of so-called projection schemes. The key idea is to predict first a velocity field from the current flow variables without taking the divergence constraint into account. In a second step, the divergence constraint is enforced by solving a pressure-Poisson equation. The velocity increment can therefore be separated into an advective–diffusive and pressure increment:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \Delta\mathbf{v}^a + \Delta\mathbf{v}^p = \mathbf{v}^* + \Delta\mathbf{v}^p. \tag{15}$$

For an explicit integration of the advective terms, one complete timestep is given by

– *Advective–diffusive prediction*: $\mathbf{v}^n \to \mathbf{v}^*$

$$\left[\frac{1}{\Delta t} - \theta\nabla\mu\nabla\right](\mathbf{v}^* - \mathbf{v}^n) + \mathbf{v}^n \cdot \nabla\mathbf{v}^n + \nabla p^n = \nabla\mu\nabla\mathbf{v}^n. \tag{16}$$

– *Pressure correction*: $p^n \to p^{n+1}$

$$\nabla \cdot \mathbf{v}^{n+1} = 0, \tag{17}$$

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} + \nabla(p^{n+1} - p^n) = 0, \tag{18}$$

which results in

$$\nabla^2(p^{n+1}-p^n)=\frac{\nabla\cdot\mathbf{v}^*}{\Delta t}. \tag{19}$$

– *Velocity correction*: $\mathbf{v}^*\rightarrow\mathbf{v}^{n+1}$

$$\mathbf{v}^{n+1}=\mathbf{v}^*-\Delta t\nabla(p^{n+1}-p^n). \tag{20}$$

At steady-state, $\mathbf{v}^*=\mathbf{v}^n=\mathbf{v}^{n+1}$ and the residuals of the pressure correction vanish, implying that the result does not depend on the timestep Δt. θ denotes the implicitness-factor for the viscous terms ($\theta=1$: first-order, fully implicit, $\theta=0.5$: second-order, Crank–Nicholson). The forward Euler integration of the advection terms imposes rather severe restrictions on the allowable timestep. For this reason, alternative explicit integration schemes have been used repeatedly [66]. Many authors have used multilevel schemes, such as the second-order Adams–Bashforth scheme. The problem with schemes of this kind is that they use the values at the current and previous timestep, which makes them awkward in the context of adaptive refinement, moving meshes, and local or global remeshing. Single step schemes are therefore preferable. Lax–Wendroff or Taylor–Galerkin schemes offer such a possibility, but in this case the result of steady-state calculations depends (albeit weakly) on the timestep (or equivalently the Courant-number) chosen. This leads us to single step schemes whose steady-state result does not depend on the timestep are preferable.

Projection schemes of this kind (explicit advection with a variety of schemes, implicit diffusion, pressure-Poisson equation for either the pressure or pressure increments) have been widely used in conjunction with spatial discretizations based on finite differences [10,11,40,4,5,1], finite volumes [35,53], and finite elements [21–23,34,49,42,48,50,51,43,61,16,37,12,41,38]), and have served as the basis for industrial CFD solvers with both element-based data structures (first generation) and edge-based data structures (second generation) [61,16,36–38].

4.1. Temporal discretization of higher order

The scheme given by Eqs. (16)–(20) is, at best, of second-order in time. It is surprising to note that apparently no attempt has been made to use multistage explicit schemes to integrate the advective terms with higher order or to accelerate the convergence to steady-state. This may stem from the fact that the implicit integration of the viscous terms apparently impedes taking the full advantage multistage schemes offer for the Euler limit of no viscosity. An interesting alternative is to integrate with different timestepping schemes the different regimes of flows with highly variable cell Reynolds-number

$$Re_h=\frac{\rho|\mathbf{v}|h}{\mu}, \tag{21}$$

where h is the mesh size. For the case $Re_h<1$ (viscous dominated), the accuracy in time of the advective terms is not so important. However, for $Re_h>1$ (advection dominated), the advantages of higher order time-marching schemes for the advective terms are considerable, particularly if one considers vortex transport over large distances. Dahlquist's theorem states that no unconditionally stable, implicit one-step scheme can be of order higher than two (this being the Crank–Nicholson scheme). However, explicit schemes of the Runge–Kutta type can easily yield higher order timestepping. A k-step, time-accurate Runge–Kutta scheme or order k for the advective parts may be written as

$$\mathbf{v}^i=\mathbf{v}^n+\alpha^i\gamma\Delta t(-\mathbf{v}^{i-1}\cdot\nabla\mathbf{v}^{i-1}-\nabla p^n+\nabla\mu\nabla\mathbf{v}^{i-1});\quad i=1,k-1, \tag{22a}$$

$$\left[\frac{1}{\Delta t}-\theta\nabla\mu\nabla\right](\mathbf{v}^k-\mathbf{v}^n)+\mathbf{v}^{k-1}\cdot\nabla\mathbf{v}^{k-1}+\nabla p^n=\nabla\mu\nabla\mathbf{v}^{k-1}. \tag{22b}$$

Here, the α^i are the standard Runge–Kutta coefficients $\alpha^i = 1/(k+1-i)$. As compared to the original scheme given by Eq. (16), the $k-1$ stages of Eq. (22) may be seen as a predictor (or replacement) of $\mathbf{v}^n$ by $\mathbf{v}^{k-1}$. The original right-hand side has not been modified, so that at steady-state $\mathbf{v}^n = \mathbf{v}^{k-1}$, preserving the requirement that the steady-state be independent of the timestep Δt. The factor γ denotes the local ratio of the stability limit for explicit timestepping for the viscous terms versus the timestep chosen. Given that the advective and viscous timestep limits are proportional to:

$$\Delta t_a \approx \frac{h}{|\mathbf{v}|}; \quad \Delta t_v \approx \frac{\rho h^2}{\mu}, \tag{23}$$

we immediately obtain

$$\gamma = \frac{\Delta t_v}{\Delta t_a} \approx \frac{\rho |\mathbf{v}| h}{\mu} \approx Re_h, \tag{24}$$

or, in its final form:

$$\gamma = \min(1, Re_h). \tag{25}$$

In regions away from boundary layers, this factor is O(1), implying that a high-order Runge–Kutta scheme is recovered. Conversely, for regions where $Re_h = \mathrm{O}(0)$, the scheme reverts back to the original one (Eq. (16)). Note also that the very tempting option of ignoring the pressure and viscous terms in Eq. (22) leads to steady-state results that are not independent of the timestep.

Besides higher accuracy, an important benefit of explicit multistage advection schemes is the larger timestep one can employ. The increase in allowable timestep is roughly proportional to the number of stages used (and has been exploited extensively for compressible flow simulations [33]). Given that for an incompressible solver of the projection type given by Eqs. (16)–(20) most of the CPU time is spent solving the pressure-Poisson system Eq. (19), the speedup achieved is also roughly proportional to the number of stages used.

5. Iterative solvers

Both Eqs. (16) and (19) lead to large (symmetric) systems of equations of the form:

$$\mathbf{K}\mathbf{u} = \mathbf{r}. \tag{26}$$

Preconditioned conjugate gradient (PCG) solvers [54] are used to solve Eq. (19). For isotropic grids, simple diagonal preconditioning has proven very effective. For highly stretched RANS grids, linelet preconditioning has proven more effective [48,59]. We also remark that we have attempted repeatedly to use multigrid as a solver [65], but that for most cases to date the simpler, highly optimized PCG solvers have proven superior.

6. Acceleration to steady-state

For steady flows, the use of a time-accurate scheme with uniform timestep Δt in the domain will invariably lead to slow convergence. In order to obtain steady results faster, a number of possibilities can be explored. Among the many we and others have tried, the following have proven the most successful:

- Local timesteps;
- Reduced iteration for the pressure;

– Substepping for the advection terms;
– Implicit treatment of the advection terms; and
– Fully implicit treatment of advection, diffusion and pressure.

The main features of these are reviewed in the sequel.

6.1. Local timestepping

Faster convergence to steady-state may be achieved by employing local timesteps. Given that the results obtained by the schemes used do not depend on the timestep, this can be readily done. One simply defines a separate timestep for each gridpoint, and marches in time until a steady solution is reached. Local timestepping works best for problems that exhibit large variation of grid size and velocity.

6.2. Reduced pressure iterations

The most time-consuming part of projection schemes is the solution of the pressure-Poisson equation at every timestep. Recall that this is the equation that defines the pressure and establishes a divergence-free state at the next timestep. If we are only interested in the steady result, obtained after many timesteps, then the maintenance of an exact divergence-free state at every intermediate timestep is not required. Therefore, one can use a less stringent convergence criterion for the pressure-Poisson solver, saving a considerable amount of CPU-time. We have experimented extensively with this option. The results obtained have been mixed. For inviscid flows (Euler equations, isotropic grids), this option works well. However, for the more demanding high Reynolds-number cases (separation, highly anisotropic grids), it has proven difficult to define reliable convergence criteria.

6.3. Substepping for the advection terms

This option was already treated above (see Eqs. (22) and (23)). The speedup achieved is roughly proportional to the stages used.

6.4. Implicit treatment of the advection terms

Any explicit integration of the advective terms implies that information can only travel at most one element per timestep. In order to allow for a faster transfer of information and larger timesteps, the advective terms have to be integrated implicitly [58]. Eq. (16) then becomes:

$$\left[\frac{1}{\Delta t}+\mathbf{v}^*\cdot\nabla-\nabla\mu\nabla\right](\mathbf{v}^*-\mathbf{v}^n)+\mathbf{v}^n\cdot\nabla\mathbf{v}^n+\nabla p^n=\nabla\mu\nabla\mathbf{v}^n, \tag{27}$$

leading to a non-symmetric system of equations of the form:

$$\mathbf{A}\Delta\mathbf{v}=\mathbf{r}. \tag{28}$$

This may be rewritten as

$$\mathbf{A}\cdot\Delta\mathbf{v}=(\mathbf{L}+\mathbf{D}+\mathbf{U})\cdot\Delta\mathbf{v}=\mathbf{r}, \tag{29}$$

where **L**, **D**, **U** denote the lower, diagonal and upper diagonal entries of **A**. Classic relaxation schemes to solve this system of equations include:

(a) Gauss–Seidel, given by

$$(\mathbf{L}+\mathbf{D})\cdot\Delta\mathbf{v}^1=\mathbf{r}-\mathbf{U}\cdot\Delta\mathbf{v}^0, \tag{30}$$

$$(\mathbf{D}+\mathbf{U})\cdot\Delta\mathbf{v}=\mathbf{r}-\mathbf{L}\cdot\Delta\mathbf{v}^1. \tag{31}$$

(b) Lower-upper symmetric Gauss–Seidel (LU-SGS), given by

$$(\mathbf{L}+\mathbf{D})\cdot\mathbf{D}^{-1}\cdot(\mathbf{D}+\mathbf{U})\cdot\Delta\mathbf{v}=\mathbf{r}. \tag{32}$$

These relaxation schemes have been optimized over the years, resulting in very efficient compressible flow solvers [45–47,56]. Key ideas include:

– Using the spectral radius ρ_A of $\mathbf{A}$ for the diagonal entries $\mathbf{D}$; for the advection case, $\rho_A=|\mathbf{v}|$, resulting in:

$$\mathbf{D}=\left[\frac{1}{\Delta t}\mathbf{M}_l^i-0.5\sum\mathbf{C}^{ij}|\mathbf{v}|_{ij}+\sum\mathbf{k}^{ij}\right]\mathbf{I}, \tag{33}$$

where $\mathbf{C}$, $\mathbf{k}$ denote the edge coefficients for the advective and viscous fluxes and $\mathbf{M}_l^i$ the lumped mass matrix at node i;

– Replacing:

$$\mathbf{A}\cdot\Delta\mathbf{v}\approx\Delta\mathbf{F}, \tag{34}$$

resulting in:

$$\Delta\mathbf{F}=\mathbf{F}(\mathbf{v}+\Delta\mathbf{v})-\mathbf{F}(\mathbf{v}). \tag{35}$$

The combined effect of these simplifications is a family of schemes that are matrix free, require no extra storage as compared to explicit schemes, and (due to lack of limiting) per relaxation sweep are faster than conventional explicit schemes. For the LU-SGS scheme, each pass over the mesh proceeds as follows:

– Forward sweep:

$$\Delta\hat{\mathbf{v}}^i=\mathbf{D}^{-1}\left[\mathbf{r}^i-0.5\sum_{j<i}\mathbf{C}^{ij}\cdot(\Delta\hat{\mathbf{F}}_{ij}-|\mathbf{v}|_{ij}\Delta\hat{\mathbf{v}}_j)+\sum_{j<i}\mathbf{k}^{ij}\Delta\hat{\mathbf{v}}_j\right]. \tag{36}$$

– Backward sweep:

$$\mathbf{r}=\mathbf{D}\cdot\Delta\hat{\mathbf{v}} \tag{37}$$

$$\Delta\mathbf{v}^i=\mathbf{D}^{-1}\left[\mathbf{r}^i-0.5\sum_{j>i}\mathbf{C}^{ij}\cdot(\Delta\mathbf{F}_{ij}-|\mathbf{v}|_{ij}\Delta\mathbf{v}_j)+\sum_{j>i}\mathbf{k}^{ij}\Delta\hat{\mathbf{v}}_j\right]. \tag{38}$$

Luo [45] has shown that no discernable difference could be observed when taking central or upwind discretizations for $\Delta\mathbf{F}$. As the CPU requirements of upwind discretizations are much higher, all relaxation passes are carried out using central schemes. Given that the same loop structure ($\mathbf{L}$, $\mathbf{D}$, $\mathbf{U}$) is required for both the Gauss–Seidel, the LU-SGS and the GMRES matrix-vector products, it is possible to write a single 'sweep' subroutine that encompasses all of these cases. The initialization of the Gauss–Seidel loop is accomplished with an LU-SGS pass.

7. Fully implicit integration

Using the notation

$$u^\theta=(1-\theta)u^n+\theta u^{n+1}, \tag{39}$$

which implies

$$u^{n+1} - u^n = \frac{u^\theta - u^n}{\theta}, \tag{40}$$

an implicit timestepping scheme may be written as follows:

$$\frac{\mathbf{v}^\theta - \mathbf{v}^n}{\theta \Delta t} + \mathbf{v}^\theta \nabla \mathbf{v}^\theta + \nabla p^\theta = \nabla \mu \nabla \mathbf{v}^\theta, \tag{41}$$

$$\nabla \cdot \mathbf{v}^\theta = 0. \tag{42}$$

Following similar approaches for compressible flow solvers [2], this system can be interpreted as the steady-state solution of the pseudo-time system:

$$\mathbf{v}^\theta_{,\tau} + \mathbf{v}^\theta \nabla \mathbf{v}^\theta + \nabla p^\theta = \nabla \mu \nabla \mathbf{v}^\theta - \frac{\mathbf{v}^\theta - \mathbf{v}^n}{\theta \Delta t}, \tag{43}$$

$$\nabla \cdot \mathbf{v}^\theta = 0. \tag{44}$$

Observe that the only difference between Eqs. (41) and (42) and the original incompressible Navier–Stokes equations given by Eqs. (1) and (2) is the appearance of new source-terms. These source terms are pointwise dependent on the variables being integrated ($\mathbf{v}^\theta$), and can therefore be folded into the left hand side for explicit timestepping without any difficulty. The idea is then to march Eqs. (41) and (42) to steady state in the pseudo-time τ using either the explicit- or implicit-advection projection scheme.

8. Examples

The new schemes have been tested on a variety of examples, a few of which are included here. We remark from the outset that the main aim is the comparison of speed. Given that the RHS discretization is the same for the second and third generation solvers, the results will not change for steady flow cases. Detailed comparison to experiments, mesh refinement studies, etc. can be found in [43,61,16,37,28,7,36,38,8,9].

8.1. NACA0012

The first example considered is the classic NACA0012 wing at $\alpha = 5°$ angle of attack. This is a steady, inviscid case (Euler). Figs. 1a and b shows the surface mesh employed, as well as the surface pressures obtained. Although the mesh is rather coarse (`nelem=368,872, npoin=68,321`), it still allows for a meaningful comparison of the different solvers. This problem was solved using (C denotes the Courant-nr.):

- The standard explicit-advection projection scheme ($C = 0.1$) with full pressure solution every 20 timesteps, and partial pressure solution (max 20 PCG iterations) in between (Ex 1);
- The explicit-advection projection scheme with 2–5 substeps for advective prediction (Ex 2–5);
- The implicit-advection projection scheme ($C = 4.0$) with several SGS relaxations (SGS 1/2/4/10);
- A predictor-corrector scheme (not outlined above) based on the implicit-advection SGS solver (SGS 2S (01,11)); and
- The implicit-advection projection scheme ($C = 4.0$) with LU-SGS-GMRES solver for the advection, again with different number of iterations per timestep (GMRES 3/10).

Each one of these runs was considered converged when the change in lift, normalized by the Courant-number, was below $t_l = 10^{-3}$ for five subsequent timesteps. We have found such a measure to be a better

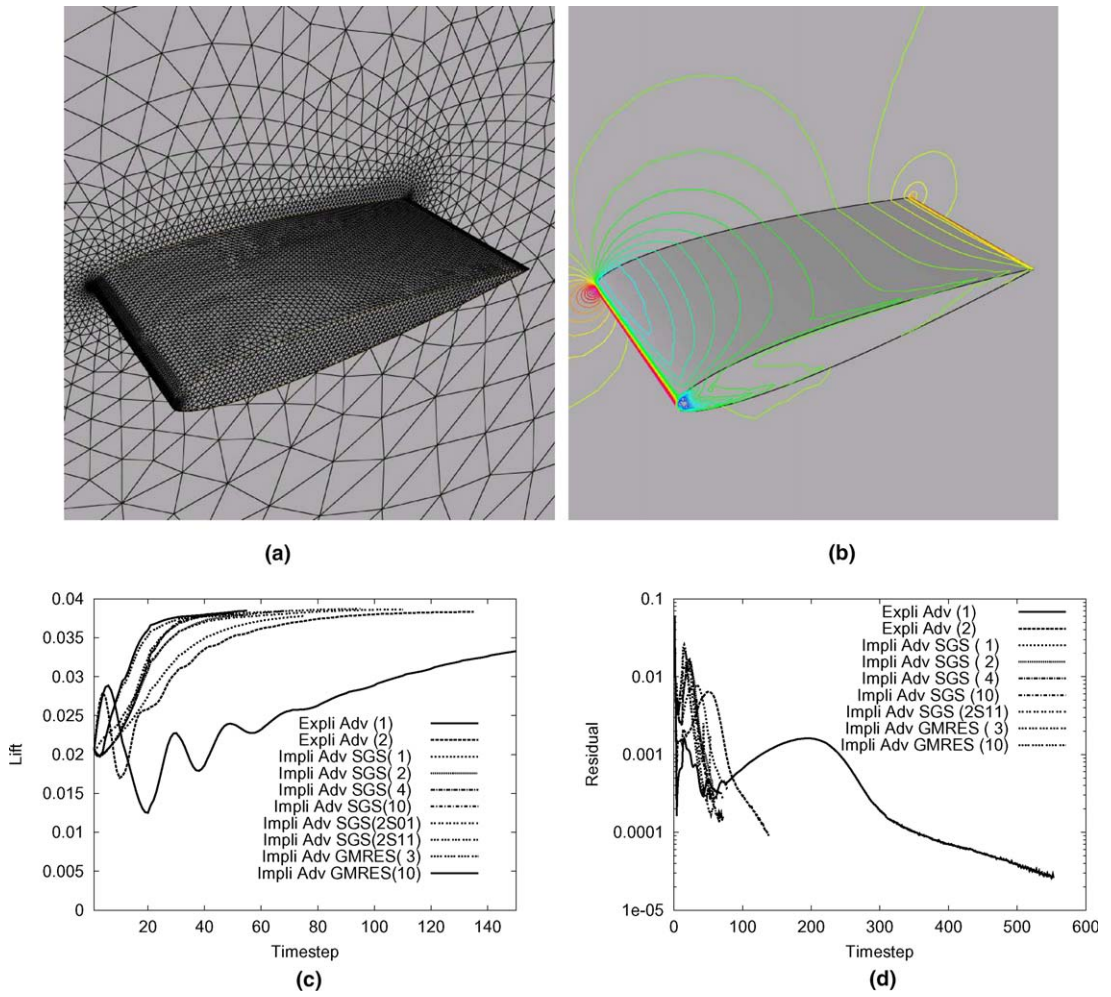


Fig. 1. (a,b) NACA0012: surface mesh and pressure, (c,d) NACA0012: convergence history for lift and residuals.

indicator of convergence than residuals, as this allows the user to state clearly what accuracy is desired. Fig. 1c and d shows the convergence history for the lift and residuals, respectively. One can see that the implicit-advection schemes converge significantly faster. The timings obtained summarized in Table 1, indicate that this faster convergence also translates into a drastic reduction in CPU requirements. Note that the 2-step explicit-advection scheme also yields a surprisingly high speedup, but falls short of the speedups obtained for the low-relaxation SGS and LU-SGS-GMRES schemes.

8.2. Wigley Hull

The second example considered is is the well known Wigley Hull, given by the analytical formula:

$$y = 0.5 \cdot B \cdot [1 - 4x^2] \cdot \left[1 - \left(\frac{z}{D}\right)^2\right],$$

Table 1
NACA-0012

Scheme	ntime	CPU [s]	Speedup
Ex 1 (0.1)	540	579	1.00
Ex 2 (0.4)	135	193	3.00
Ex 3 (0.6)	90	157	3.69
Ex 5 (0.8)	70	166	3.48
ImSGS (1)	75	172	13.46
ImSGS (2)	55	142	16.31
ImSGS (4)	70	188	12.30
ImSGS (10)	51	532	4.35
ImSGS 2 (01)	110	556	4.16
ImSGS 2 (11)	50	280	8.27
ImGMRES (3)	70	216	10.72
ImGMRES (10)	55	772	3.00

where B and D are the beam and the draft of the ship at still water. The case considered here, which had $D = 0.0625$, $B = 0.1$, has been studied before by several authors [29,17,1,30,31,43,67]. This is a steady, inviscid case (Euler) with free surface (no mesh movement, i.e. geometric free surface conditions). Fig. 2a and b shows the surface mesh employed (the volume mesh had `nelem=360,247, npoin=67,785`), as well as the wavepattern obtained. Fig. 2c and d shows the convergence history for the wavedrag and residuals, respectively. The wavedrag coefficient obtained for the present mesh was $c_w = 0.532 \times 10^{-4}$, and compares favourably with the experimental data of $c_w = 0.567 \times 10^{-4}$ [32] and the moving mesh result of $c_w = 0.607 \times 10^{-4}$ [43]. One can see that the implicit-advection schemes converge significantly faster. The timings, summarized in Table 2, confirm a considerable speedup. As before, note the surprising speedup obtained by the multistep explicit-advection schemes.

8.3. von Karman vortex street

The third example considered is also a well known benchmark case [55]. A circular cylinder is suspended in a uniform stream of incompressible fluid. The separation at the back of the cylinder generates the so-called von Karman vortex street, whose characteristics depend on the Reynolds number

$$Re = \frac{\rho V_\infty D}{\mu},$$

where D denotes the diameter of the cylinder. This is essentially a 2-D example, but was run with the 3-D solver. A mesh of `nelem=113,056, npoin=23,228` was used for the simulation, with special placement of points in the vicinity of the cylinder. The parameters were chosen such that the resulting Reynolds number was $Re = 190$.

Fig. 3a and b shows the surface grid and the absolute value of the velocity in a cut plane. In order to compare the different schemes, the run was started impulsively using the explicit-advection scheme. The run was continued until the vortex street was fully developed. Starting from this (restart) state, the different schemes were exercised and compared to one another. The lift of the cylinder as a function of time for the original explicit-advection projection scheme, the 2/3/5-step explicit-advection projection scheme, as well as the implicit-advection Crank–Nicholson ($\theta = 0.50$) scheme with different timesteps is displayed in Fig. 3c. For the implicit timestepping scheme, the pseudo steady state system given by Eqs. (43) and (44) was solved using a local Courant-nr. of $C = 4.0$, implicit advection and SGS relaxation. Typically, 10–20 SGS iterations were required for convergence. Observe that for the schemes that employ a reasonable timestep,

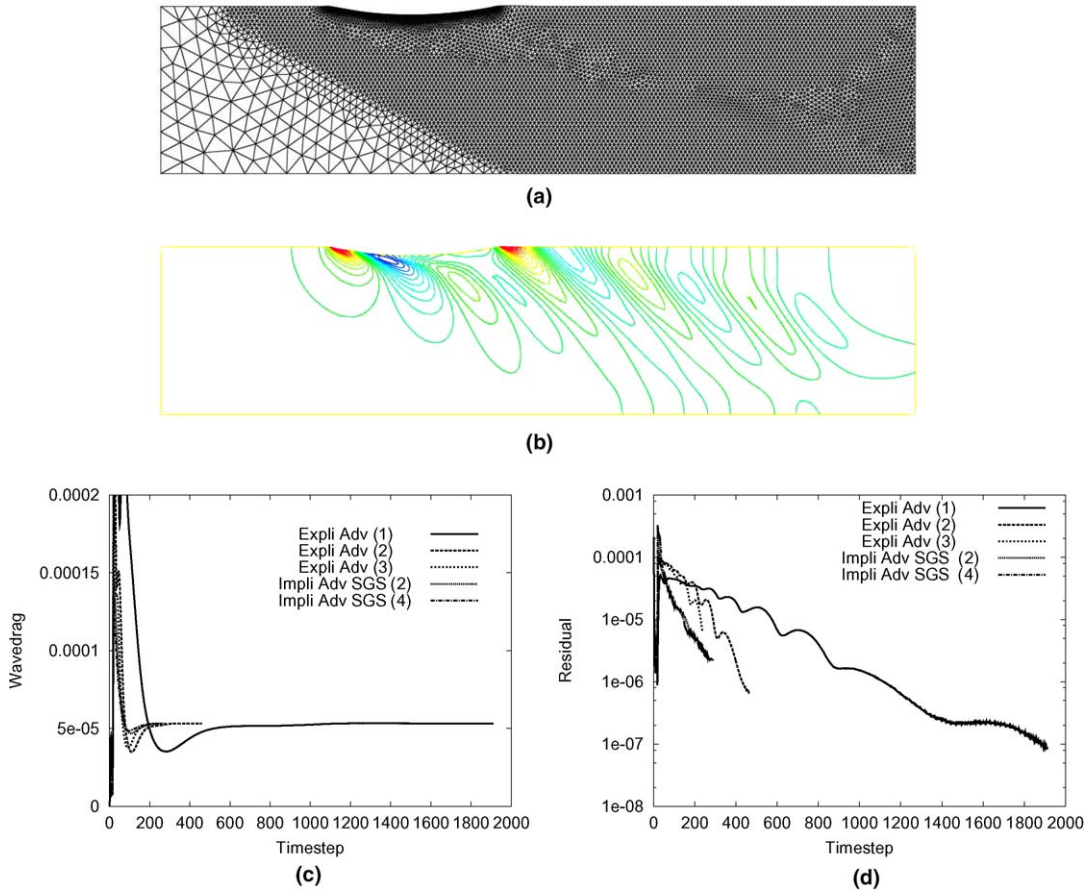
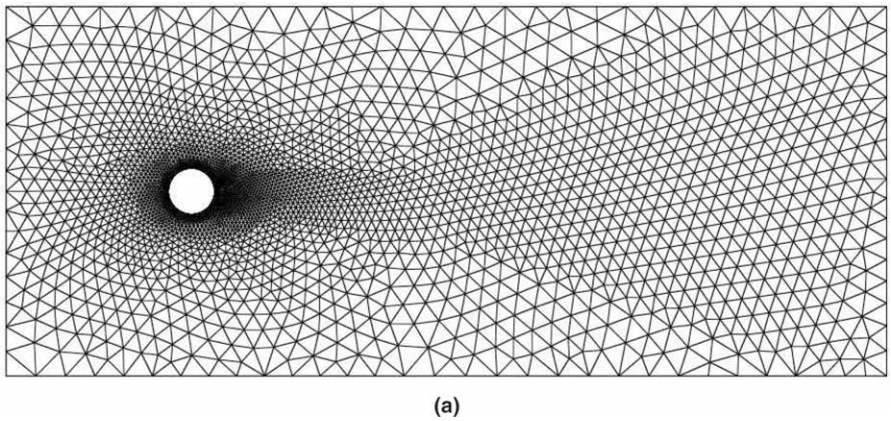


Fig. 2. (a) Wigley Hull: surface mesh, (b) Wigley Hull: free surface elevation, (c,d) Wigley Hull: convergence history for wavedrag and residuals.

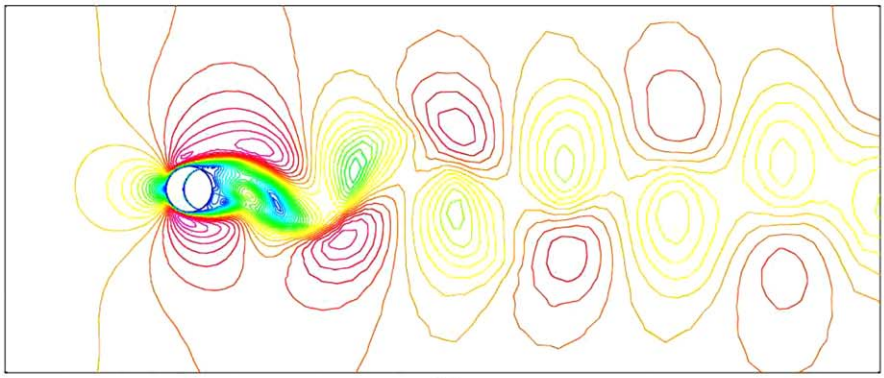
Table 2
Wigley Hull

Scheme	ntime	CPU [s]	Speedup
Ex 1	1910	3232	1.00
Ex 2	350	890	3.63
Ex 3	235	778	4.15
ImSGS (2)	285	682	4.74
ImSGS (4)	260	713	4.53

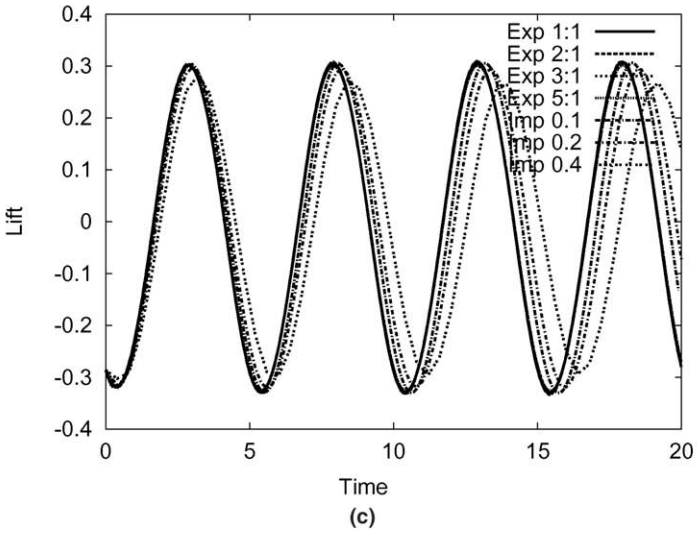
and in particular all the explicit-advection multistep schemes, the results are almost identical. The Strouhal number obtained is approximately $S = 0.2$, in good agreement with experiments [55]. Not surprisingly, as the timestep is increased, the implicit schemes remain stable, but the solution deteriorates. Recall that for $\Delta t = 0.4$, a particle in the free stream crosses the cylinder in 2.5 timesteps (!). The timings recorded are summarized in Table 3. We remark that the timestep for the explicit-advection schemes is approximate.



(a)



(b)



(c)

Fig. 3. (a) von Karman vortex street: surface mesh, (b) von Karman vortex street: Abs(Veloc), (c) von Karman vortex street: lift history.

Table 3
von Karman vortex street

Scheme	Δt	ntime	CPU [s]	Speed
Ex 1 0.1	0.002	9961	12,929	1.00
Ex 2 0.4	0.008	2490	4194	3.08
Ex 3 0.6	0.012	1660	3296	3.92
Ex 5 0.8	0.016	1245	3201	4.03
Ex 5 1.2	0.025	830	1546	8.36
Ex 5 1.6	0.033	623	1114	11.60
Ex 5 1.8	0.037	554	995	12.99
ImSGS (5)	0.1	200	3189	4.05
ImSGS (5)	0.2	100	1612	8.02
ImSGS (5)	0.4	50	962	13.43

8.4. High-Re bump

The fourth example considered is high Reynolds-number flow past a parabolic bump. As before, this is essentially a 2-D example, but was run with the 3-D solver. The parameters were set to: $\rho = 1.0$, $\mathbf{v} = (1, 0, 0)$, $\mu = 10^{-5}$, $L = 1$, which implies a Reynolds-number of $Re = 10^5$. The Baldwin–Lomax turbulence model [3] was used. The resulting flow is steady and shows the development of a boundary layer. Fig. 4a and b shows a zoom of the surface mesh employed, as well as the velocity field obtained. As in the previous cases, although the mesh is coarse (`nelem=23,785, npoin=5,674`), it still allows for a meaningful comparison of the different solvers.

As before, this problem was solved using:

- The standard explicit-advection projection scheme ($C = 0.1$) with full pressure solution every 20 time-steps, and partial pressure solution in between (max 20 PCG iterations);
- The implicit-advection projection scheme ($C = 4.0$) with SGS relaxation;
- The implicit-advection projection scheme ($C = 4.0$) with LU–SGS–GMRES solver for the advection.

For the pressure-Poisson equation (as well as the velocity PCG solver of the explicit-advection projection scheme), linelet preconditioning was used. Fig. 4c and d shows the convergence history for the lift and residuals, respectively. As before, one can see that the implicit-advection schemes converge significantly faster. The timings recorded have been summarized in Table 4.

8.5. Ahmed car body

The fifth example considered is high Reynolds-number flow past the so-called Ahmed body. This is a standard test case for external car aerodynamics. The parameters were set to: $\rho = 1.0$, $\mathbf{v} = (1, 0, 0)$, $\mu = 2.33 \times 10^{-7}$, $L = 1$, which implies a Reynolds-number of $Re = 4.29 \times 10^6$. The $k-\epsilon$ turbulence model was used. The resulting flow is quasi-steady and shows the development of a vortex train behind the body. Fig. 5a–c shows the surface mesh employed, as well as the pressure and velocity field obtained. Note the boundary layer mesh (the complete mesh consisted of (`nelem=420,245, npoin=77,279`).

This problem was solved using the fully implicit scheme with a timestep of $\Delta t = 0.1$, a local Courant-nr. of $C = 4.1$ and LU-SGS-GMRES relaxation. Fig. 5d,e show the forces computed. Note that after $t = 30$, a quasi-steady state is reached. The speedup compared to the explicit-advection scheme is approximately 1:10.

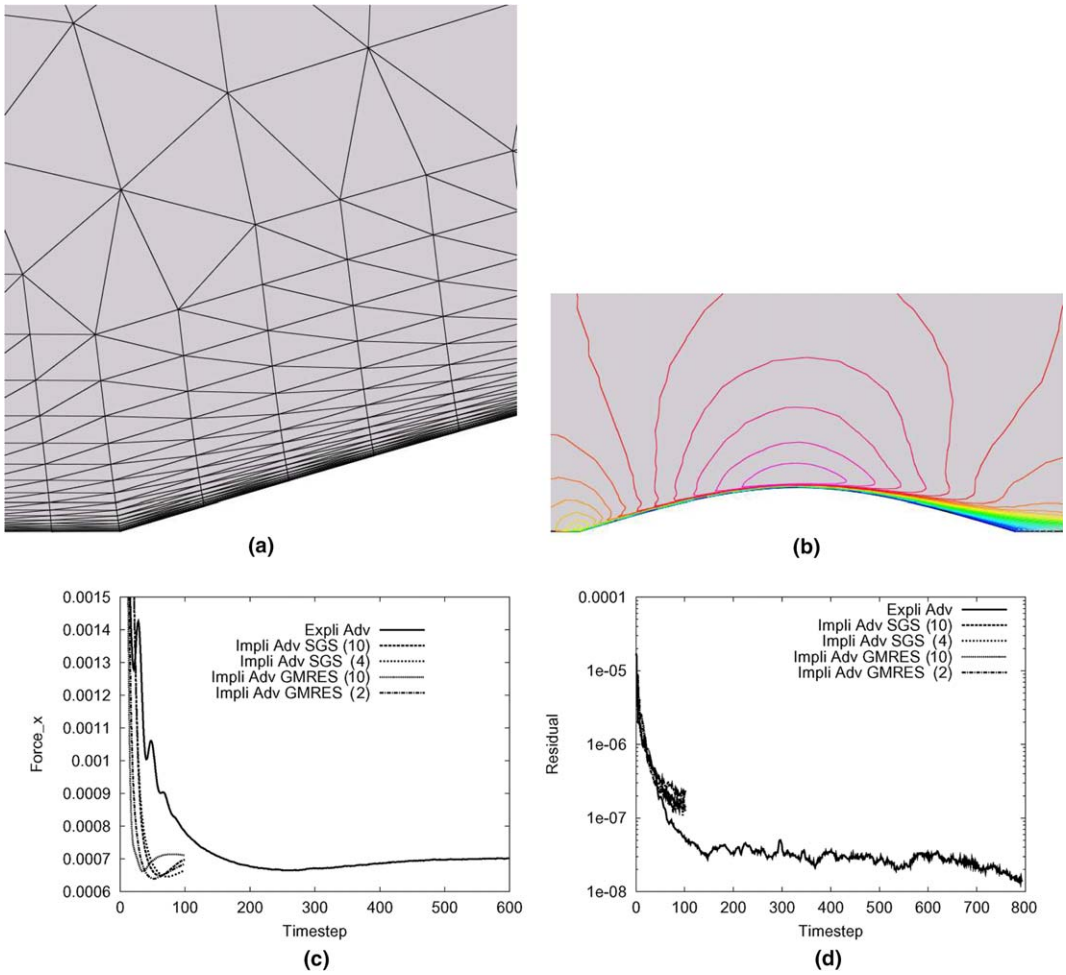


Fig. 4. (a) Bump: detail of surface mesh, (b) bump: velocities, (c,d) bump: convergence history for lift and residuals.

Table 4
High-Re bump

Scheme	ntime	CPU [s]	Speedup
Exp	790	144	1.00
ImSGS (10)	100	31	4.65
ImSGS (4)	100	25	5.76
ImGMRES (10)	100	38	3.79
ImGMRES (2)	100	25	5.76

8.6. Generic car model

The sixth example considered is a realistic car configuration, such as those commonly encountered for external car aerodynamics. The parameters were set to: $\rho = 1.2$, $\mathbf{v} = (12, 0, 0)$, $\mu = 1.80 \times 10^{-5}$, which implies a Reynolds-number of $Re = 3.2 \times 10^6$. The Smagorinsky turbulence model was used. As with the

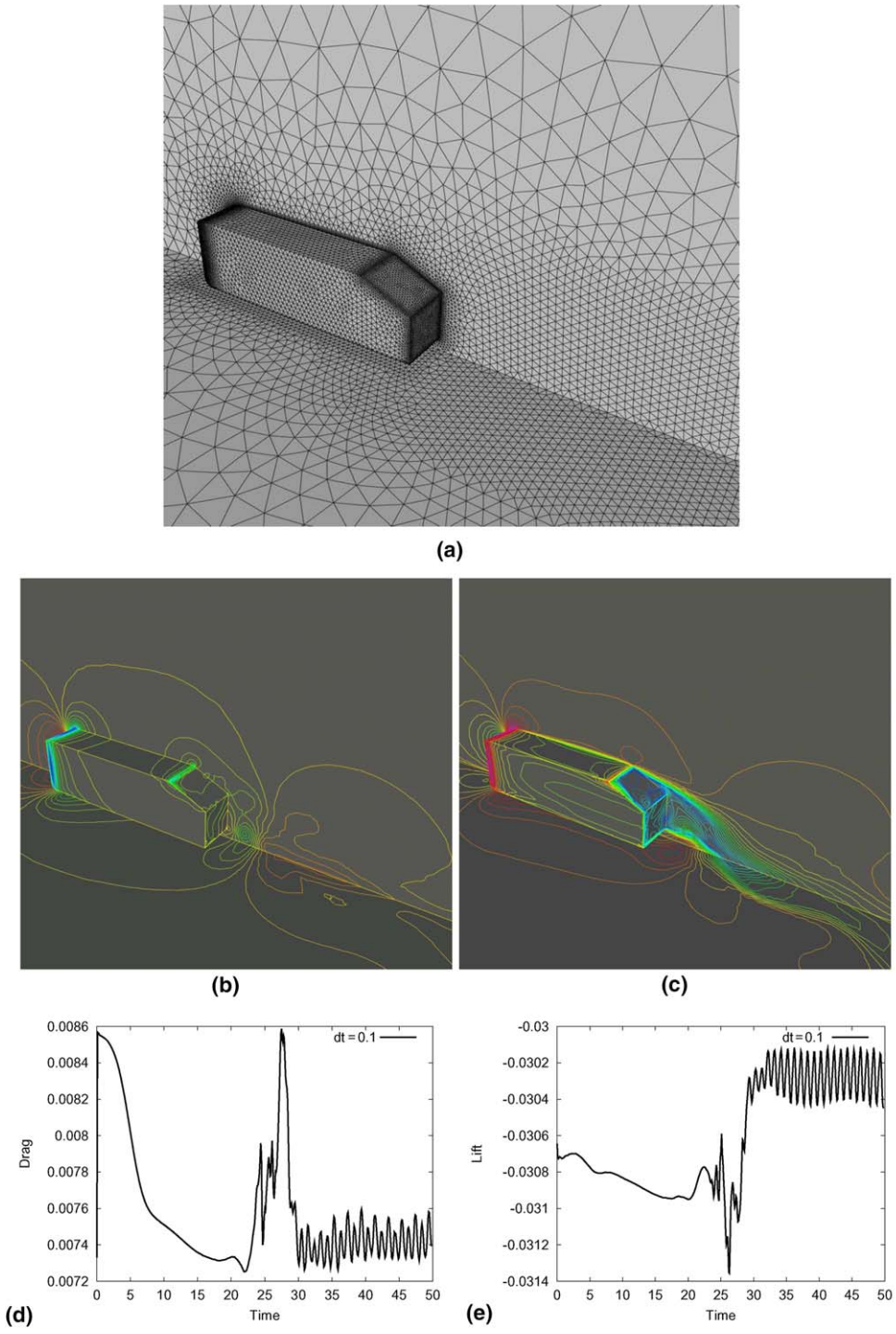


Fig. 5. (a) Ahmed body: surface mesh, (b,c) Ahmed body: surface pressure and speed, (d,e) Ahmed body: drag and lift history.

Ahmed body, the resulting flow is quasi-steady and shows the development of a vortex train behind the body. Fig. 6a–c shows the surface mesh employed, as well as the pressure and velocity field obtained. Note, as before, the boundary layer mesh (the complete mesh consisted of (nelem=3,690,728, npoin=657,696).

This problem was run in two stages. First, local timestepping with a local Courant-nr. of $C = 4.1$ and LU-SGS-GMRES relaxation was used to achieve quickly the overall flowfield. Then, the fully implicit scheme with a global timestep of $\Delta t = 0.01$, a local Courant-nr. of $C = 4.1$ and LU-SGS-GMRES relaxation was employed to run the time-accurate simulation. Fig. 6d shows the drag force computed. For this case, the speedup compared to the explicit-advection scheme is in excess of 1:10.

8.7. Airwake of T-AKE 1 ship

This seventh example demonstrates the use of the multistage explicit-advection projection scheme for a typical production run. Specifically, the topside air-wake flow and stack gas simulation for the T-AKE 1 ship is considered. This LES study was performed for a 0 degree angle of attack of inflow. The parameters for this study were set to: angle of attack of inflow $\alpha = 0.0$, density $\rho = 1.225\ \mathrm{kg/m^3}$, velocity $v = (28.8, 0, 0)\ \mathrm{m/s}$ and viscosity $\mu = 1.789 \times 10^{-}5\ \mathrm{kg/m/s}$, yielding a Reynolds number of $Re = 8 \times 10^7$. Turbulence was modelled with the Smagorinsky model [57]. The numerical study was performed for the ship at full-scale: the length from stern to bow is 200 m, the height from the water line to the tip of the stacks is 40 m. Fig. 7a shows the geometry definition of the T-AKE 1. Fig. 7b shows a close up to the surface mesh used. The volume mesh contains 8,863,995 elements and 1,574,805 points. The time integration was performed using the explicit-advection projection scheme with five stages, allowing a Courant number of $C = 1.0$. This gives a speedup of approximately 1:10 over the second generation explicit-advection projection scheme. The solution was integrated for 2 min of real time with a previous initialization to bring the flow to a pseudo-steady state. The timestep was of approximately $\Delta t = 0.36 \times 10^{-2}$ s, which permits the accurate capture of the unsteady features for this turbulent flow. Using the implicit-advection option does not yield any significant gain in speed, and may compromise the temporal accuracy of the solution. The unsteady flow can be seen in the instantaneous streamlines shown in Fig. 7c. The (real) time needed to complete a run like this is of approximately 1 month using 32 processors on an SGI 3800. Comparisons with wind tunnel measurements [14] support the accuracy of the multistage explicit-advection projection scheme. A remarkable agreement between the experiment and the computational results is presented in Table 5 for the temperature measured at different heights above the helipad of the ship. Levels of concentration were calculated for two chemical species. Fig. 7d shows an iso-surface of concentration. For a detailed report of this run, see [52,9].

8.8. Paper mill optimization

The combined enhancements in solver speed and computer power have made it possible to use less sophisticated optimization techniques to improve products or processes. A typical case is the paper machine headbox shown in Fig. 8. This problem was taken from [26]. The objective is to change the shape of the upper surface so that the outflow at all three lower exits is equal.

Four Bezier-points that describe the upper surface were taken as design parameters. The optimization was carried out using a steepest descent algorithm, where the gradient of the objective function was computed via second order finite differences. Although the geometry is 2-D, the case was run with the 3-D code on a PC (2.2 GHz Intel P4, Linux OS). A typical mesh had O(25 Kels), and the CFD runs required less than 1 min to converge. Fig. 8 shows the geometry/outline of the problem (initial:straight, final:curved), the final velocity field, as well as the convergence history of the objective function.

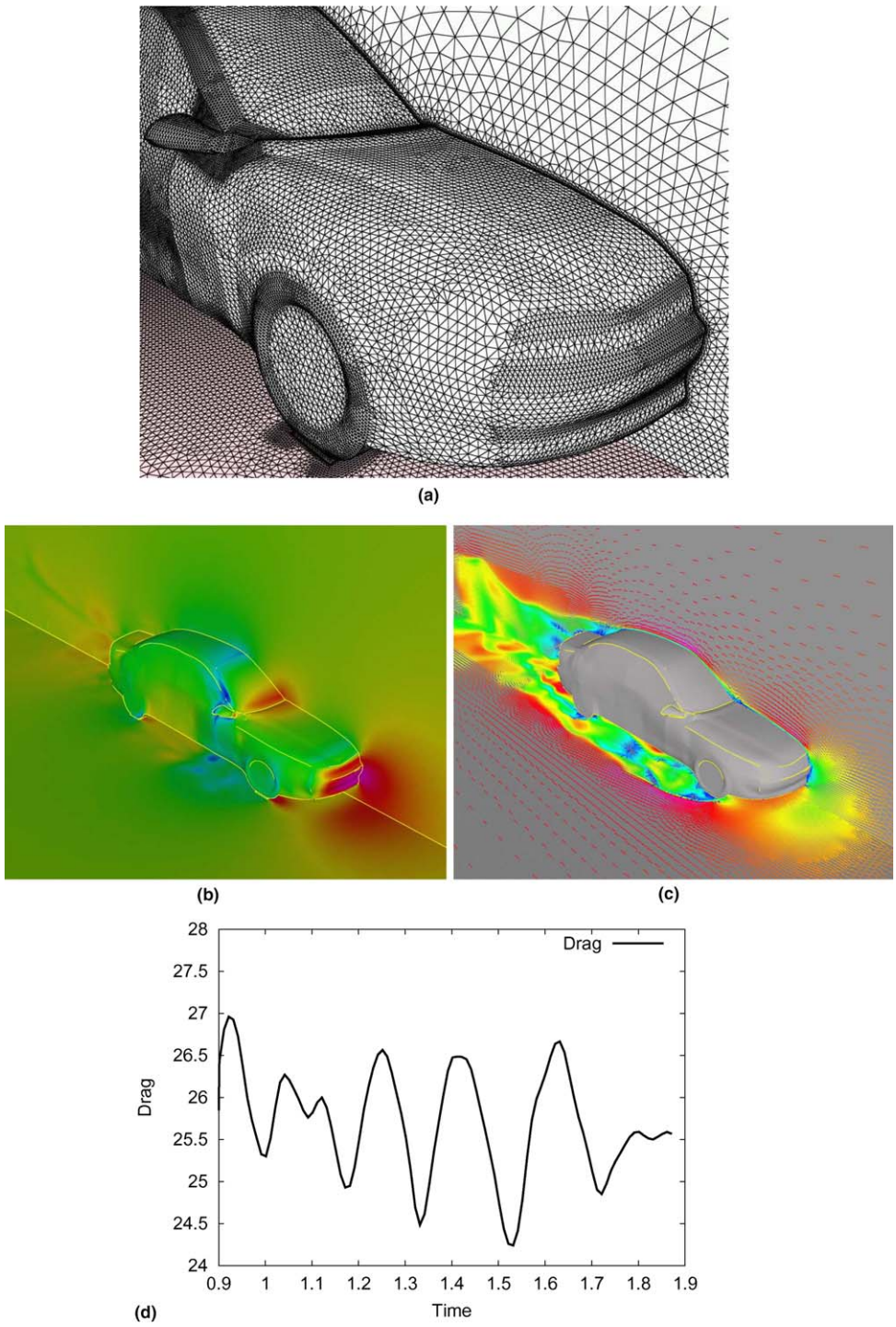


Fig. 6. (a) Generic car model: surface mesh, (b, c) Generic car model: surface pressure and velocities, (d) Generic car model: Body: drag history.

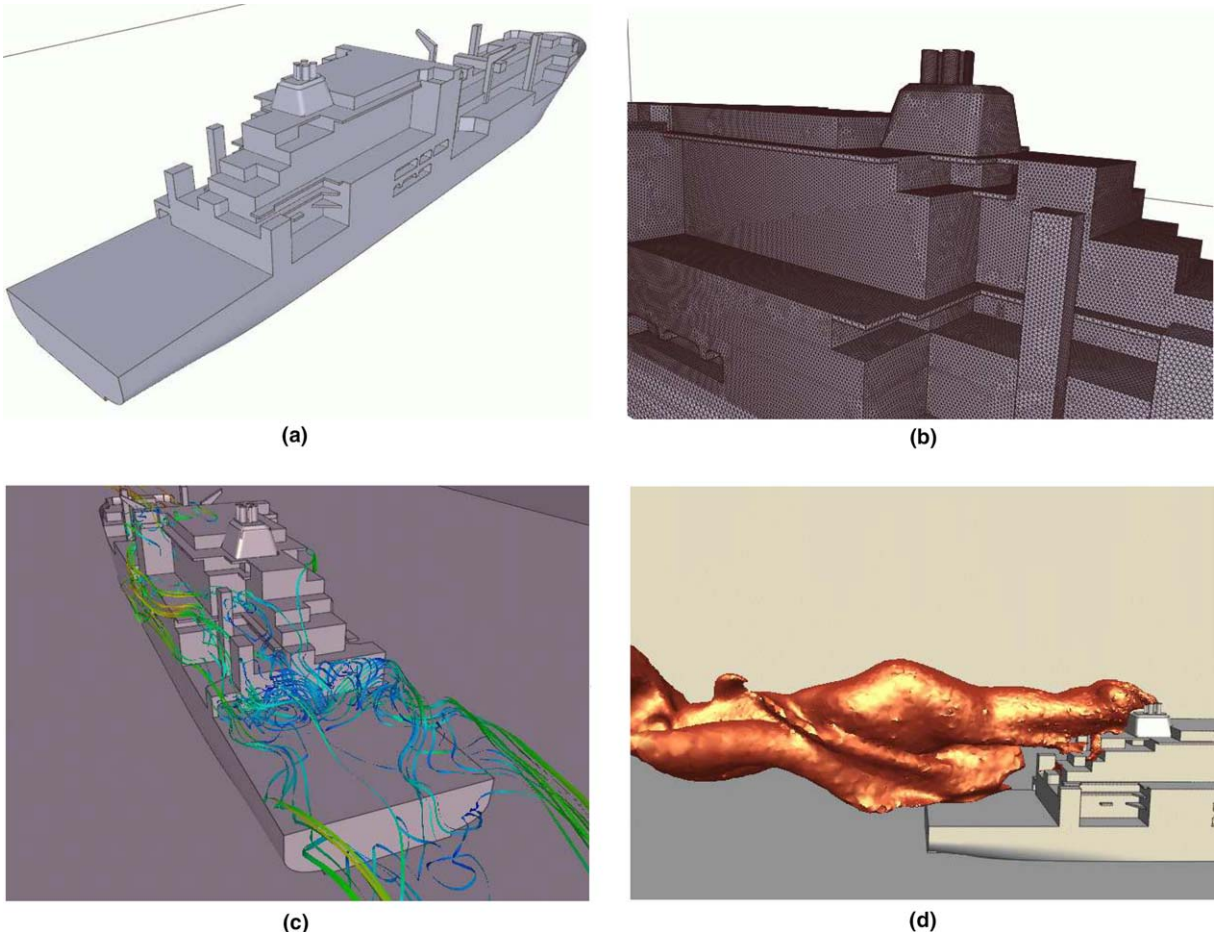


Fig. 7. (a) T-AKE 1 Ship: geometry definition, (b) T-AKE 1 Ship: detail of surface mesh, (c) T-AKE 1 Ship: instantaneous streamlines, (d) T-AKE 1 Ship: iso-surface of concentration.

Table 5
Comparison to experiments

Height (m)	T_{exp} (°C)	T_{cfd} (°C)	Diff. (%)
5	20.10	20.14	0.2
10	20.10	20.23	0.6
15	20.30	20.50	1.0
20	21.10	21.38	1.3
25	22.50	23.86	6.0

9. Conclusions and outlook

A second-generation incompressible flow solver of the projection type has been improved by considering:

- Runge–Kutta substepping for the advective terms;
- Implicit advection treatment via SGS and LU-SGS-GMRES;

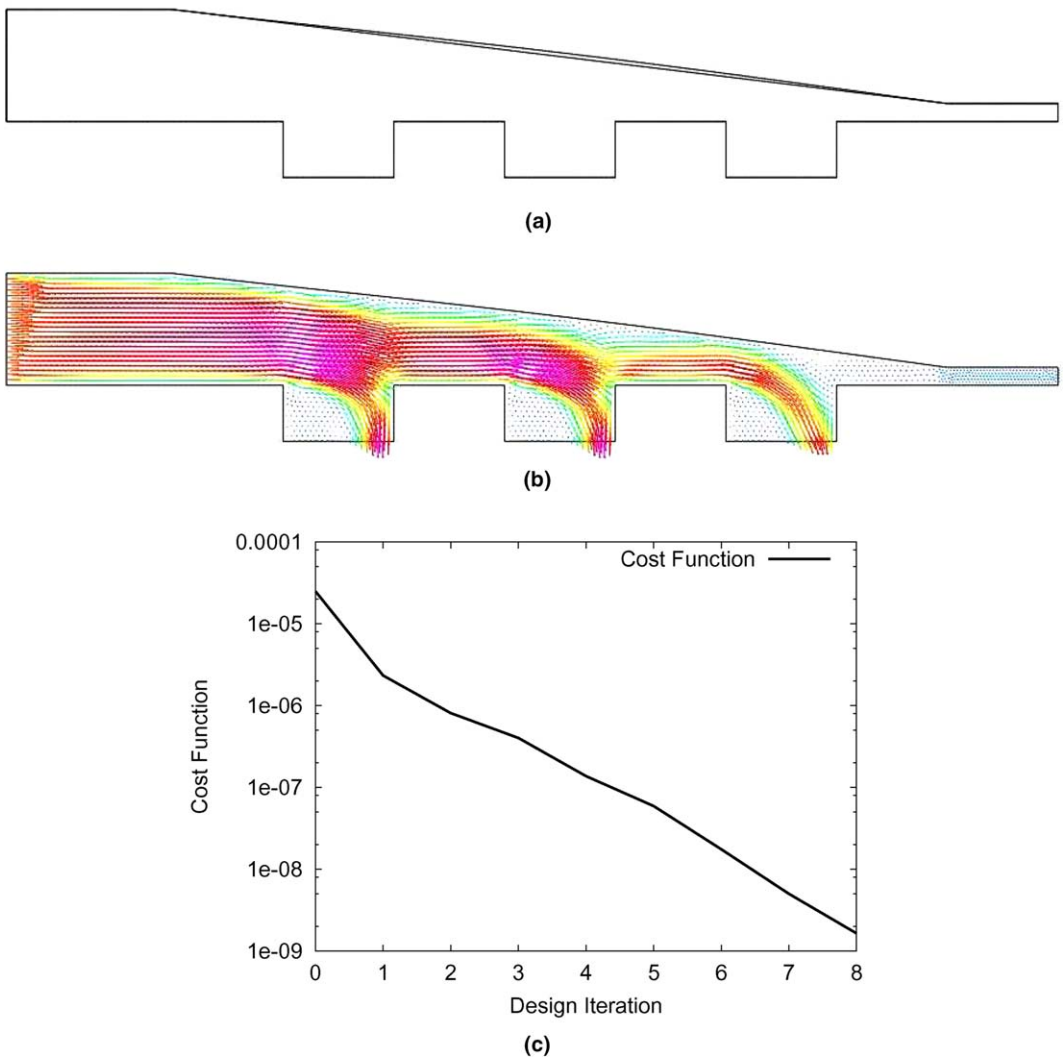


Fig. 8. (a) Paper machine headbox: initial and final geometry, (b) paper machine headbox: velocities for final geometry, (c) paper machine headbox: convergence of cost function.

– Implicit timestepping via solution of steady-state problems in pseudo-time; and
– Linelet preconditioning for highly anisotropic grids commonly encountered for RANS applications.

The combined effect of these enhancements has led to speedups of the order of O(1:5–1:10) without deterioration of results, as demonstrated on a variety of academic and industrial problems. Future work will consider:

– Implicit coupling to the energy equation via Boussinesq approximation;
– Further exploration of (parallel) acceleration techniques;
– Inclusion of vorticity confinement for implicit solvers;
– Adaptive, physics-dependent timestepping; and
– Adaptive mesh refinement for wakes and vortical structures.

Acknowledgements

This work was partially supported by DTRA and AFOSR. The technical monitors were Drs. Michael Giltrud, Young Sohn and Leonidas Sakell.

It is a pleasure to acknowledge the help, guidance and permission of General Motors Corp. to use the generic car model for benchmarking purposes.

References

[1] B. Alessandrini, G. Delhommeau, A multigrid velocity-pressure-free surface elevation fully coupled solver for calculation of turbulent incompressible flow around a hull, in: Proc. 21st Symp. on Naval Hydrodynamics, Trondheim, Norway, June 1996.
[2] J. Alonso, L. Martinelli, A. Jameson, Multigrid unsteady Navier–Stokes calculations with aeroelastic applications, AIAA-95-0048, 1995.
[3] B.S. Baldwin, H. Lomax, Thin-layer approximation and algebraic model for separated turbulent flows, AIAA-78-257, 1978.
[4] J.B. Bell, P. Colella, H. Glaz, A second-order projection method for the Navier–Stokes equations, J. Comp. Phys. 85 (1989) 257–283.
[5] J.B. Bell, D.L. Marcus, A second-order projection method for variable-density flows, J. Comp. Phys. 101 (1992) 2.
[6] A.N. Brooks, T.J.R. Hughes, Streamline upwind/Petrov Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1982) 199–259.
[7] F. Camelli, R. Löhner, Combining the Baldwin–Lomax and Smagorinsky turbulence models to calculate flows with separation regions, AIAA-02-0426, 2002.
[8] F. Camelli, O. Soto, R. Löhner, W. Sandberg, R. Ramamurti, Topside LPD17 flow and temperature study with an implicit monolithic scheme; AIAA-03-0969, 2003.
[9] F. Camelli, R. Löhner, W.C. Sandberg, R. Ramamurti, VLES study of ship stack gas dynamics, AIAA-04-0072, 2004.
[10] A.J. Chorin, A numerical solution for solving incompressible viscous flow problems, J. Comp. Phys. 2 (1967) 12–26.
[11] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comp. 22 (1968) 745–762.
[12] R. Codina, Pressure stability in fractional step finite element methods for incompressible flows, J. Comp. Phys. 170 (2001) 112–140.
[13] Conferences—See the following Conference Series: Finite Elements in Fluis I–IX, John Wiley & Sons, Int. Conf. Numer. Methods Fluid Dyn. I-XII, Spinger Lecture Notes in Physics, AIAA CFD Conf. I–XV, AIAA CP, Numer. Methods Laminar and Turbulent Flow, Pinerige Press, and others.
[14] Wind-Tunnel Tests with T-AKE 1; Danish Maritime Institute Rep. DMI-2002031, June 2002.
[15] J. Donea, S. Giuliani, H. Laval, L. Quartapelle, Solution of the unsteady Navier–Stokes equations by a fractional step method, Comput. Methods Appl. Mech. Engrg. 30 (1982) 53–73.
[16] E. Eaton, Aero-acoustics in an automotive HVAC Module, in: American PAM User Conf., Birmingham, Michigan, 24–25 October 2001.
[17] J.R. Farmer, L. Martinelli, A. Jameson, A fast multigrid method for solving incompressible hydrodynamic problems with free surfaces, AIAA J. 32 (6) (1993) 1175–1182.
[18] M. Fortin, F. Thomasset, Mixed finite element methods for incompressible flow problems, J. Comp. Phys. 31 (1979) 113–145.
[19] L.P. Franca, T.J.R. Hughes, A.F.D. Loula, I. Miranda, A new family of stable elements for the stokes problem based on a mixed Galerkin/Least-squares finite element formulation, in: T.J. Chung, G. Karr (Eds.), Proc. 7th Int. Conf. Finite Elements in Flow Problems, Huntsville, AL, 1989, pp. 1067–1074.
[20] L.P. Franca, S.L. Frey, Stabilized finite element methods: II. The incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 99 (1992) 209–233.
[21] P.M. Gresho, C.D. Upson, S.T. Chan, R.L. Lee, Recent progress in the solution of the time-dependent, three-dimensional, incompressible Navier–Stokes equations, in: T. Kawai (Ed.), Proc. 4th Int. Symp. Finite Element Methods in Flow Problems, University of Tokio Press, 1982, pp. 153–162.
[22] J.P. Gregoire, J.P. Benque, P. Lasbleiz, J. Goussebaile, 3-D industrial flow calculations by finite element method, Springer Lecture Notes in Physics 218 (1985) 245–249.
[23] P.M. Gresho, S.T. Chan, On the theory of semi-implicit projection methods for viscous incompressible flows and its implementation via a finite element method that introduces a nearly-consistent mass matrix, Int. J. Numer. Methods Fluids 11 (1990) 621–659.
[24] M.D. Gunzburger, Mathematical Aspects of Finite Element Methods for Incompressible Viscous Flows, in: Dwoyer, Hussaini, Voigt (Eds.), Finite Elements: Theory and Application, Springer-Verlag, 1987, pp. 124–150.

[25] M.D. Gunzburger, R. Nicolaides (Eds.), Incompressible Computational Fluid Dynamics: Trends and Advances, Cambridge University Press, 1993.
[26] J. Hämäläinen, T. Malkamäki, J. Toivanen, Genetic algorithms in shape optimization of paper machine headbox, in: K. Miettinen, M. Mäkelä, P. Neittaanmäki, J. Periaux (Eds.), Evolutionary Algorithms in Engineering and Computer Science, John Wiley & Sons, 1999, pp. 435–443.
[27] M.M. Hafez (Ed.), Numerical Simulations of Incompressible Flows, World Scientific, 2003.
[28] S.R. Hanna, S. Tehranian, B. Carissimo, R.W. Macdonald, R. Löhner, Comparisons of model simulations with observations of mean flow and turbulence within simple obstacle arrays, Atmospheric Environment 36 (2002) 5067–5079.
[29] T. Hino, Computation of free surface flow around an advancing ship by the Navier–Stokes equations, in: Proc. 5th Int. Conf. on Numerical Ship Hydrodynamics, Hiroshima, Japan 1989.
[30] T. Hino, L. Martinelli, A. Jameson, A finite-volume method with unstructured grid for free surface flow, in: Proc. 6th Int. Conf. on Numerical Ship Hydrodynamics, Iowa City, August 1993.
[31] T. Hino, An unstructured grid method for incompressible viscous flows with a free surface; AIAA-97-0862 1997.
[32] Cooperative Experiments on Wigley Parabolic Models in Japan, 17th ITTC Resistance Committee Report, 2nd ed., 1983.
[33] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume methods using Runge–Kutta time-stepping schemes, AIAA-81-1259, 1981.
[34] T.C. Jue, B. Ramaswamy, J.E. Akin, Finite element simulation of 2-D Benard convection with gravity modulation, FED-Vol. 123, ASME 1991, pp. 87–101.
[35] Y. Kallinderis, A. Chen, An incompressible 3-D Navier–Stokes method with adaptive hybrid grids; AIAA-96-0293, 1996.
[36] Y. Li, T. Kamioka, T. Nouzawa, T. Nakamura, Y. Okada, N. Ichikawa, Verification of aerodynamic noise simulation by modifying automobile front-pillar shape; JSAE 20025351, JSAE Annual Conf., Tokyo, July 2002.
[37] K.J. Karbon, S. Kumarasamy, Computational aeroacoustics in automotive design, computational fluid and solid mechanics, in: Proc. First MIT Conference on Computational Fluid and Solid Mechanics, Boston, June 2001, pp. 871–875.
[38] K.J. Karbon, R. Singh, Simulation and design of automobile Sunroof buffeting noise control, in: 8th AIAA-CEAS Aero-Acoustics Conf., Brenckridge, June 2002.
[39] D.W. Kelly, S. Nakazawa, O.C. Zienkiewicz, J.C. Heinrich, A note on anisotropic balancing dissipation in finite element approximation to convection diffusion problems, Int. J. Numer. Methods Engrg. 15 (1980) 1705–1711.
[40] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comp. Phys. 59 (1985) 308–323.
[41] Y. Li, T. Kamioka, T. Nouzawa, T. Nakamura, Y. Okada, N. Ichikawa, Verification of aerodynamic noise simulation by modifying automobile front-pillar shape, JSAE 20025351, JSAE Annual Conf., Tokyo, July 2002.
[42] R. Löhner, A fast finite element solver for incompressible flows, AIAA-90-0398, 1990.
[43] R. Löhner, C. Yang, E. Oñate, S. Idelssohn, An unstructured grid-based, parallel free surface solver, Appl. Numer. Math. 31 (1999) 271–293.
[44] R. Löhner, Applied CFD Techniques, John Wiley & Sons, 2001.
[45] H. Luo, J.D. Baum, R. Löhner, A fast, matrix-free implicit method for compressible flows on unstructured grids, J. Comp. Phys. 146 (1998) 664–690.
[46] H. Luo, J.D. Baum, R. Löhner, An accurate, fast, matrix-free implicit method for computing unsteady flows on unstructured grids, AIAA-99-0937, 1999.
[47] H. Luo, D. Sharov, J.D. Baum, R. Löhner, A class of matrix-free implicit methods for compressible flows on unstructured grids, in: First International Conference on Computational Fluid Dynamics, Kyoto, Japan, 10–14 July 2000.
[48] D. Martin, R. Löhner, An implicit linelet-based solver for incompressible flows, AIAA-92-0668, 1992.
[49] A.T. Patera, A spectral element method for fluid dynamics: laminar flow in a channel expansion, J. Comp. Phys. 54 (1984) 468–488.
[50] J. Peraire, K. Morgan, J. Peiro, The simulation of 3D incompressible flows using unstructured grids, in: D.A. Caughey, M.M. Hafez (Eds.), Frontiers of Computational Fluid Dynamics, John Wiley, 1994 (Chapter 16).
[51] R. Ramamurti, R. Löhner, A parallel implicit incompressible flow solver using unstructured meshes, Comput. Fluids 5 (1996) 119–132.
[52] R. Ramamurti, W.C. Sandberg, LPD-17 Topside Aerodynamic Study: FEFLO; NRL Memorandum Rep. NRL/MR/6410-00-8498, LCP&FD, NRL, 2000.
[53] A. Rizzi, L. Eriksson, Computation of inviscid incompressible flow with rotation, J. Fluid Mech. 153 (1985) 275–312.
[54] Y. Saad, Iterative Methods for Sparse Linear Systems, PWS Publishing, Boston, 1996.
[55] H. Schlichting, Boundary Layer Theory, McGraw-Hill, 1979.
[56] D. Sharov, H. Luo, J.D. Baum, R. Löhner, Implementation of unstructured grid GMRES + LU-SGS method on shared-memory, cache-based parallel computers, AIAA-00-0927, 2000.
[57] J. Smagorinsky, General circulation experiments with the primitive equations. I. The basic experiment, Mon. Weather Rev. 91 (1963) 99–164.

[58] O. Soto, R. Löhner, J.R. Cebral, An implicit monolithic time accurate finite element scheme for incompressible flow problems, AIAA-01-2616-CP, 2001.
[59] O. Soto, R. Löhner, F. Camelli, A linelet preconditioner for incompressible flow solvers, Int. J. Heat Fluid Flow 13 (1) (2003) 133–147.
[60] A. Soulaimani, M. Fortin, Y. Ouellet, G. Dhatt, F. Bertrand, Simple continuous pressure elements for two- and three-dimensional incompressible flows, Comput. Methods Appl. Mech. Engrg. 62 (1987) 47–69.
[61] A. Takamura, M. Zhu, D. Vinteler, Numerical simulation of pass-by maneuver using ale technique, in: JSAE Annual Conf. (Spring), Tokyo, May 2001.
[62] C. Taylor, P. Hood, A numerical solution of the Navier–Stokes equations using the finite element method, Comp. Fluids 1 (1973) 73–100.
[63] T.E. Tezduyar, R. Shih, S. Mittal, S.E. Ray, Incompressible flow computations with stabilized bilinear and linear equal-order interpolation velocity-pressure elements, UMSI Rep. 90 (1990).
[64] F. Thomasset, Implementation of Finite Element Methods for Navier–Stokes Equations, Springer-Verlag, 1981.
[65] J. Waltz, R. Löhner, A grid coarsening algorithm for unstructured multigrid applications, AIAA-00-0925, 2000.
[66] P. Wesseling, Principles of Computational Fluid Dynamics, Springer, 2001.
[67] C. Yang, R. Löhner, Fully nonlinear ship wave calculation using unstructured grids and parallel computing, in: Proc. 3rd Osaka Colloquium on Advanced CFD Applications to Ship Flow and Hull Form Design, Osaka, Japan, May 1998, pp. 125–150.

Further reading

[i] M.J. Guillot, M.A. Walker, Unsteady analysis of the air wake over the LPD-17, AIAA-00-4125, 2000.
[ii] J.D. Huffenus, D. Khaletzky, A finite element method to solve the Navier–Stokes equations using the method of characteristics, Int. J. Numer. Methods Fluids 4 (1984) 247–269.