Computability and adaptivity in CFD

J. Hoffman[1] and C. Johnson[2]

[1] *Courant Institute, 251 Mercer Street, New York, NY-10012, USA*
[2] *Department of Computational Mathematics, Chalmers University, SE-41296 Göteborg, Sweden*

Abstract

KEY WORDS: adaptivity, computability, predictability, adaptive finite element method, a posteriori error estimate, incompressible flow, turbulence

1. Introduction

The *Navier-Stokes equations* form the basic mathematical model in *fluid mechanics* and describe a large variety of phenomena of *fluid flow* occurring in hydro- and aero-dynamics, processing industry, biology, oceanography, meteorology, geophysics and astrophysics. Fluid flow may contain features of *incompressible* and *compressible* flow, *Newtonian* and *non-Newtonian* flow, and *turbulent* and *laminar* flow, with turbulent flow being irregular with rapid fluctuations in space and time and laminar flow being more organized. *Computational Fluid Dynamics* CFD concerns the digital/computational simulation of fluid flow by solving the Navier-Stokes equations numerically.

The basic issues of CFD is *computability* relating to errors from numerical computation, and *predictability* relating to errors from imprecision in given data. The basic question of computability/predictability for a given flow situation may be formulated as follows: what quantity can be computed/predicted to what tolerance/norm to what cost? We emphasize the *quantitative* aspects concerning both the choice of *quantity of interest*, or *output*, the error tolerance/norm and the cost. For computability the cost reflects the precision of the computation with direct connection to the computational work (number of arithmetical operations and memory requirements), and for predictability the cost reflects the required precision of data. We may expect a turbulent flow to be more costly than a laminar flow. Further, we expect a pointwise quantity (e.g the viscous stresses a specific points) to be more costly than an average quantity (e.g. the drag or lift), or more generally the cost to increase with the strength of the norm, and of course also with decreasing tolerance. The purpose of an *adaptive* computational method is to approach a *computational goal* of computing a quantity of interest at a minimal computational cost.

In these notes we give a survey of our work on computability of fluid flow using *adaptive finite element methods*, with focus on non-stationary incompressible Newtonian laminar and beginning turbulent flow. Our presentation follows the general approch to adaptive finite element methods developed together with the group of Prof. Rannacher, which is based on *a posteriori error estimates*

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

derived by *duality* techniques. For an overview of adaptive finite element methods including references, we refer to the survey articles Eriksson *et al*., 1995, Becker and Rannacher, 2001, and the books Eriksson *et al*., 2001, and **?**, containing many details on various aspects of adaptive finite element methods omitted in these notes. For an overview of finite element methods for the incompressible Navier-Stokes equations including references, we refer to Rannacher, 1999, and for more details on the class of methods considered in these notes, we refer to Hoffman and Johnson, 2002a. For a survey of turbulence modeling we refer to Gatski *et al*., 1996, and Wagner and Liu, 1999, and references therein.

Computational simulation of turbulent flow presents special challenges. To computationally resolve all scales of the flow in a *Direct Numerical Simulation* DNS may be possible for Reynolds numbers Re of the order up to $10^3 - 10^4$, while Re larger than say 10^5 is beyond present computational power, because the smallest eddies typically are of size $Re^{-3/4}$ and thus the required number of mesh-points for full resolution typically scales like $Re^{9/4}$. In many applications e.g. in aero- and hydrodynamics we may in fact have $Re = 10^6$, or even larger. In these cases *turbulence modeling* is needed to account for the effect of the unresolved *subgrid scales* on the computationally resolved scales. Turbulence modeling is one of classical physics outstanding open problems where today computational methods in the form of *Large Eddy Simulation LES* open new possibilities for generation of *subgrid models*.

We view the *discretization error* (resulting from using a finite element method to solve the Navier-Stokes equations) together with the *modeling error* (resulting from using a subgrid model in LES), to form the total *computational error*, which thus connects to aspect of computability. In an adaptive method both the finite element discretization (mesh) and the subgrid model will be chosen from *feed back* information from computation. With this view the subgrid model is a part of the computational procedure which thus is designed adaptively through computation (and not ad hoc a priori).

An a posteriori error estimate underlying an adaptive finite element method for the Navier-Stokes equations for computing a certain quantity of interest, involves an integral in space-time of a *discretization residual* times an associated *dual weight*, and in LES also a *modeling residual* times another associated dual weight. The dual weights are obtained by solving an associated linearized dual problem, with data depending on the quantity of interest, and contains information about error propagation in space-time. The discretization residual measures to what extent the finite element solution satisfies the Navier-Stokes equations (pointwise), the modeling residual similarly measures the error in the subgrid model used in the LES computation, and the corresponding terms in the a posteriori error estimate including the dual weights measure the effect of the residuals on the output. The size of the dual weights may vary with the flow, the output and the error norm, and indicate the relative difficulty of computing e.g. a pointwise quantity vs a global quantity such as a time-mean of a drag force. Altogether, such a posteriori error estimates may be used to adaptively choose, with respect to a certain output and tolerance level/norm, both an optimal finite element mesh leading to minimal computational work and the best of available subgrid models.

To discretize the Navier-Stokes equations to get a discrete system of equations on each time step, we use the general stabilized Galerkin/least squares space-time finite element method developed over the years together with Hughes, Tezduyar and coworkers, here referred to as the General Galerkin G^2-method. This method includes the *streamline diffusion method* on Eulerian space-time meshes, the *characteristic Galerkin method* on Lagrangian space-time meshes with orientation along particle trajectories, and *Arbitrary Lagrangian-Eulerian ALE methods* with different mesh orientation. The G^2-method constitutes a general flexible methodology for the discretization of the incompressible and compressible Navier-Stokes equations applicable to a great variety of flow problems from creeping viscous flow to slightly viscous flow, including free or moving boundaries.

With continuous piecewise polynomials in space of order p and discontinuous or continuous

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

piecewise polynomials in time of order q, we refer to this method as cG(p)dG(q) or cG(p)cG(q). In the computations presented in these notes we use cG(1)cG(1) with continuous piecewise linears in space for both velocity and pressure on tetrahedral meshes and Crank-Nicolson time stepping.

We now give a short general introduction to adaptive finite element methods based on a posteriori error estimation, and we then pass into the specific application to CFD which is the theme of this presentation, including discretization, discrete solvers, subgrid models, a posteriori error estimation, and a range of applications from laminar to turbulent flow.

1.1. Computability, predictability and adaptivity

We consider a mathematical model of the form

$$A(u) = f, \tag{1}$$

where A is a *differential operator* on functions $v = v(x)$ defined on a domain in $\mathbb{R}^3$ with x-coordinates, $f = f(x)$ is given *data*, and $u = u(x)$ is the *solution*. The model is subject to perturbations of *data* represented by $\hat{f}$, *modeling* represented by $\hat{A}$, and errors from *discretization* represented by U viewed as a finite element approximate solution to a perturbed problem

$$\hat{A}(\hat{u}) = \hat{f} \tag{2}$$

with exact solution $\hat{u}$.

Let now $g(u)$ represent a quantity of interest or output with $g(\cdot)$ a given function. We may say that the *error from data/modeling* is equal to $g(u) - g(\hat{u})$ and the *error from discretization* is equal to $g(\hat{u}) - g(U)$, and that the total error $g(u) - g(U) = g(u) - g(\hat{u}) + g(\hat{u}) - g(U)$, thus has a contribution from data/modeling and a contribution from discretization. The model perturbation $\hat{A}$ may represent (i) known perturbations of the coefficients of A, which be viewed as perturbations of given data, and then $g(u) - g(\hat{u})$ represents the toal error from data (from $\hat{f}$ and $\hat{A}$). Alternatively, $\hat{A}$ may represent (ii) a subgrid model, and then $g(u) - g(U)$ will be the total computational error with $g(u) - g(\hat{u})$ now representing the error from modeling and as before $g(\hat{u}) - g(U)$ the error from discretization. In our applications to fluid mechanics below, we will focus on (ii) with $\hat{A}$ corresponding to a turbulence model in LES.

We define a solution u of $A(u) = f$ to be *computable* with respect to a given output $g(u)$, norm $\|\cdot\|$, tolerance TOL and computational work, if the computational goal $\|g(u) - g(U)\| \leq TOL$ can be achieved with the given computational work, where U is a finite element solution of a perturbed problem $\hat{A}(\hat{u}) = f$ and thus includes errors from both discretization and modeling.

Similarly, we define a solution u of $A(u) = f$ to be *predictable* with respect to a given output $g(u)$, norm $\|\cdot\|$, tolerance $TOL > 0$ and (a suitable measure of) perturbation level, if $\|g(u) - g(\hat{u})\| \leq TOL$ with data/modeling perturbations below the perturbation level, where $\hat{u}$ solves the perturbed problem $\hat{A}(\hat{u}) = \hat{f}$.

A mathematical model with predictable and computable solutions, (including errors from data, modeling and discretization) may be useful in the sense that quantities of interest may be computed up to a tolerance with given precision of data and computational work.

If the uncertainty in data/modeling is too large, individual solutions may effectively be unpredictable, but such solutions may still be computable in the sense that the computational error in output for each specific choice of data/model may be below the chosen tolerance. In such cases, accurate computations on a set of data/models may give useful information of a statistical nature. This occurs frequently, since

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

computational work is cheap, while data (from experiments/measurement) usually is expensive. On the other hand, non-computable solutions do not seem to be useful.

An *adaptive method* for solving $A(u) = f$ includes a *feed-back process*, where the quality of computed solutions U of perturbed models $\hat{A}(\hat{u}) = \hat{f}$, are investigated with the objective of decreasing the modeling error $u - \hat{u}$ by improving the model $\hat{A}$, and/or the discretization error $\hat{u} - U$, the latter typically by appropriately modifying the local mesh size. An adaptive method is based on *a posteriori error estimates* estimating the data/modeling and discretization errors in terms of computable residuals such as $f - A(U)$ or $\hat{f} - \hat{A}(U)$, or estimated residuals in terms of the unknown solution $\hat{u}$.

Adaptive feed-back in modeling and discretization may be viewed as one aspect of *optimization* with the objective of minimizing the errors from modeling and discretization for a certain amount of computational work. Adding also aspects of optimization of solutions, which is often the main objective, one gets a full picture of solution optimization including optimization of modeling and discretization. This problem is of the same general form with now the equation $A(u) = f$ representing a Lagrange system of equations characterizing solution optimality.

1.2. A posteriori error analysis for the discretization error

We now present the key steps in the derivation of an a posteriori error estimate for the discretization error in a Galerkin finite element method for the equation $A(u) = f$ of the form: Find $U_h \in V_h$ such that $(A(U_h), v) = (f, v)$ for $v \in V_h$, where V_h is a finite dimensional subspace, on a mesh with mesh size $h = h(x)$, of a Hilbert space V with scalar product $(\cdot, \cdot)$ and norm $\|\cdot\|$, and $A : V \to V$ is Frechet differentiable with derivative $A' : V \to V$. Suppose that the quantity of interest is a linear functional of the form $g(u) = (u, \psi)$, where ψ is a given element in V, and thus that we want to bound $|g(u) - g(U_h)| = |(e, \psi)|$ with $e = u - U$. We then write

$$\begin{aligned} A(u) - A(U_h) &= \int_0^1 \frac{d}{ds} A(su + (1-s)U_h)\, ds \\ &= \int_0^1 A'(su + (1-s)U_h)\, ds\, e \equiv A'(u, U_h)e, \end{aligned}$$

and let $\varphi \in V$ be the solution to the *linearized dual problem* $(A'(u, U_h)w, \varphi) = (w, \psi)$ for all $w \in V$. Choosing $w = e$ we obtain the following *error representation*

$$(e, \psi) = (A'(u, U_h)e, \varphi) = (A(u) - A(U_h), \varphi) = (f - A(U_h), \varphi) = (R(U_h), \varphi),$$

in terms of the residual $R(U_h) = f - A(U_h)$ and the dual solution ϕ. We may then use *Galerkin orthogonality* to obtain, with $\Phi \in V_h$ an interpolant of φ satsifying an *interpolation error estimate* of the form $\|h^{-2}(\varphi - \Phi)\| \leq C_i \|D^2\varphi\|$,

$$(e, \psi) = (R(U_h), \varphi - \Phi) \leq C_i \|h^2 R(U_h)\|\, \|D^2\varphi\| \leq C_i S \|h^2 R(U_h)\|\, \|\psi\|,$$

where D^2 represents a second derivative, $h = h(x)$ respresents the mesh size, $S = \|D^2\varphi\| / \|\psi\|$ is a stability factor and C_i an interpolation constant. Normalizing ψ, we obtain an a posteriori error estimate of the form

$$|(e, \psi)| \leq C_i S \|h^2 R(U_h)\|$$

estimating the error in terms of the residual $R(U_h)$, the mesh size h and the stability and interpolation factors S and C_i. The interpolation factor C_i only depends on the finite elements used and is easy to determine. To determine S we compute the dual solution with $A(u, U_h)$ replaced by $A(U_h, U_h)$, assuming U_h approximates u pointwise sufficiently well.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

1.3. A posteriori error analysis including modeling error

We now consider a problem with subgrid scales where U_h cannot be expected to approximate the exact solution u pointwise, since u contains subgrid scales which are not resolved by the mesh h. We here view h to be the final finest mesh size obtained through an adaptive method, and thus the exact solution u contains scales which are not resolved even on the finest mesh h.

As indicated, we distinguish two different cases: In the first case the subgrid scales of the solution u originate from subgrid scales in the data (the right hand side f or the coefficients of the differential operator A including initial or boundary conditions). In this case we may obtain the modified operator $\hat{A}$ underlying the Galerkin method by direct interpolation of the coefficients of A on the mesh h. We may also seek to define the coefficients (*effective parameters*) of $\hat{A}$ on the scale h by a process of *homogenization* with the objective of increasing the precision.

In the second case the subgrid features arise even with smooth data through nonlinearities of A, as happens when a laminar flow goes turbulent and develops small scale features. This is the case of interest in this presentation, where thus the modified operator $\hat{A}$ corresponds to a turbulence model.

We thus now consider a situation where the exact solution u contains unresolvable subgrid scales and we have to aim for computing a pointwise approximation U_h on a mesh h of a local average u^h of u on the scale h, rather than u itself. We are then led to seek an equation for the average u^h to give the modified equation underlying the Galerkin method. To obtain an equation for u^h we average the equation $A(u) = f$ on the scale h and obtain an equation of the form $A(u^h) + F_h(u) = f^h$, where $F_h(u) \equiv (A(u))^h - A(u^h)$ has to be modeled in terms of u^h to give a modified equation of the form $\hat{A}(\hat{u}) = A(\hat{u}) + \hat{F}_h(\hat{u}) = f^h$, where now $\hat{u}$ is an approximation of u^h and $\hat{F}_h(\hat{u}) \approx F_h(u)$ represents the subgrid model.

We then solve the Galerkin equation: find $U_h \in V_h$ such that $(A(U_h) + \hat{F}_h(U_h), v) = (f, v)$, for all $v \in V_h$, and expect U_h to be a pointwise approximation of u^h. As a consequence, the appropriate linearized dual problem in this case takes the form $(A'(u^h, U_h)w, \varphi) = (w, \psi)$, for all $w \in V$, where $A'(u^h, U_h)e = A(u^h) - A(U_h)$, that is, we linearize the dual problem at u^h and not at the exact solution u, as above. When we solve the dual problem numerically, we replace u^h by U_h, which we anticipate to be possible because U_h may approximate u^h pointwise. We then get the following error representation for the error $e = u^h - U_h$:

$$\begin{aligned}
(e, \psi) &= (u^h - U_h, \psi) = (A'(u^h, U_h)e, \varphi) = (A(u^h) - A(U_h), \varphi) \\
&= (f^h - F_h(u) - A(U_h), \varphi) = (f^h - A(U_h) - \hat{F}_h(U_h), \varphi) \\
&+ (\hat{F}_h(U_h) - F_h(u), \varphi) = (R_D(U_h), \varphi) + (R_M(u, U_h), \varphi),
\end{aligned}$$

where $R_D(U_h) = f^h - A(U_h) - \hat{F}_h(U_h)$ is a computable numerical residual related to the discretization error in solving the equation $A(\hat{u}) + \hat{F}_h(\hat{u}) = f^h$, and $R_M(u, U_h) = \hat{F}_h(U_h) - F_h(u)$ is a modeling residual related to the error in the subgrid model $\hat{F}_h$. The modeling residual $R_M(u, U_h)$ is not directly computable, because of the presence of u, but has to be estimated somehow. We retirn to this problem below. We further note that we do not have any Galerkin orthogonality property for the modeling residual $R_M(u, U_h)$.

The linearized dual problem is independent of both $F_h(u)$ and $\hat{F}_h$, but in some cases there might be advantages in including the subgrid model $\hat{F}_h$ in the dual problem. We then get a linearized dual problem of the form: $(\hat{A}'(u^h, U_h)w, \varphi) = (w, \psi)$, for all $w \in V$, where $\hat{A}'(u^h, U_h)e =$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

$\hat{A}(u^h) - \hat{A}(U_h)$, and we get the following error representation for the error $e = u^h - U_h$:

$$\begin{aligned}(e,\psi) &= (u^h - U_h,\psi) = (\hat{A}'(u^h,U_h)e,\varphi) = (\hat{A}(u^h) - \hat{A}(U_h),\varphi)\\ &= (A(u^h) + \hat{F}_h(u^h) - A(U_h) - \hat{F}_h(U_h),\varphi)\\ &= (f^h - F_h(u) + \hat{F}_h(u^h) - \hat{F}_h(U_h) - A(U_h),\varphi)\\ &= (f^h - A(U_h) - \hat{F}_h(U_h),\varphi) + (\hat{F}_h(u^h) - F_h(u),\varphi)\\ &= (R_D(U_h),\varphi) + (R_M(u,U_h),\varphi),\end{aligned}$$

where $R_D(U_h) = f^h - A(U_h) - \hat{F}_h(U_h)$ is a computable numerical residual related to the discretization error in solving the equation $A(\hat{u}) + \hat{F}_h(\hat{u}) = f^h$, and $R_M(u,U_h) = \hat{F}_h(u^h) - F_h(u)$ is a modeling residual related to the error in the subgrid model $\hat{F}_h$. Including the subgrid model $\hat{F}_h$ in the dual problem may be preferable if this results in a regularization of the dual solution.

1.4. A posteriori error analysis for stabilized Galerkin methods

Below we will use a stabilized Galerkin method for the numerical solution of the Navier-Stokes equations. We now address i general terms the modifications in the derivation of the a posteriori error estimates motivated by the stabilization. We first recall that stabilized Galerkin methods may be obtained by applying a standard Galerkin method to a properly modified equation $\hat{A}(\hat{u}) = \hat{f}$. For example, a Galerkin least squares stabilized method is of the form: Find $U_h \in V_h$ such that

$$(A(U_h), v + \delta A(v)) = (f, v + \delta A(v)) \tag{3}$$

for all $v \in V_h$, where δ is a stabilization parameter. For simplicity we assume that A is a linear operator in this section. We may alternatively consider this problem as a standard Galerkin method for a modified equation $\hat{A}(\hat{u}) = A(\hat{u}) + \delta A^* A(\hat{u}) = f + \delta A^* f = \hat{f}$, where '$*$' denotes the adjoint, that is find $U_h \in V_h$ such that

$$(A(U_h) + \delta A^* A(U_h), v) = (f + \delta A^* f, v) \tag{4}$$

for all $v \in V_h$. In the a posteriori error analysis, we introduce a dual problem of the form: Find $\varphi \in V$ such that $(\hat{A}'(u,U_h)w,\varphi) = (w,\psi)$, for all $w \in V$, where $\hat{A}'(u,U_h)e = \hat{A}(u) - \hat{A}(U_h)$, which leads to an error representation for $e = u - U_h$ of the form

$$\begin{aligned}(e,\psi) &= (u - U_h,\psi) = (\hat{A}'(u,U_h)e,\varphi) = (\hat{A}(u) - \hat{A}(U_h),\varphi)\\ &= (f - A(U_h) + \delta A^*(f - A(U_h)),\varphi),\end{aligned}$$

where we may use the Galerkin orthogonality from (4) to subtract an interpolant of φ and estimate the interpolation error.

Including subgrid modeling with the averaged exact solution u^h satisfying $A(u^h) + F_h(u) = f^h$, stabilized Galerkin method for the equation $A(\hat{u}) + \hat{F}_h(\hat{u}) = f^h$ reads: find $U_h \in V_h$ such that

$$(A(U_h) + \hat{F}_h(U_h) + \delta A^*(A(U_h) + \hat{F}_h(U_h)), v) = (f + \delta A^* f, v) \tag{5}$$

for all $v \in V_h$. We are then led to an error representation for $e = u^h - U_h$ of the form

$$\begin{aligned}(e,\psi) &= (u^h - U_h,\psi) = (\hat{A}'(u^h,U_h)e,\varphi) = (\hat{A}(u^h) - \hat{A}(U_h),\varphi)\\ &= (f^h - F_h(u) - A(U_h) + \delta A^*(f - F_h(u) - A(U_h)),\varphi)\\ &= (f^h - \hat{F}_h(U_h) - A(U_h) + \delta A^*(f^h - \hat{F}_h(U_h) - A(U_h)),\varphi)\\ &\quad + (\hat{F}_h(U_h) - F_h(u) + \delta A^*(\hat{F}_h(U_h) - F_h(u)),\varphi)\\ &= (R_D(U_h),\varphi) + (R_M(u,U_h),\varphi),\end{aligned}$$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

where we may use the Galerkin orthogonality of (5) to subtract an interpolant of φ in the first term and estimate the corresponding interpolation error.

2. The incompressible Navier-Stokes equations

The incompressible Navier-Stokes equations expressing conservation of momentum and incompressibility of a unit density constant temperature Newtonian fluid with constant *kinematic viscosity* $\nu > 0$ enclosed in a volume Ω in $\mathbb{R}^3$, take the form: find (u, p) such that

$$\begin{aligned} D_{u,t}u - \nu\Delta u + \nabla p &= f && \text{in } \Omega \times I, \\ \operatorname{div} u &= 0 && \text{in } \Omega \times I, \\ u &= w && \text{on } \partial\Omega \times I, \\ u(\cdot, 0) &= u^0 && \text{in } \Omega, \end{aligned} \tag{6}$$

where $u(x,t) = (u_i(x,t))$ is the *velocity* vector and $p(x,t)$ the *pressure* of the fluid at (x,t), and f, w, u^0, $I = (0,T)$, is a given driving force, Dirichlet boundary data, initial data and time interval, respectively. Further,

$$D_{u,t}v \equiv \dot{v} + (u \cdot \nabla)v \tag{7}$$

is the *particle derivative* of $v(x,t)$ measuring the rate of change $\frac{d}{dt}v(x(t),t)$ of $v(x(t),t)$ along the trajectory $x(t)$ of a fluid particle with velocity u, satisfying $\dot{x}(t) = u(x(t),t)$, where as usual $\dot{v} = \partial v/\partial t$. The quantity $\nu\Delta u - \nabla p$ represents the total fluid force, and may alternatively be expressed as

$$\nu\Delta u - \nabla p = \operatorname{div} \sigma(u,p), \tag{8}$$

where $\sigma(u,p) = (\sigma_{ij}(u,p))$ is the *stress tensor*, with components $\sigma_{ij}(u,p) = 2\nu\epsilon_{ij}(u) - p\delta_{ij}$, composed of the *stress deviatoric* $2\nu\epsilon_{ij}(u)$ with zero trace and an isotropic pressure: Here $\epsilon_{ij}(u) = (u_{i,j} + u_{j,i})/2$ is the *strain tensor*, with $u_{i,j} = \partial u_i/\partial x_j$, and δ_{ij} is the usual Kronecker delta, the indices i and j ranging from 1 to 3. A Neumann type boundary condition, corresponding to the boundary stress being prescribed, takes the form $\sigma \cdot n = g$, where $(\sigma \cdot n)_i = \sum_j \sigma_{ij}n_j$ and $g = (g_i)$ is a given boundary stress with g_i the force component in the x_i-direction.

In the model (6) we assume that the *temperature* T is constant. In the general case with variable density ρ and temperature T, (6) is modified by replacing $D_{u,t}$ by $\rho D_{u,t}$, and adding the following equations expressing conservation of mass and energy:

$$\begin{aligned} D_{u,t}\rho &= 0 && \text{in } \Omega \times I, \\ D_{u,t}T - \nabla \cdot (\mu\nabla T) &= F && \text{in } \Omega \times I, \end{aligned} \tag{9}$$

together with boundary and initial conditions, where μ is a heat conduction coefficient and F a heat source, assuming the heat capacity is equal to one. We note that since $\nabla \cdot u = 0$, we have $D_{u,t}\rho = \dot{\rho} + \nabla \cdot (\rho u) = 0$, which is the usual equation expressing mass conservation.

We assume that (6) is normalized so that the reference velocity and typical length scale are both equal to one. The Reynolds number Re is then equal to ν^{-1}. Of course, the specification of the length scale may not be very obvious and thus the Reynolds number may not have a very precise quantitative meaning.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

3. Discretization: General Galerkin G^2

In this section we present the general space-time Galerkin least squares stabilized finite element method, referred to as the General Galerkin G^2- method, for the incompressible Navier-Stokes equations (6). This method includes the *streamline diffusion method* on Eulerian space-time meshes, the *characteristic Galerkin method* on Lagrangian space-time meshes with orientation along particle trajectories, and *Arbitrary Lagrangian-Eulerian ALE methods* with different mesh orientation. Further, the least-squares stabilizations present in the G^2-method, does take care of the two difficulties traditionally met in the discretization of the incompressible Navier-Stokes equations, namely

- instabilities from Eulerian discretization of convection terms,
- pressure instabilities in equal order interpolation of velocity and pressure.

Altogether, G^2 offers a general flexible methodology for the discretization of the incompressible Navier-Stokes equations applicable to a great variety of flow problems from creeping viscous flow to slightly viscous flow, including free or moving boundaries.

Let $0 = t_0 < t_1 < ... < t_N = T$ be a sequence of discrete time steps with associated time intervals $I_n = (t_{n-1}, t_n]$ of length $k_n = t_n - t_{n-1}$ and space-time slabs $S_n = \Omega \times I_n$, and let $W_n \subset H^1(\Omega)$ be a finite element space consisting of continuous piecewise polynomials of degree p on a mesh $\mathcal{T}_n = \{K\}$ of mesh size $h_n(x)$ with W_{0n} the functions in W_n vanishing on Γ. To define the G^2-method for (6) with homogeneuos Dirichlet boundary conditions for the velocity ($w = 0$), let for a given velocity field β on $S_n = \Omega \times I_n$ vanishing on $\Gamma \times I_n$, the particle paths $x(\bar{x}, \bar{t})$ be defined by

$$\begin{aligned} \frac{dx}{d\bar{t}} &= \beta(x, \bar{t}) \quad \bar{t} \in I_n, \\ x(\bar{x}, t_n) &= \bar{x}, \quad \bar{x} \in \Omega, \end{aligned} \tag{10}$$

and introduce the corresponding mapping $F_n^\beta : S_n \to S_n$ defined by $(x, t) = F_n^\beta(\bar{x}, \bar{t}) = (x(\bar{x}, \bar{t}), \bar{t})$, where $x = x(\bar{x}, \bar{t})$ satisfies (10). Define for a given $q \geq 0$, the spaces

$$\begin{aligned} \bar{V}_n^\beta &= \{\bar{v} \in H^1(S_n)^3 : \bar{v}(\bar{x}, \bar{t}) = \sum_{j=0}^{q} (\bar{t} - t_n)^j U_j(\bar{x}),\, U_j \in [W_{0n}]^3\}, \\ \bar{Q}_n^\beta &= \{\bar{q} \in H^1(S_n) : \bar{q}(\bar{x}, \bar{t}) = \sum_{j=0}^{q} (\bar{t}_n - t_n)^j q_j(\bar{x}),\, q_j \in W_n\}, \end{aligned}$$

together with their analogs in (x, t)-coordinates:

$$V_n^\beta = \{v : \bar{v} \in \bar{V}_n^\beta\}, \qquad Q_n^\beta = \{q : \bar{q} \in \bar{Q}_n^\beta\}, \tag{11}$$

where $v(x, t) = \bar{v}(\bar{x}, \bar{t})$ and $q(x, t) = \bar{q}(\bar{x}, \bar{t})$. Defining finally $V^\beta \times Q^\beta = \prod_n V_n^\beta \times Q_n^\beta$, we can now formulate the G^2-method as follows: Find $(U, P) \in V^\beta \times Q^\beta$, such that for $n = 1, 2, ..., N$,

$$\begin{aligned} &(\dot{U} + (U \cdot \nabla)U, v)_n - (P, \operatorname{div} v)_n + (q, \operatorname{div} U)_n + (2\nu\epsilon(U), \epsilon(v))_n \\ &+ (\delta_1 a(U; U, P), a(U; v, q))_n + (\delta_2 \operatorname{div} U, \operatorname{div} v)_n + ([U^{n-1}], v_+^{n-1}) \\ &= (f, v + \delta_1 a(U; v, q))_n \quad \forall (v, q) \in V_n^\beta \times Q_n^\beta, \end{aligned} \tag{12}$$

where $a(w; v, q) = D_{w,t} v + \nabla q - \nu \Delta v$ with the Laplacian defined elementwise, $\delta_1 = \frac{1}{2}(k_n^{-2} + |U|^2 h_n^{-2})^{-1/2}$ in the convection-dominated case $\nu < U h_n$ and $\delta_1 = \kappa_1 h^2$ otherwise, $\delta_2 = \kappa_2 h$ if

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

$\nu < Uh_n$ and $\delta_2 = \kappa_2 h^2$ otherwise, with κ_1 and κ_2 positive constants of unit size, and

$$(v,w)_n = \int_{I_n} (v,w)dt, \quad (v,w) = \sum_{K\in\mathcal{T}_n} \int_K v\cdot w\,dx,$$
$$(\epsilon(v),\epsilon(w)) = \sum_{i,j=1}^{3} (\epsilon_{ij}(v),\epsilon_{ij}(w)).$$

Further, $[v^n] = v_+^n - v_-^n$ is the jump across the time level t_n with $v_\pm^n$ the limit from $t > t_n/t < t_n$. In the Eulerian *streamline diffusion method* we choose $\beta = 0$, which means that the mesh does not move in time. The *characteristic Galerkin method* is obtained by choosing $\beta = U$ (and then $\delta_1 = \kappa_1 h^2$), which means that the mesh moves with the fluid particles. We may also choose β differently which gives various versions of ALE-methods, with the mesh and particle velocity being (partly) different; for example we may move the mesh with the particle velocity at a free boundary, while allowing the mesh to move differently inside the domain.

The variational formulation (12) with $\delta_1 = \delta_2 = 0$ is obtained by multiplying the momentum equation by v, integrating over S_n including integration by parts, and adding the incompressibility equation multiplied by q and integrating over S_n. Choosing δ_1 and δ_2 positive as indicated introduces stabilizing least-squares terms. Note that the viscous term $(2\nu\epsilon(U),\epsilon(v))_n$ may alternatively occur in the form $(\nu\nabla U,\nabla v)_n = \sum_{i=1}^3 (\nu\nabla U_i,\nabla v_i)_n$. In the case of Dirichlet boundary conditions the corresponding variational formulations will be equivalent, but not so in the case of Neumann boundary conditions, see below. Note finally that we may write the term $-(P,\operatorname{div} v)$ alternatively in the form $(\nabla P, v)$ if v vanishes on the boundary.

In extreme situations with very large velocity gradients, we may add residual dependent *shock-capturing artificial viscosity*, replacing ν by $\hat\nu = \max(\nu, \kappa_3|R(U,P)|h^2)$, where $R(U,P) = \sum_{i=1}^4 R_i(U,P)$ with

$$\begin{array}{rcl} R_1(U,P) &=& |\dot U + U\cdot\nabla U + \nabla P - f - \nu\Delta U|, \\ R_2(U,P) &=& \nu D_2(U), \\ R_3(U,P) &=& |[U^{n-1}]|/k_n \quad \text{on } S_n, \\ R_4(U,P) &=& |\operatorname{div} U|, \end{array} \tag{13}$$

where

$$D_2(U)(x,t) = \max_{y\in\partial K} (h_n(x))^{-1} |[\frac{\partial U}{\partial n}(y,t)]| \tag{14}$$

for $x \in K$, with $[\cdot]$ the jump across the element edge ∂K, and κ_3 is a positive constant of unit size. Note that $R_1(U,P)$ is defined elementwise and that with piecewise linears in space, the Laplacian ΔU is zero. In the computations presented below, we chose $\kappa_3 = 0$ corresponding to shutting off the artificial viscosity. Note that $R_1(U,P) + R_2(U,P)$ bounds the residual of the momentum equation, with the Laplacian term bounded by the second order difference quotient $D_2(U)$ arising from the jumps of normal derivatives across element boundaries.

The special case of the Stokes equations is of course obtained omitting the nonlinear terms $(U\cdot\nabla)U$ and $(U\cdot\nabla)v$, and setting $\delta_1 = \kappa_1 h^2$, $\delta_2 = \kappa_2 h^2$. This method contains the pressure stabilizing term $(\delta_1\nabla P,\nabla q)$, which corresponds to a weighted Laplacian equation for the pressure in terms of the velocity.

Since in the local Lagrangean coordinates $(\bar x,\bar t)$ on each slab S_n with $\beta = U$,

$$\frac{\partial \bar U}{\partial \bar t} \equiv \frac{\partial}{\partial \bar t} U(x(\bar x,\bar t),\bar t) = \dot U + U\cdot\nabla U,$$

the convection term $U \cdot \nabla U$ effectively dissappears in the characteristic Galerkin method, when expressed in the characteristic coordinates $(\bar{x}, \bar{t})$, and thus the discrete equations on each time step effectively correspond to a Stokes problem.

The order of the G^2-method with polynomials of degree p in space/time is generally $p + 1/2$, see Eriksson *et al.*, 1996. The time stepping method in (12) is dG(q), the discontinuous Galerkin method with piecewise polynomials of order q, which is of order $2q + 1$ seen as an ODE-solver.

3.1. Neumann boundary conditions

If we change to Neumann boundary conditions $\sigma \cdot n = g$ on a part Γ_1 of the boundary Γ, then W_{0n} is chosen to be the functions in W_n vanishing on the remaining Dirichlet part Γ_0 of the boundary, and the right hand side is supplemented with an integral over Γ_1 of $g \cdot v$. As usual this implements the Neumann boundary condition in weak form through the presence of the term $(-P, \operatorname{div} v) + (2\nu\epsilon(U), \epsilon(v)) = (\sigma, \epsilon(v))$ on the left hand side, which when integrated by parts generates an integral over Γ_1 of $(\sigma \cdot n) \cdot v$. If the viscous term appears in the form $(\nu \nabla U, \nabla v)_n$ the corresponding Neumann boundary condition has the form $\nu \frac{\partial u}{\partial n} - pn = 0$, where $\frac{\partial u}{\partial n}$ is the derivative in the unit outward normal direction n.

3.2. Outflow boundary conditions

To simulate an outflow boundary condition we may use a Neumann condition with $g = 0$ corresponding to a zero force at outflow, simulating outflow into a large empty reservoir. The alternative condition $\nu \frac{\partial u}{\partial n} - pn = 0$ acts slightly differently as an approximation of a *transparent* outflow boundary condition attempting to let the flow leave the domain with minimal obstruction.

3.3. The Eulerian cG(1)dG(0) method

We now consider the the G^2-method (12) with $p = 1$, $q = 0$ and $\beta = 0$ for (6), which is the Eulerian cG(1)dG(0) method with continuous piecewise linears in space (cG(1)) and piecewise constants in time (dG(0)) corresponding to the backward Euler method. We then seek an approximate velocity $U(x, t)$ such that $U(x, t)$ is continuous and piecewise linear in x for each t, and $U(x, t)$ is piecewise constant in t for each x. Similarly, we seek an approximate pressure $P(x, t)$ which is continuous piecewise linear in x and piecewise constant in t. More precisely, we seek $U^n \in V_n^0 = W_{0n}^3$ and $P^n \in Q_n^0 = W_n$ for $n = 1, ..., N$, and we define

$$
\begin{aligned}
U(x,t) &= U^n(x) \quad x \in \Omega, \quad t \in (t_{n-1}, t_n], \\
P(x,t) &= P^n(x) \quad x \in \Omega, \quad t \in (t_{n-1}, t_n].
\end{aligned}
\tag{15}
$$

We can now write the cG(1)dG(0) method without stabilization as follows: For $n = 1, ..., N$, find $(U^n, P^n) \in V_n^0 \times Q_n^0$ such that

$$
\begin{aligned}
&(\frac{U^n - U^{n-1}}{k_n}, v) + (U^n \cdot \nabla U^n + \nabla P^n, v) + (\nabla \cdot U^n, q) \\
&+ (\nu \nabla U^n, \nabla v) = (f^n, v) \quad \forall (v, q) \in V_n^0 \times Q_n^0,
\end{aligned}
\tag{16}
$$

The cG(1)dG(0) method with δ_1-stabilization takes the form: For $n = 1, ..., N$, find $(U^n, P^n) \in V_n^0 \times Q_n^0$ such that

$$
\begin{aligned}
&(\frac{U^n - U^{n-1}}{k_n}, v) + (U^n \cdot \nabla U^n + \nabla P^n, v + \delta_1(U^n \cdot \nabla v + \nabla q)) + (\nabla \cdot U^n, q) \\
&+ (\nu \nabla U^n, \nabla v) = (f^n, v + \delta_1(U^n \cdot \nabla v + \nabla q)) \quad \forall (v, q) \in V_n^0 \times Q_n^0,
\end{aligned}
\tag{17}
$$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

where $\delta_1 = \frac{1}{2}(k_n^{-2} + |U|^2 h_n^{-2})^{-1/2}$ in the convection-dominated case $\nu < Uh_n$. Note that if $k \approx \frac{h}{U}$, which is a natural choice of time step respecting a CFL-condition, then $\delta_1 \approx \frac{h}{U}$. Note that the stabilized form of the cG(1)dG(0) method is obtained by replacing v by $v + \delta_1(U^n \cdot \nabla v + \nabla q)$ in the terms $(U^n \cdot \nabla U^n + \nabla P^n, v)$ and (f^n, v). In principle, we should make the replacement throughtout, but in the present case of the cG(1)dG(0), only the indicated terms get involved because of the low order of the approximations. The perturbation in the stabilized method is of size δ_1, and thus the stabilized method has the same order as the original method (first order in h if $k \sim h$).

Letting v vary in (17) while choosing $q = 0$, we get the following equation (the discrete momentum equation):

$$\begin{aligned}(\frac{U^n - U^{n-1}}{k_n}, v) + (U^n \cdot \nabla U^n + \nabla P^n, v + \delta_1 U^n \cdot \nabla v)\\ + (\nu \nabla U^n, \nabla v) = (f^n, v + \delta_1 U^n \cdot \nabla v) \quad \forall v \in V_n^0,\end{aligned} \tag{18}$$

and letting q vary while setting $v = 0$, we get the following discrete "pressure equation"

$$(\delta_1 \nabla P^n, \nabla q)) = -(\delta_1 U^n \cdot \nabla U^n, \nabla q) - (\nabla \cdot U^n, q) + (\delta f^n, \nabla q) \quad \forall q \in Q_n^0. \tag{19}$$

The cG(1)dG(0) has a backward Euler first order accurate time stepping, and thus in general is too dissipative.

3.4. *The Eulerian cG(1)cG(1) method*

We now present the a cG(1)cG(1) variant of the above cG(1)dG(0) method using the continuous Galerkin method cG(1) in time instead of dG(0). With cG(1) in time the trial functions are continuous piecewise linear and the test functions piecewise constant in time. The cG(1)cG(1) variant with δ_1-stabilization reads: For $n = 1, ..., N$, find $(U^n, P^n) \in V_n^0 \times Q_n^0$ such that

$$\begin{aligned}(\frac{U^n - U^{n-1}}{k_n}, v) + (\hat{U}^n \cdot \nabla \hat{U}^n + \nabla P^n, v + \delta_1(\hat{U}^n \cdot \nabla v + \nabla q)) + (\nabla \cdot \hat{U}^n, q)\\ + (\nu \nabla \hat{U}^n, \nabla v) = (f^n, v + \delta_1(\hat{U}^n \cdot \nabla v + \nabla q)) \quad \forall (v, q) \in V_n^0 \times Q_n^0,\end{aligned} \tag{20}$$

where $\hat{U}^n = \frac{1}{2}(U^n + U^{n-1})$. This method corresponds to a second order accurate Crank-Nicolson time-stepping, but the stabilization suffers from an inconsistency up to the term $\delta_1 \dot{u}$ resulting from the piecewise constancy of the test functions. The inconsistency seems to be be acceptable unless $\dot{u}$ is large, and we use cG(1)cG(1) in the computations presented below. The Eulerian cG(1)dG(1)-method would have consistent stabilization, but has two degrees of freedom in time per time step and thus twice as many degrees of freedom.

4. Discrete solvers

The cG(1)cG(1)-method with δ_1-stabilization leads to a system of the following principal form in each step of an outer fixed point iteration with the convection velocity being given from the previous iteration:

$$\begin{aligned}AU^n + k_n BP^n = k_n F^n,\\ -B^\top U^n + CP^n = G^n,\end{aligned} \tag{21}$$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

where $A = M_n + k_n N_n - k_n \nu \Delta_n$ with M_n a mass matrix, N_n representing a discrete analog of the convection term with frozen velocity from the previous iteration, Δ_n is a discrete Laplacian, B is a discrete gradient, $B^\top$ a discrete divergence, and $C = -\delta_1 \Delta_n$. In the computations presented below we solved this system using a fixed point inner iteration, where we first solve for $P^{n,j+1}$ in terms of $U^{n,j}$ from the equation

$$CP^{n,j+1} = G^n + B^\top U^{n,j}$$

using a multigrid method, and then solve for $U^{n,j+1}$ from the equation

$$AU^{n,j+1} = k_n F^n - k_n BP^{n,j+1}$$

using GMRES. The inner iteration converges if k_n/δ_1 is small enough. Since typically $\delta_1 \approx h_n/U^n$, we need $\frac{U^n k_n}{h_n}$ to be small enough, which is a CFL-like condition.

We may also apply GMRES directly to the equation $AU^n + k_n BP^n = k_n F^n$ with P^n solved in terms of U^n from the equation $B^\top U^n + CP^n = G^n$ using multigrid. The number of GMRES iterations would then depend on the condition number of the matrix $M_n + k_n N_n - k_n \nu \Delta_n + \frac{k_n}{\delta_1} B \Delta_n^{-1} B^\top$, which is bounded with k_n/h_n, $k_n \nu / h_n^2$ and k_n/δ_1.

In both variants the full iterative procedure converges in a few iterations in our typical applications of non-stationary high Reynolds number flow with k_n/h_n and k_n/δ_1 bounded by 1.

5. A posteriori error estimation

We now prove an a posteriori error estimate for (16) following the general methodology presented above, where we aim at error control of $(e(T), \psi)$ with $e = u - U$ and $\psi \in [L_2(\Omega)]^3$ is a given function. We introduce the following linearized dual problem: Find $(\varphi, \theta) \in L_2(I; [H_0^1(\Omega)]^3 \times L_2(\Omega)) \equiv W$ such that in $Q = \Omega \times (0, T)$

$$\begin{aligned}
-\dot{\varphi} - (u \cdot \nabla)\varphi + \nabla U \cdot \varphi + \nabla \theta - \nu \Delta \varphi &= 0 \qquad \text{in } Q \\
\operatorname{div} \varphi &= 0 \qquad \text{in } Q \\
\varphi &= 0 \qquad \text{on } \Gamma \times I, \\
\varphi(\cdot, T) &= \psi \qquad \text{in } \Omega,
\end{aligned} \tag{22}$$

where $(\nabla U \cdot \varphi)_j = U_{,j} \cdot \varphi$. Multiplying the first equation by e, integrating over Q together with integration by parts, using that $(u \cdot \nabla)u - (U \cdot \nabla)U = (u \cdot \nabla)e + (e \cdot \nabla)U$, and assuming that $e(0) = 0$, gives

$$\begin{aligned}
(e(T), \psi) &= \sum_{n=0}^{N} \{(-\dot{\varphi} - (u \cdot \nabla)\varphi + \nabla U \cdot \varphi, e)_n + (\nabla \theta, e)_n + (\nu \nabla \varphi, \nabla e)_n\} \\
&= \sum_{n=1}^{N} \{(\varphi, e_t)_n + ((u \cdot \nabla)e, \varphi)_n + ((e \cdot \nabla)U, \varphi)_n \\
&\quad - (\theta, \operatorname{div} e)_n + (\nu \nabla \varphi, \nabla e)_n - (p - P, \operatorname{div} \varphi)_n\} + \sum_{n=1}^{N} ([U^{n-1}], \varphi_+^{n-1})
\end{aligned}$$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

$$
\begin{aligned}
&= \sum_{n=1}^{N} \{(\dot{u} + u\cdot\nabla u + \nabla p, \varphi)_n + (\nu\nabla u, \nabla\varphi)_n - (\dot{U} + U\cdot\nabla U + \nabla P, \varphi)_n \\
&- (\nu\nabla U, \nabla\varphi)_n + (\theta, \operatorname{div} U)_n\} + \sum_{n=1}^{N}([U^{n-1}], \varphi_+^{n-1}) \\
&= -\sum_{n=1}^{N}\{(\dot{U} + U\cdot\nabla U + \nabla P - f, \varphi - \Phi)_n \\
&- (\nu\nabla U, \nabla(\varphi - \Phi))_n + (\operatorname{div} U, \theta - \Theta)_n\} + \sum_{n=1}^{N}([U^{n-1}], \varphi_+^{n-1} - \Phi_+^{n-1}).
\end{aligned}
$$

Estimating now the interpolation errors $\varphi - \Phi$ and $\theta - \Theta$, and recalling the definition (13), we obtain an estimate of the form

$$
\begin{aligned}
|(e(T), \psi)| \leq \sum_{i=1}^{3} \int_Q R_i(U)(Ch^m|D^m\varphi| + Ck|\dot{\varphi}|)dx\,dt \\
+ \int_Q R_4(U)(Ch^m|D^m\theta| + Ck|\dot{\theta}|)dx\,dt,
\end{aligned}
\tag{23}
$$

for $m = 1, 2$, where D^m measures derivatives with respect to x of order m, and C represents interpolation constants. To get a concrete a posteriori error estimate, we solve the dual problem numerically and compute approximations of the derivatives of the dual solution involved. With adaptive choice of meshing, we choose $h_n(x)$ and k_n from a principle of equidistribution with the derivatives of the dual solution entering as weights.

5.1. *A posteriori error estimates with stability factors*

In the a posteriori error estimate (23), certain derivatives of the the dual solution appear as a weights in a space time integral over the residuals R_i. We may estimate these space-time integral in various ways; for instance using Cauchy's inequality with $L_2(I; (L_2(\Omega))$-norms $\|\cdot\|_I$ with $I = [0, T]$, we obtain an a posteriori error estimate of the form

$$
\begin{aligned}
|(e(T), \psi)| \leq C\|\dot{\varphi}\|_I \sum_{i=1}^{3} \|kR_i(U, P)\|_I + C\|D^2\varphi\|_I \sum_{i=1}^{3} \|h^2 R_i(U, P)\|_I \\
+ C\|\dot{\theta}\|_I \|kR_4(U, P)\|_I + C\|D^1\theta\|_I \|hR_4(U, P)\|_I
\end{aligned}
\tag{24}
$$

with $\|\dot{\varphi}\|_I$, $\|D^2\varphi\|_I$, $\|\dot{\theta}\|_I$ and $\|D^1\theta\|_I$ entering as multiplicative *stability factors*. We may alternatively use the max-norm in time for the residuals and the L_1-norm in time for the dual solution.

The size of the stability factor of course directly couples to computability: as the stability factors grow large the residuals have to be very small and the computational work increases.

Obviously (23) is a sharper estimate than (24), and thus better suited for use in an adaptive algorithm. On the other hand, the estimate (24) is useful to quantify computability of the a certain quantity in a certain problem. Large stability factors indicate that a large computational effort is needed, while small stability constants indicate that the problem can be solved with a minor computational effort.

The dependence of the stability factors on the length T of the simulation of course couples to computability; if the stablity factors grow quickly in time then only short time simulation is

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

possible, while if the stability factors grow slowly then long-time simulation is feasible. In our related Encyclopedia presentation Eriksson *et al.*, 2003 on Parabolic problems we propose to use the time-dependence of stability factors as a means of classification: In particular, we use the term *parabolic* to identify a problem with the stability factors being bounded for all T (up to possibly a slow logarithmic growth). Roughly speaking this connects to *diffusion-dominated convection-diffusion-reaction* problems, while in *convection-dominated* problems we may meet a linear (or faster) growth, and for highly demanding problems, such as the computation of a point value in a turbulent flow, the stability factors may locally grow exponentially, see Eriksson *et al.*, 2001 for a study of the Lorenz system as a simple model for the Navier-Stokes equations.

Of particular interest is the growth of stability factors for mean-value quantities in turbulent flow. In Section 8.1 we compute stability factors for different outputs in laminar an turbulent flow with direct coupling to computability. A fundamental observation from these studies is that mean-values in turbulent flow appear to be computable with desk-top computational power, thus indicating very good prospects for CFD.

5.2. Computation of lift and drag

Suppose we want to compute an approximation of the quantity

$$N(\sigma(u,p)) = \frac{1}{T}\int_0^T \int_{\Gamma_1} \sum_{i,j=1}^{3} \sigma_{i,j}(u,p) n_j \psi_i \, ds, \tag{25}$$

where $\Gamma = \Gamma_0 \cup \Gamma_1$ is a decomposition of the boundary Γ, and $\psi = (\psi_i)$ is a given function on Γ_1, and (u,p) solves (6). The quantity $N(\sigma(u,p))$ may represent the mean value over $[0,T]$ of the force in the direction ψ on a body with boundary Γ_1 immersed in a flow. If ψ is in the direction of the mean flow we get tha drag force, and if ψ is in a direction perpendicular to the mean flow we get the lift force in that direction.

Instead of directly using (25), we may use the following alternative expression with the idea of increasing the precision, see Giles *et al.*, 1997,

$$N(\sigma(u,p)) = \frac{1}{T}\int_0^T ((\dot{u} + u \cdot \nabla u - f, \psi) - (p, \operatorname{div} \psi) + (2\nu\epsilon(u), \epsilon(\psi)))\, dt, \tag{26}$$

where ψ now is an extension of the given ψ into Ω with $\psi = 0$ on Γ_0, which follows by integrating by parts in the last two terms, and using the momentum equation for the solution (u,p). We note that the representation does not depend on the particular extension of ψ being used. We are thus led to approximate $N(\sigma(u,p))$ by the quantity

$$N_h(\sigma(U,P)) = \frac{1}{T}\int_0^T ((\dot{U} + U \cdot \nabla U - f, \Psi) - (P, \operatorname{div} \Psi) + (2\nu\epsilon(U), \epsilon(\Psi)))\, dt$$

where Ψ is a finite element function satisfying $\Psi = \psi$ on Γ_1, assuming ψ is the restriction to Γ_1 of a finite element function and (U,P) is a finite element solution of (6). Again, the discrete momentum equation shows that $N_h(\sigma(U,P))$ is independent of the extension Ψ. Let now (φ,θ) be the solution of the linearized dual problem (22) with $\varphi(T) = 0$ and $\varphi(\cdot,t) = \psi$ on Γ_1 and $\varphi(\cdot,t) = 0$ on Γ_0 for all $t \in [0,T]$. Reasoning as above we then obtain an a posteriori error estimate for $N(\sigma(u,p)) - N_h(\sigma(U,P))$ of the same form as (23)-(24).

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

6. Adaptive finite element methods for laminar flow

A posteriori error estimates of the type presented above; an integral of a residual weighted by the solution of a dual problem, have been shown to generate effective error indicators as well as sharp stopping criterions for adaptive finite element methods in the case of stationary 2d flows, see Becker and Rannacher, 2001. In this section we present a couple of examples to illustrate the ideas extended to steady and unsteady 3d flow.

6.1. Computation of the drag force

We consider the problem of computing the drag force on a cylinder with square cross-section immersed into a viscous incompressible flow, and compare with results from the collection of benchmark computations for laminar flow around a cylinder in 2d and 3d presented in Schäfer and Turek, 1996.

6.1.1. Steady flow We start considering steady 3d flow around a cylinder with square cross-section $D \times D$ with $D = 0.1$ centered at $(0.5, 0.2, 0.205)$ aligned in the x_3-direction, in a channel of dimensions $2.5 \times H \times H$, with $H = 0.41$. We have no slip boundary conditions on the cylinder and the channel walls. At the outflow we use a transparant outflow condition, and the inflow condition is given by $u_1^0(0, x_2, x_3) = 16U_m x_2(H - x_2)x_3(H - x_3)/H^4$ and $u_2^0 = u_3^0 = 0$. The kinematic viscosity is $\nu = 10^{-3}$ and $U_m = 0.45$, which gives a Reynolds number $Re = \bar{U}D/\nu = 20$, with $\bar{U} = 4U(0, H/2, H/2)/9$. The goal is to compute the drag coefficient c_D, defined by

$$c_D = \frac{2N(\sigma(u,p))}{\bar{U}^2 DH}. \tag{27}$$

The values of c_D obtained by the different participants reported in Schäfer and Turek, 1996 lie in the (quite wide) interval $[6.08, 8.09]$.

We compute c_D using (26) and an adaptive cG(1)cG(1)-method on tetrahedral meshes in space based on an a posteriori error estimate with dual weights. Starting from the coarse initial mesh in Fig.1 we refine approximately 50% of the elements in each step of the adaptive method. In Fig.3 we show adaptively refined meshes after 4 and 6 refinements, and we plot the computed dual solution in Fig.4.

We alternatively base the mesh refinement (ad hoc) solely on the residuls (no dual weights) and show in Fig.5 the corresponding mesh obtained after 4 refinements.

As a reference value to be used to evaluate the convergence of the two different adaptive methods, we choose the value $c_D = 7.22$ obtained from a computation with 2.013.984 unknowns. In Fig.2 we plot the error vs the number of unknowns for the two adaptive stategiess and note that the one with th dual weights seems to gives the best value.

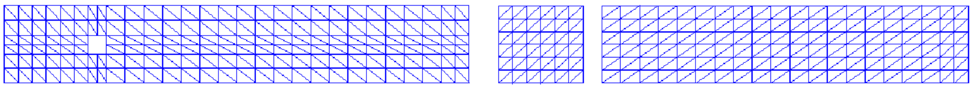

Figure 1. Cross-sections of the meshes in the xy-plane at $z = 0.205$ (left) and in the xz-plane at $y = 0.2$ (right), for the initial mesh.

6.1.2. Unsteady flow We now consider the same problem as in Section 6.1.1, but now with $U_m = 2.25$ which gives an unsteady flow with $Re = 100$. Here the value for the maximum of c_D over three

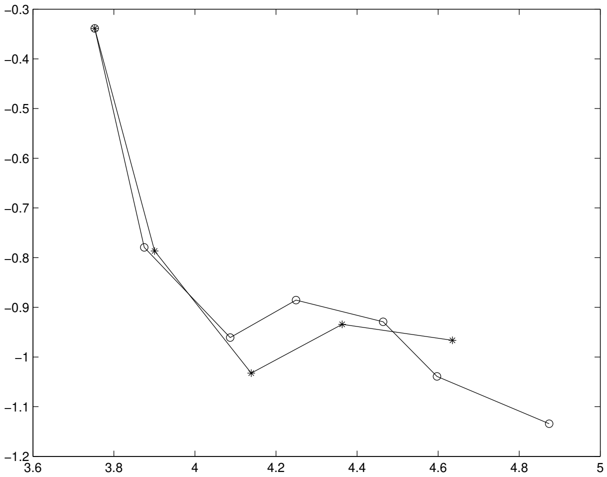

Figure 2. Convergence rates for duality based refinement ('o') and residual based refinement ('*') for $Re = 20$, as a log-log plot of number of unknowns versus relative errors.

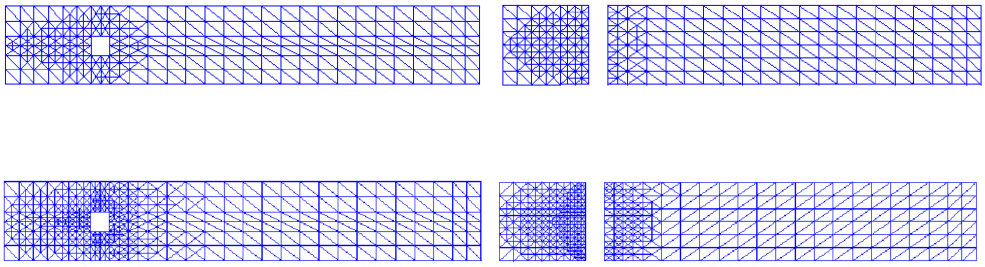

Figure 3. Cross-sections of the meshes in the xy-plane at $z = 0.205$ (left) and in the xz-plane at $y = 0.2$ (right), for computing the drag force with $Re = 20$, after 4 (upper) and 6 (lower) adaptive mesh refinements respectively.

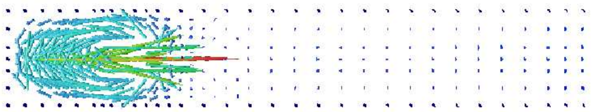

Figure 4. Dual solution after 4 adaptive mesh refinements in the xy-plane at $z = 0.205$.

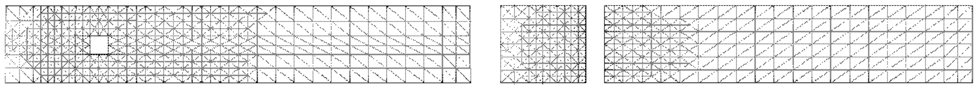

Figure 5. Cross-sections of the meshes in the xy-plane at $z = 0.205$ (left) and in the xz-plane at $y = 0.2$ (right), for computing the drag force after 4 adaptive mesh refinements based on the residuals.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

periods is estimated in Schäfer and Turek, 1996 to be in the interval $[4.3, 4.5]$, and values between $[4.30, 5.57]$ are presented. Based on the on the same reference mesh as in the previous section we set the reference value to $c_D = 4.96$. In Fig.7 we show the adaptively refined meshes after 4 and 6 refinements, and in Fig.6 we plot the convergence rates with respect to the reference value.

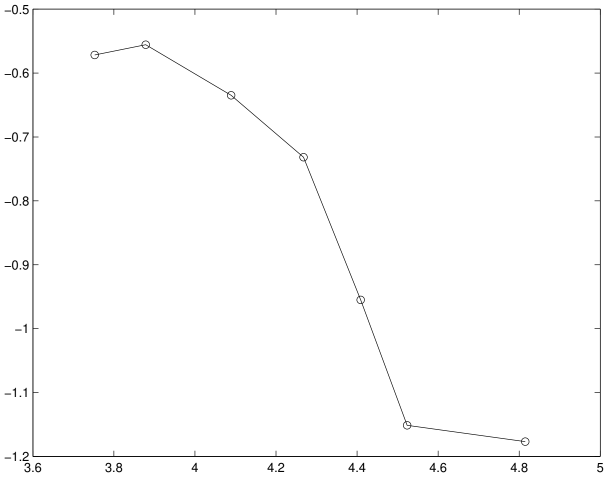

Figure 6. Convergence rates for duality based refinement for $Re = 100$, as a log-log plot of number of unknowns versus relative errors.

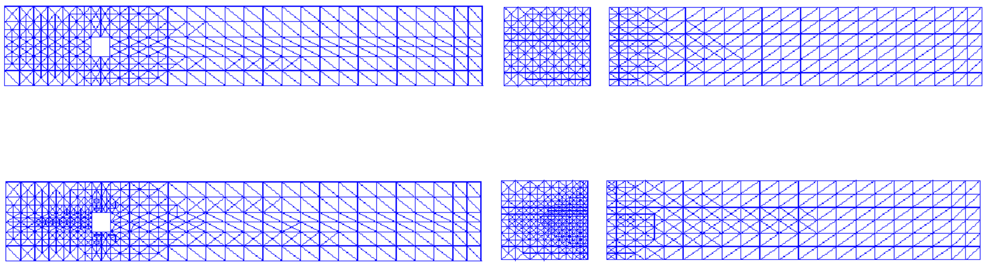

Figure 7. Cross-sections of the meshes in the xy-plane at $z = 0.205$ (left) and in the xz-plane at $y = 0.2$ (right), for computing the drag force with $Re = 100$, after 4 (upper) and 6 (lower) adaptive mesh refinements respectively.

6.2. Computation of a local mean value

To illustrate how the adaptive algorithm works for a different computational goal, we show in Fig.8 the resulting adaptively refined meshes from a computation where the output functional is the local mean value of the first component of the velocity over a cube with side length 0.05 centered at $(1.5, 0.05, 0.3)$, in the case of $U_m = 0.45$.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

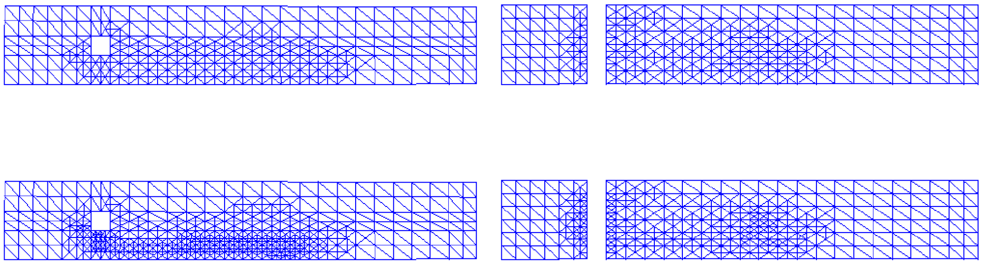

Figure 8. Cross-sections of the meshes in the xy-plane at $z = 0.205$ (left) and in the xz-plane at $y = 0.2$ (right), for computing the average over the square with side length 0.05 centered at $(1.5, 0.05, 0.3)$, after 4 (upper) and 6 (lower) adaptive mesh refinements respectively.

7. Adaptive finite element methods for turbulent flow

To construct adaptive LES finite element methods for turbulent flow computations, we have to take into consideration not only the error from discretization, but also the modeling error from the subgrid model we use. In the case of a DNS the same framework as for the laminar flows applies (no subgrid modeling).

7.1. The averaged Navier-Stokes equations

In a turbulent flow with pointwise unresolvable scales, we may aim at computing instead a *running average* (u^h, p^h) of (u, p), where

$$v^h(x,t) = \frac{1}{h^4} \int_{Q_h} \int_{-h/2}^{h/2} v(x+y, t+s)\, dy\, ds, \tag{28}$$

where h is the local mesh size, $Q_h = \{y \in \mathbb{R}^3 : |y_i| \le h/2\}$, and thus v^h represents a local mean value of a function v. Noting that averaging commutes with space and time differentiation if h is constant, we are led to the following Navier-Stokes like equations for (u^h, p^h) by taking the running average of the equations (6), with suitable constructions near the boundary $\partial\Omega$,

$$\begin{aligned} \dot{u}^h + (u^h \cdot \nabla)u^h - \nu \Delta u^h + \nabla p^h + F_h(u) &= f \qquad && \text{in } \Omega \times I, \\ \operatorname{div} u^h &= 0 && \text{in } \Omega \times I, \\ u^h &= w && \text{on } \partial\Omega \times I, \\ u^h(\cdot, 0) &= u^0 && \text{in } \Omega, \end{aligned} \tag{29}$$

where $F_h(u) = \operatorname{div} \tau^h(u)$, and $\tau^h_{ij}(u) = (u_i u_j)^h - u^h_i u^h_j$ are the components of the *Reynolds stress tensor* $\tau^h(u)$. Alternatively, we may restrict u^h to averaging in space only, or to other averages or *filters*, see Gatski *et al.*, 1996, The basic problem of *Large eddy simulation* LES is now to model $F_h(u)$ in terms of u^h in a subgrid model $\hat{F}_h(u^h)$, or $\tau^h(u)$ in a model $\hat{\tau}_h(u^h)$. LES is assumed to resolve all the scales down to the *inertial range*, which refers to a range of (smallest) scales for which the energy spectrum has a power law behaviour. In the rest of this paper we usually let u^h be a spatial average only, defined by

$$u^h(x,t) = \frac{1}{h^3} \int_{Q_h} u(x+y, t)\, dy. \tag{30}$$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

7.2. Subgrid modeling

The different LES subgrid models proposed take the general form (Ansatz) of a *mixed model*:

$$\hat{\tau}^h_{ij} - \frac{1}{3}\hat{\tau}^h_{kk} = \check{\tau}^h_{ij}(u^h) + \check{\nu}_{ij}\epsilon_{ij}(u^h), \tag{31}$$

with an *algebraic* first part and a *viscous* second part with a *turbulent viscosity* (tensor) $\check{\nu}_{ij}$, also referred to as an *eddy viscosity*. Note that the Ansatz takes trace-free form with the isotropic part being absorbed into the pressure, see e.g. Lesieur, 1997.

The classical eddy viscosity model is the *Smagorinsky model* with $\check{\tau}^h_{ij} = 0$ and $\check{\nu}_{ij} = \nu_T$, where

$$\nu_T = (C_S h)^2 |\epsilon(u^h)|, \tag{32}$$

where C_S is the *Smagorinsky constant*, commonly set to $0.1-0.2$. Eddy viscosity models are in general considered too dissipative and are unable to predict *backscatter*, where subgrid scales feed energy into resolved scales.

Scale similarity models, first introduced by Bardina *et al.*, 1980 and further developed by e.g. Liu *et al.*, 1994, take the form (with $\check{\nu}_{ij} = 0$),

$$\check{\tau}^h_{ij}(u^h) = C_L \tau^H_{ij}(u^h) = C_L((u^h_i u^h_j)^H - (u^h_i)^H (u^h_j)^H), \tag{33}$$

where H represents a coarser scale than h, and C_L is a scale similarity constant. In a scale similarity models the Reynolds stresses on the computational scale h are assumed to be proportional to Reynolds stresses of the resolved field on coarser scales. The scale similarity models can predict backscatter but are considered not to be dissipative enough, and thus often are combined with an eddy viscosity model in a mixed model.

In *dynamic models* first introduced by Germano *et al.*, 1991, the parameters C_S and C_L are determined (locally in space-time) e.g. by comparing resolved Reynolds stresses on different scales.

The *Variational multiscale method* by Hughes *et al.*, 2000 is a modified eddy viscosity model, where the eddy viscosity only on the finest resolved scales. There are further *Fractal models*, see e.g. Scotti and Meneveau, 1995, based on fractal interpolation of the velocity field for a direct evaluation of the Reynolds stresses, and also models based on *homogenization*, see e.g. Frisch, 1995.

In an adaptive LES method including errors from subgrid modeling, we want to adaptively choose the best parameters in a particular subgrid model and also the best of different subgrid models. For this purpose we need a posteriori error estimates separating the subgrid modeling error from the discretization error. We return to this topic below.

7.2.1. Scale similarity of turbulent solutions Turbulent flow show some features of scale similarity, which is expressed in the *Kolmogorov* $5/3$-*law* (se e.g. Frisch, 1995), corresponding to Hölder continuity of the velocities with exponent $1/3$. This gives some hope for scale similarity models, but features of small-scale *coherent structures* also present in turbulent flow, pose challenges. In Section 7.5 we investigate features of scale similarity in a computed turbulent shear flow using a Haar Multi-Resolution-Aanalysis MRA.

To motivate Hölder continuity of the velocities with exponent $1/3$ we may argue as follows: If l is the smallest scale present in the flow and v is the corresponding velocity amplitude, then we should have $vl \sim \nu$ (local Reynolds number ~ 1) and $\nu(v^2/l^2) \sim 1$ (significant turbulent dissipation on the smallest scale), which gives $v^3 \sim l$, that is Hölder continuity with exponent $1/3$ on the smallest scale, and by scale similarity we should have the same exponent also on coarser scales.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

7.3. The G^2-method for turbulent flow

The G^2-method for LES using the Smagorinsky subgrid model takes the form (Hoffman, 2002): Find $(U_h, P_h) \in V^\beta \times Q^\beta$, such that for $n = 1, 2, ..., N$,

$$
\begin{aligned}
&(\dot{U}_h + (U_h \cdot \nabla)U_h, v)_n - (P_h, \operatorname{div} v)_n + (q, \operatorname{div} U_h)_n \\
&+(\tilde{\nu}\epsilon(U_h), \epsilon(v))_n + (\delta_1 a(U_h; U_h, P_h), a(U_h; v, q))_n \\
&-(\check{\tau}^h(U_h), \nabla v)_n + (\delta_2 \operatorname{div} U_h, \operatorname{div} v)_n + ([U_h^{n-1}], v_+^{n-1}) \\
&=(f, v + \delta_1 a(U_h; v, q))_n \quad \forall (v, q) \in V_n^\beta \times Q_n^\beta,
\end{aligned}
\tag{34}
$$

where $a(w; v, q) = D_{w,t} v + \nabla q - \nu \Delta v$ with the Laplacian defined elementwise, $\delta_1 = \frac{1}{2}(k_n^{-2} + |U_h|^2 h_n^{-2})^{-1/2}$ in the convection-dominated case $\tilde{\nu} < U_h h_n$ and $\delta_1 = \kappa_1 h^2$ otherwise, $\delta_2 = \kappa_2 h$ if $\tilde{\nu} < U_h h_n$ and $\delta_2 = \kappa_2 h^2$ otherwise, with κ_1 and κ_2 positive constants of unit size, $\tilde{\nu} = \max(\nu + \nu_T, \kappa_3 |R(U_h, P_h)| h^2)$, where ν_T is a turbulent eddy viscosity and $\check{\tau}^h$ is an algebraic part in the mixed model (31), and $R(U_h, P_h) = \sum_{i=1}^4 R_i(U_h, P_h)$ with

$$
\begin{aligned}
R_1(U_h, P_h) &= |\dot{U}_h + U_h \cdot \nabla U_h + \nabla P_h - f + \operatorname{div} \check{\tau}^h(U_h) - \tilde{\nu}\Delta U_h|, \\
R_2(U_h, P_h) &= \tilde{\nu} D_2(U_h), \\
R_3(U_h, P_h) &= |[U_h^{n-1}]|/k_n \quad \text{on } S_n, \\
R_4(U_h, P_h) &= |\operatorname{div} U_h|,
\end{aligned}
$$

with $D_2(U_h)$ defined by (14), and where $R_1(U_h, P_h)$ is defined elementwise and with piecewise linears in space, the Laplacian ΔU_h is zero. In the computations presented below, we choose $\kappa_3 = 0$ corresponding to shutting off the artificial viscosity.

7.3.1. The cG(1)cG(1)-method for turbulent flow The corresponding cG(1)cG(1)-method reads: For $n = 1, ..., N$, find $(U_h^n, P_h^n) \in V_n^0 \times Q_n^0$, such that

$$
\begin{aligned}
&(\frac{U_h^n - U_h^{n-1}}{k_n}, v) + (\hat{U}_h^n \cdot \nabla \hat{U}_h^n + \nabla P_h^n, v + \delta_1(\hat{U}_h^n \cdot \nabla v + \nabla q)) \\
&+ \delta_2(\operatorname{div} \hat{U}_h^n, \operatorname{div} v) + (\nabla \cdot \hat{U}_h^n, q) + (\tilde{\nu}\nabla\hat{U}_h^n, \nabla v) - (\check{\tau}_n^h(\hat{U}_h^n), \nabla v) \\
&= (f^n, v + \delta_1(\hat{U}_h^n \cdot \nabla v + \nabla q)) \quad \forall (v, q) \in V_n^0 \times Q_n^0,
\end{aligned}
\tag{35}
$$

where $\hat{U}_h^n = \frac{1}{2}(U_h^n + U_h^{n-1})$.

7.4. A posteriori error estimation for turbulent flow

To derive an a posteriori error estimate for $u^h - U_h$, where u^h is the running average of the exact velocity u satisfying (29), we have to take into account both the error from discretization and that from subgrid modeling. Aiming at error control of the quantity $\int_Q (u^h - U_h) \cdot \psi \; dx \; dt$ in $Q = \Omega \times I$, with $\psi \in L_2(I; [L_2(\Omega)]^3)$ given, we introduce the following linearized dual problem: Find $(\varphi, \theta) \in L_2(I; [H_0^1(\Omega)]^3 \times L_2(\Omega)) \equiv W$ such that

$$
\begin{aligned}
-\dot{\varphi} - (u^h \cdot \nabla)\varphi + \nabla U_h \cdot \varphi + \nabla\theta - \tilde{\nu}\Delta\varphi &= \psi && \text{in } Q \\
\operatorname{div} \varphi &= 0 && \text{in } Q \\
\varphi &= 0 && \text{on } \Gamma \times I, \\
\varphi(\cdot, T) &= 0 && \text{in } \Omega,
\end{aligned}
\tag{36}
$$

where $(\nabla U_h \cdot \varphi)_j = (U_h)_{,j} \cdot \varphi$, and we note that we use the viscosity $\tilde{\nu}$, including the eddy viscosity ν_T. Depending of the choice of ψ, the quantity $\int_Q (u^h - U_h) \cdot \psi \, dx \, dt$ may represent different norms of the error, or local or global mean values such as the mean drag or lift force. Following the general methodology presented above, we arrive at an error representation formula using integrations by parts from which we can derive various a posteriori error estimates.We state the results in the following two theorems from Hoffman, 2002:

Theorem 1. *With u^h the solution to (29) and $\psi \in L_2(I; [L_2(\Omega)]^3)$ given, we have the following error representation formula for $(U_h, P_h) \in V^\beta \times Q^\beta$:*

$$\begin{aligned}
\int_Q (u^h - U_h) \cdot \psi \, dx \, dt = & \int_Q R_1(U_h, P_h) \cdot \varphi \, dx \, dt + \int_Q R_4(U_h, P_h) \cdot \theta \, dx \, dt \\
& + \frac{1}{2} \sum_{n=1}^{N} \sum_{K \in \mathcal{T}_n} \int_{\partial K \times I_n} R_2(U_h, P_h) \cdot \varphi \, ds \, dt \\
& + \sum_{n=1}^{N} \int_\Omega R_3(U_h, P_h) \cdot \varphi(t_{n-1}) \, dx \\
& + \int_Q R_M(u, U_h) \cdot \varphi \, dx \, dt
\end{aligned}$$

where (φ, θ) are the solutions to the dual problem (36), and

$$\begin{aligned}
R_1(U_h, P_h) &= f - (\dot{U}_h + U_h \cdot \nabla U_h + \nabla P_h - \tilde{\nu} \Delta U_h + \operatorname{div} \check{\tau}^h(U_h)), \\
R_2(U_h, P_h) &= \tilde{\nu} \left[\frac{\partial U_h}{\partial n} \right] \\
R_3(U_h, P_h) &= [U_h^{n-1}] \quad \textit{on } S_n, \\
R_4(U_h, P_h) &= \operatorname{div} U_h, \\
R_M(u, U_h) &= \operatorname{div} \check{\tau}^h(U_h) - (\operatorname{div} \tau^h(u) + \nu_T \Delta u^h),
\end{aligned}$$

with $(\check{\tau}^h, \nu_T)$ from (34).

We note that Theorem 1 is valid for any $(U_h, P_h) \in V^\beta \times Q^\beta$. If (U_h, P_h) are computed using a Galerkin method we may use the Galerkin orthogonality property for the discretization error to subtract interpolants of the dual solution (φ, θ), and then estimate the interpolation errors in terms of derivatives of (φ, θ) and powers of the space and time discretization parameters. As an example, we present the corresponding error estimates for the cG(1)cG(1)-method, and for simplicity we consider the case when $\delta_1 = \delta_2 = 0$.

Theorem 2. *If u^h solves (29), $(U_h, P_h) \in V_n^0 \times Q_n^0$ solves (35) with $\delta_1 = \delta_2 = 0$, (φ, θ) solves (36), and $\psi \in L_2(I; [L_2(\Omega)]^3)$ is given, then*

$$\begin{aligned}
& | \int_Q (u^h - U_h) \cdot \psi \, dx \, dt | \leq \sum_{n=1}^{N} \int_{I_n} \sum_{K \in \mathcal{T}_n} \{ \int_K \{ \, |R_M(u, U_h)| \cdot |\varphi| \\
& + |R_1(U_h)| \cdot (C_h h_{n,K}^m \|D^m \varphi\|_{\infty,K,n} + C_k k_n \|\dot{\varphi}\|_{\infty,K,n}) \\
& + |R_4(U_h)| \, (C_h h_{n,K}^m \|D^m \theta\|_{\infty,K,n} + C_k k_n \|\dot{\theta}\|_{\infty,K,n}) \, \} \, dx \\
& + \int_{\partial K} |R_2(U_h)| \cdot (C_h h_{n,K}^m \|D^m \varphi\|_{\infty,\partial K,n} + C_k k_n \|\dot{\varphi}\|_{\infty,\partial K,n}) \, ds \, \} \, dt,
\end{aligned}$$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

for $m = 1, 2$, *where* $\|\cdot\|_{\infty,K,n} \equiv \max_{(x,t)\in K\times I_n} |\cdot|$, $|w| \equiv (|w_1|, ..., |w_n|)$ *for* $w \in \mathbb{R}^n$, $h_{n,K} = \max_{t\in I_n} h_K(t)$ *with* $h_K(t)$ *the diameter of element* K *at t*, D^m *measures derivatives with respect to* x *of order* m, *and* C_h, C_k *represents interpolation constants.*

Remark 3. *We note that there are several possibilities to pose the dual problem. In (36) we chose to include the turbulent viscosity* ν_T *in the dual problem, but we could alternatively have chosen to only use the viscosity* ν, *which would have given a different modeling residual* $R_M(u, U_h)$. *The motivation for using the larger turbulent viscosity is improved regularity of the dual solution* (φ, θ).

Remark 4. *If* $\delta_1, \delta_2 \neq 0$, *we may view the stabilizing terms as a modification of the continuous equation, which we solve by a standard Galerkin method, according to Section 1.4. We then modify the dual problem accordingly, to obtain Galerkin orthogonality for the discretization error of the solution of the stabilized equations.*

7.5. Estimation of the modeling residual

The subgrid residual on the scale h cannot be directly evaluated, because it involves the unknown solution u, or rather the components of the exact Reynold's stresses $\tau_{ij}^h(u)$, and we propose to instead estimate these components by scale similarity from evaluating Reynold's stress components on coarser scales $H > h$ based on computed velocities on the scale h. We start seeking this way to evaluate the subgrid modeling residual in the case we are not using any subgrid model, that is we simply seek to estimate the components $\tau_{ij}^h(u)$. Using the scale similarity model (33), or the ideas in Hoffman *et al.*, 2000; Hoffman, 2000; Hoffman, 2001; Hoffman, 2002, we may seek to extrapolate $\tau_{ij}^h(u)$ from coarser scales $2h$ and $4h$ as follows:

$$\tilde{\tau}_{ij}^h(U_h) = g(\tau_{ij}^h(U_h), \tau_{ij}^{2h}(U_h), \tau_{ij}^{4h}(U_h)) \tag{37}$$

with $g(a, b, c)$ from Hoffman and Johnson, 2002a, defined by

$$g(a, b, c) = (1 - (\frac{c - b^{4h}}{b^{4h} - a^{4h}})^{\log_2(h_f/h)}) \frac{b^{4h} - a^{4h}}{\dfrac{c - b^{4h}}{b^{4h} - a^{4h}} - 1} \tag{38}$$

and h_f is the finest scale in the exact solution. This extrapolation formula is based on an Ansatz of $\tau_{ij}^h(u)$ of the form

$$E_h(v, w)(x) = C(x)h^{\mu(x)} \tag{39}$$

for covariances of the form $E_h(v, w) = (vw)^h - v^h w^h$, and a fundamental question is now if the Ansatz (39) is valid for $\tau_{ij}^h(u)$ in the computations, that is if we have scale similarity. We test this hypothesis for the computed solution U_h, where we compute $a_{ij}^1 = \tau_{ij}^{2h}(U_h) - \tau_{ij}^h(U_h)$, $a_{ij}^2 = \tau_{ij}^{4h}(U_h) - \tau_{ij}^{2h}(U_h)$, and $a_{ij}^3 = \tau_{ij}^{8h}(U_h) - \tau_{ij}^{4h}(U_h)$. As an approximation of the running average operator on the scale h we use a projection $[\cdot]^h$ onto the space of piecewise constant functions on the mesh corresponding to h. The spaces of piecewise constant functions on successively uniformly refined meshes form a Haar Multi Resolution Analysis MRA in $L_2(\Omega)$, where the uniform refinement dividing one tetrahedron into eight new ones is described in Figure 9.

In Hoffman *et al.*, 2000; Hoffman, 2000, covariances with respect to a Haar MRA is investigated, and it is shown that $[u_i u_j]^h - [u_i]^h [u_j]^h = \sum_{k\leq h}$ {Haar coeff. of u_i × Haar coeff. of u_j, where we sum over scales on scales k finer than or equal to h, and $a_{ji}^1, a_{ji}^2, a_{ji}^3$ now represents the sum of Haar

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

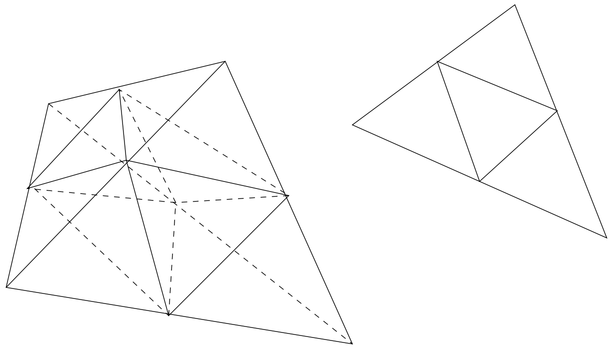

Figure 9. Uniformly refined tetrahedral and triangular elements.

coefficients of $\tau^{8h}(U_h)$ on the scales $2h$, $4h$, and $8h$ respectively. If Ansatz (39) is valid ($\tau^{8h}(U_h)$ is scale similar and thus possible to extrapolate) a^1_{ji}, a^2_{ji}, and a^3_{ji} should decrease regularly. If we have scale similarity in the coarser scales $2h$, $4h$, and $8h$, we anticipate scale similarity in finer scales, since we assume that we are in the inertial range. In Figure 10 we present the L_1-norms of a^1_{ji}, a^2_{ji}, and a^3_{ji}, from Hoffman, 2002, showing that we have a certain degree of scale similarity, and we find that the decrease is typically by a factor 1.5. From Hoffman *et al.*, 2000; Hoffman, 2000, we have a dependence $a^k_{ij} \sim (2^k h)^{\delta_i+\delta_j}$, for δ_i, δ_j being the Hölder exponents of u_i and u_j respectively. If we assume that $\delta_i = \delta_j = \delta$, we get that $a^{k+1}_{ij}/a^k_{ij} = (2^{k+1}h)^{2\delta}/(2^k h)^{2\delta} = 2^{2\delta} = 1.5$, which gives that $\delta = \log(1.5)/2\log(2) \approx 0.29$, which is very close to $\delta = 1/3$, corresponding to the velocity being Hölder continuous with exponent 1/3, which is consistent with the *Kolmogorov 5/3-law* for the energy spectrum, see Frisch, 1995. The results in Figure 10 support the Ansatz (39), although we note that these are global results (using the global L_1-norm).

If we use a subgrid model in the computations we need to estimate the difference $F_h(u) - \hat{F}_h(U_h)$, and we are thus lead to model terms of the form $\Delta_h = \tau^h_{ij}(u) - \hat{\tau}^h_{ij}(U_h)$. We may base our estimation of Δ_h on extrapolation, and we then have to find approximations to $\Delta_{2h} = \tau^{2h}_{ij}(u) - \hat{\tau}^{2h}_{ij}(U_{2h})$ and $\Delta_{4h} = \tau^{4h}_{ij}(u) - \hat{\tau}^{4h}_{ij}(U_{4h})$. Using the Ansatz (39), may use the approximation $\tau^{2h}_{ij}(u) \approx \tilde{\tau}^{2h}_{ij}(U_h)$, where $\tilde{\tau}^{2h}_{ij}(U_h) = \tilde{g}_{2h}(\tau^h_{ij}(U_h), \tau^{2h}_{ij}(U_h), \tau^{4h}_{ij}(U_h))$ with $\tilde{g}_{2h}(a,b,c)$ defined by

$$\tilde{g}_{2h}(a,b,c) = (1 - (\frac{c-b^{4h}}{b^{4h}-a^{4h}})^{1+\log_2(h_f/h)})\frac{c-b^{4h}}{\frac{c-b^{4h}}{b^{4h}-a^{4h}} - 1} \tag{40}$$

and in a similar way we get $\tau^{4h}_{ij}(u) \approx \tilde{\tau}^{4h}_{ij}(U_h)$, where $\tilde{\tau}^{4h}_{ij}(U_h) = \tilde{g}_{4h}(\tau^h_{ij}(U_h), \tau^{2h}_{ij}(U_h), \tau^{4h}_{ij}(U_h))$ with $\tilde{g}_{4h}(a,b,c)$ defined by

$$\tilde{g}_{4h}(a,b,c) = (1 - (\frac{c-b^{4h}}{b^{4h}-a^{4h}})^{2+\log_2(h_f/h)})\frac{c-b^{4h}}{1 - \frac{b^{4h}-a^{4h}}{c-b^{4h}}}. \tag{41}$$

We then get $\Delta_h \approx \tilde{\Delta}_h = g(0, \tilde{\Delta}_{2h}, \tilde{\Delta}_{4h})$, with $\tilde{\Delta}_{2h} = \tilde{\tau}^{2h}_{ij}(U_h) - \hat{\tau}^{2h}_{ij}(U_{2h})$, $\tilde{\Delta}_{4h} = \tilde{\tau}^{4h}_{ij}(U_h) - \hat{\tau}^{4h}_{ij}(U_{4h})$, and $g(a,b,c)$ defined by (38).

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

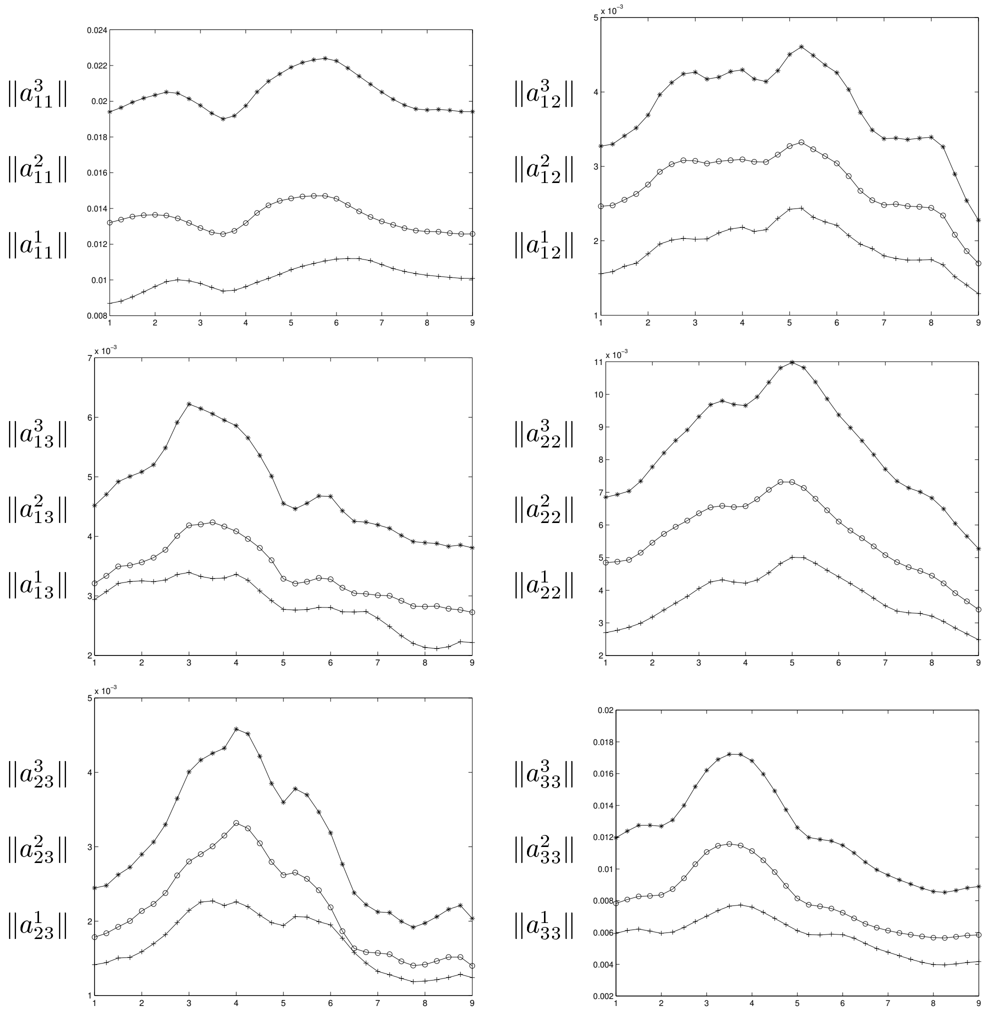

Figure 10. $\|a^1_{ij}\|_1$ ('$+$'), $\|a^2_{ij}\|_1$ ('o') and $\|a^3_{ij}\|_1$ ('$*$')

7.6. *Discretization error vs. modeling error*

We now proceed to estimate in a concrete flow the errors from both discretization and modeling using the a posteriori error estimate from Theorem 2. We consider a turbulent flow obtained after transition to turbulence in plane Couette flow using the cG(1)cG(1)-method from Section 7.3.1 without subgrid model. We compute on the unit cube using a regular tetrahedral mesh with $65 \times 65 \times 65$ nodes and we use periodic boundary conditions in the streamwise x_1-direction and in the spanwise x_3-direction, and with the streamwise velocity equal to ± 1 on top and bottom, and we set $\nu = 1/10000$ and thus the Reynolds number is 10000. We compute on the time interval $[20, 30]$ where the flow has become turbulent after transition, using a computed velocity at $t = 20$ as exact initial condition. We may view the computation as a DNS at an effective Reynolds number in the range 100-1000.We thus consider a "real" turbulent flow, and not a flow with an artificially constructed "turbulent" inlet velocity. For

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

more details of the computation and a study of the transition process from laminar to turbulent flow in Couette flow, see Hoffman and Johnson, 2002b. In Fig.11 we plot the velocity isosurfaces for $|u| = 0.2$, after transition to turbulence, and of course note that the flow appears highly irregular.

Recalling Theorem 2, we have that

$$|\int_Q (u^h - U_h) \cdot \psi \, dx \, dt| \leq e_D + e_M,$$

where

$$\begin{aligned} e_D = \sum_n \int_{I_n} \sum_{K \in \mathcal{T}_n} \{ \int_K \{ \, & |R_1(U_h)| \cdot (C_h h_{n,K}^m \|D^m \varphi\|_{\infty,K,n} + C_k k_n \|\dot{\varphi}\|_{\infty,K,n}) \\ & + |R_4(U_h)| \, (C_h h_{n,K}^m \|D^m \theta\|_{\infty,K,n} + C_k k_n \|\dot{\theta}\|_{\infty,K,n}) \, \} \, dx \\ & + \int_{\partial K} |R_2(U_h)| \cdot (C_h h_{n,K}^m \|D^m \varphi\|_{\infty,\partial K,n} + C_k k_n \|\dot{\varphi}\|_{\infty,\partial K,n}) \, ds \, \} \, dt, \end{aligned}$$

represents the error from discretization and

$$e_M = \sum_n \int_{I_n} \sum_{K \in \mathcal{T}_n} \int_K |R_M(u, U_h)| \cdot |\varphi| \, dx \, dt,$$

represents the error from modeling. Note that since $h = 1/64$ and $\nu = 1/10000$, we expect the exact flow to contain scales finer scales than h, and thus we expect an error from modeling. We now seek to evaluate e_D and e_M using Theorem 2 with computed dual solutions (φ, θ) as input. The discretization residuals are directly computable from the approximate solutions (U_h, P_h), whereas the modeling residual $R_M(u, U_h)$ has to be estimated. Here we use (33), with $C_L = 1$, to estimate the modeling residual without a subgrid model. In the estimate of the discretization error we use $C_h = 1/8$ and $C_k = 1/2$, which are approximations of the interpolation constants motivated by a simple analysis on a reference element.

If the modeling error without a subgrid model is neglible compared to the discretization error, then we do not need a subgrid model. If on the other hand the modeling error dominates, we need to either use a subgrid model or to refine the computational mesh.

In Fig.12 we present estimates of the relative discretization error and modeling error, normalized by $U_T = \|U_h(30)\|_1 \approx 0.43$ ($\|\cdot\|_1 = \int_\Omega |\cdot| \, dx$), in the computation of a space-time average over $\omega \times [30 - d(\omega), 30]$ of the solution u^h, with ω being a spatial cube with side length $d(\omega)$, centered at $(0.5, 0.5, 0.5)$. This corresponds to $\psi = \chi_{\omega \times [30-d(\omega),30]} / |\chi_{\omega \times [30-d(\omega),30]}|$ in the dual problem (36), where χ_D is the characteristic function for D, and $|D|$ denotes the space-time volume of D. We display the estimated errors as functions of T, where we compute on the time interval $[T, 30]$ assuming the initial velocity at each initial time T (taken from the simulation on $[20, 30]$), is exact. .

We find that the estimates of the discretization error and the modeling error are of the same order in this computation, both errors are less than a few percent of the size of the solution, and the errors of course increase if we compute over a longer time. In the estimate of the discretization error we have neglected the residual $R_2(U_h)$, since the other residuals dominate for ν small when we do not use an eddy viscosity subgrid model. We note that both e_D and e_M are larger for smaller space-time averages, supporting our belief that it is harder to compute smaller space-time averages than larger. We also note that the difference between e_D and e_M is smaller for larger $d(\omega)$. Altogether, we estimate the error from modeling to be significant as compared to the error from discretization, and thus that subgrid modeling could be motivated.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

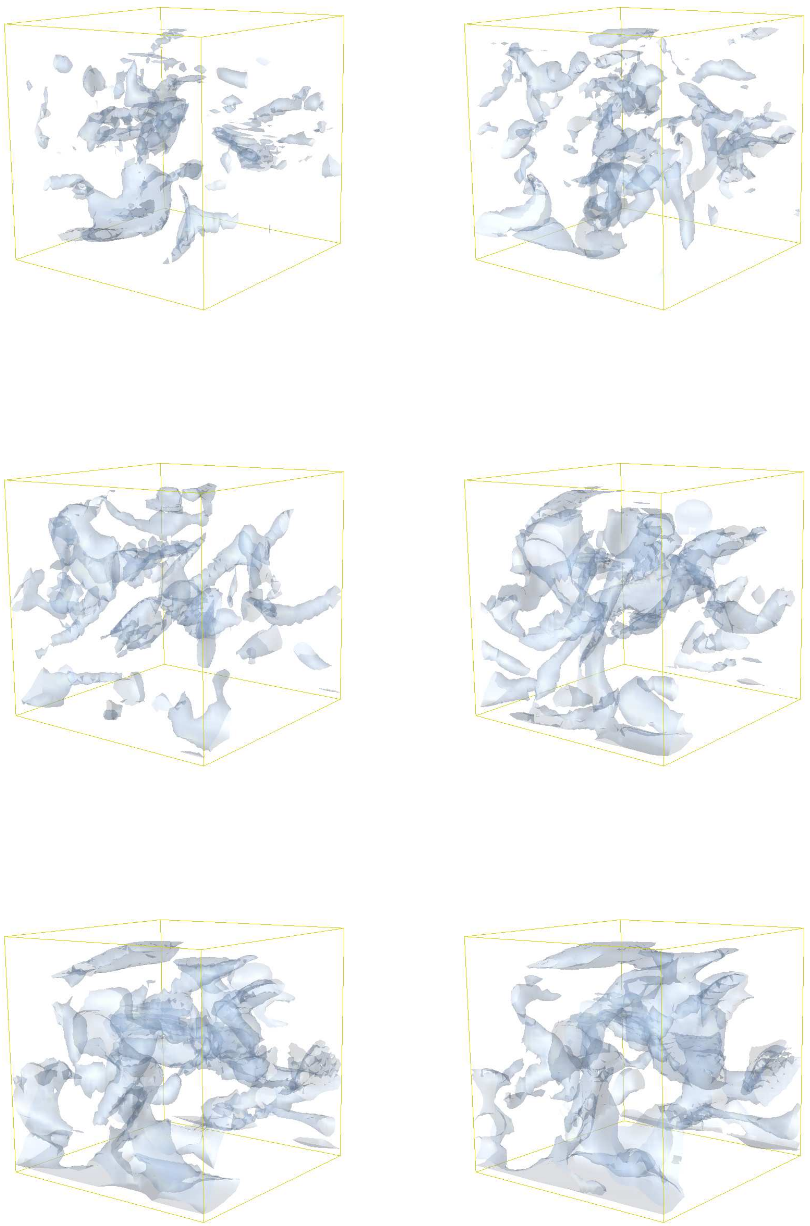

Figure 11. Velocity isosurfaces for $|u| = 0.2$ in Couette flow for $t = 20, 22, 24, 26, 28, 30$

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

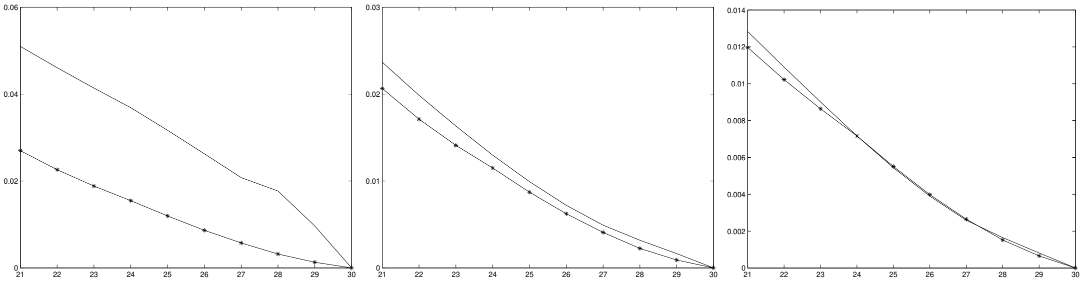

Figure 12. e_D/U_T ('-') and e_M/U_T('*') for $d(\omega) = 0.125$ (left), 0.25 (middle), 0.5 (right), as functions of time

In this test, we use the computed velocity with $h = 1/64$ as exact DNS and we estimated the modeling error on the coarser scale $h = 1/32$. The unresolved scale is thus just one multiple of 2, or one scale, and the modeling error accordingly small. With the DNS instead on say $h = 1/512$, we would have four scales to model and the modeling error would increase. Thus: the larger the Reynolds number is the stronger is the need for subgrid modeling (and the better the chances of achieving improvements from subgrid modeling).

In Fig.13 we present plots of the discretization residuals $R_1(U_h, P_h)$, $R_4(U_h)$ and the modeling residual $R_M(u, U_h)$. We see that first $R_1(U_h, P_h)$ is large in the middle of the domain, but after some time $R_1(U_h, P_h)$ is largest at the top and bottom. This is because the flow is changing from a Couette profile (linear profile in the vertical direction of the streamwise velocity), where the residual is large in the middle, into a solution with small velocities in the middle and sharp boundary layers at top and bottom that the mesh is not capable of resolving, causing large residuals in these layers. The modeling residual $R_M(u, U_h)$ behaves similarily, whereas $R_4(U_h)$, on the other hand, is more isotropic.

Remark 5. *In the computation of the dual problem we use a cG(1)cG(1)-method, corresponding to the method used for the primal problem, on a uniform tetrahedral mesh with* $32 \times 32 \times 32$ *nodes, and we approximate both* u^h *and* U_h *with* U_h*, projected onto this mesh.*

Remark 6. *Since we use a stabilized Galerkin method there are also terms from the stabilization present in* e_D *and* e_M*. In this study we assume these terms to be small compared to the other terms since they are weighted by a small stabilization parameter.*

7.7. *Evaluation of different subgrid models*

In Section 7.6 we estimated the errors from discretization and modeling in a case without subgrid model. We now consider the problem of estimating the error from modeling for different subgrid models, with the goal of being able to adaptively choosing the best from a set of available subgrid models. We seek u^h, with $h = 1/32$, and we use $\tau^h(U_{h/2})$ as an approximation of the true Reynolds stresses $\tau^h(u)$. We compare the scale similarity model (33) and the Smagorinsky model (32).

Experience tells us, see e.g. Gatski *et al.*, 1996, that neither an eddy viscosity model nor a scale similarity model may work as a stand alone subgrid model. Instead, a combination of the two in a mixed model may be superior. A possible explanation is that $\tau^h(u)$ is combined of a low frequency part and a high frequency part, and that an eddy viscosity model has typically a better chance to model the high frequency part whereas a scale similarity model typically has a better chance to model the low frequency part. As a test we try to fit the the scale similarity model (33) and the eddy viscosity model

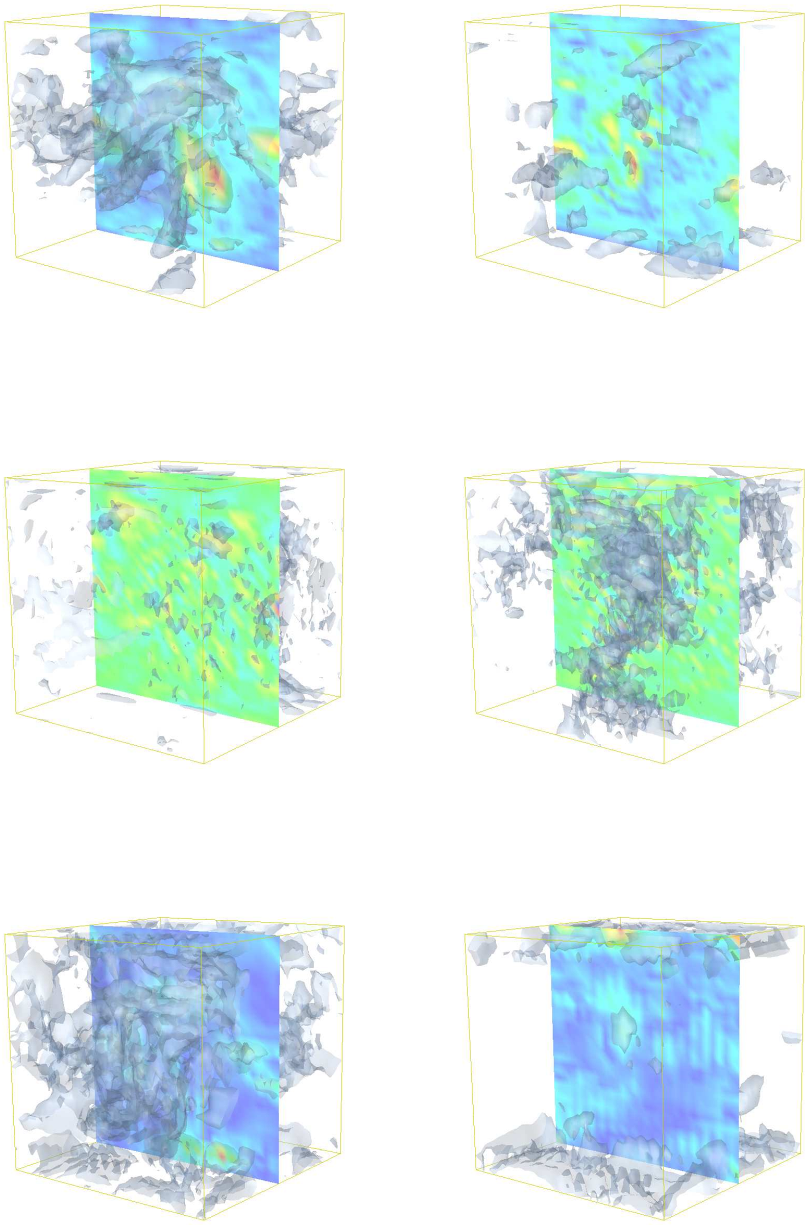

Figure 13. High value isosurf.: $|R_1(U_h, P_h)|$ (upper), $|R_4(U_h)|$ (middle), $|R_M(u, U_h)|$ (lower), $t = 20, 30$

(32) to our approximation of the true Reynolds stresses $\tau^h(u) \approx \tau^h(U_{h/2})$, by changing the constants C_S and C_L, where we base the models on the solution $U_{h/2}$, projected onto the scale h.

We find that we are unable to fit the Smagorinsky model by changing C_S. Using the scale similarity model we are able to reduce the modeling error by 20%, for $C_L = 0.25$. Admittedly, our test is limited, and more tests are need to draw any conclusions.

Remark 7. *We note that this test does not rule out the Smagorinsky model as a possible subgrid model, since it is known Gatski et al., 1996, that eddy viscosity models does poor in these type of tests, possibly because of the form of the Reynolds stresses being composed of a low frequency part as well as a high frequency part. We may split the Reynolds stresses as* $\tau^h(u) = (\tau^h(u))^H + (\tau^h(u) - (\tau^h(u))^H), \quad H > h$*, where a scale similarity model might be a good model for the low frequency part* $(\tau^h(u))^H$*, and an eddy viscosity model would be a good model for the high frequency part* $\tau^h(u) - (\tau^h(u))^H$.

Remark 8. *The role of the Smagorinsky viscosity, which is kind of* artificial vicosity *is to ensure that the discrete model contains sufficient turbulent dissipation on the finest scales of the computation, or rather the right amount of dissipation. Now, the Galerkin discretization process itself, apart from the modeling, typically also introduces artificial viscosity in the form of stabilizing least-squares terms or shock-capturing residual-dependent viscosity, and thus artifical viscosity is introduced both from discretization and Smagorinsky subgrid modeling, and it may be difficult to tell these contributions apart. It may be that the proper amount of artificial viscosity in the discretization would ensure just the right amount of turbulent dissipation, and then Smagorinsky would not lead to improvement, as our test indicates. Of course the key question is to decide if the discretization gives the right amount of dissipation; we hope to be able to do so through a posteriori error estimation as indicated.*

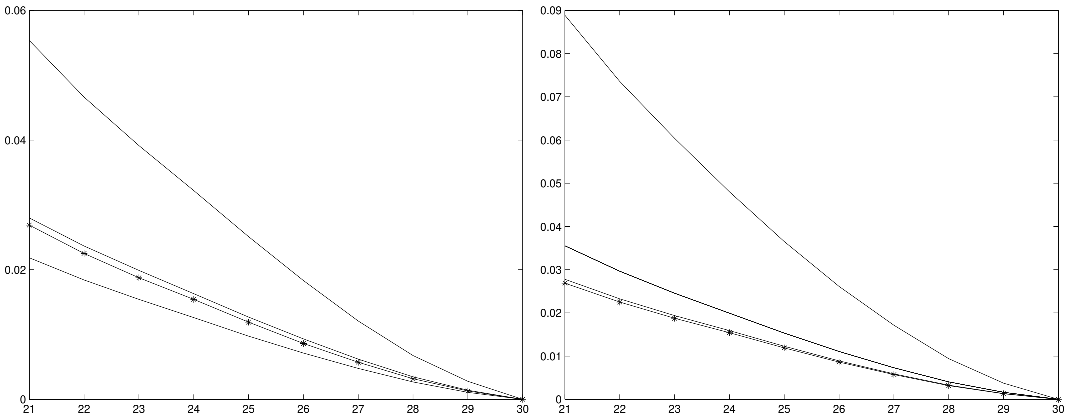

Figure 14. e_M/U_T for $C_L = 0.25, 0.5, 1.0$ ('-') and $C_L = 0$ ('*') (left), and $C_S = 0.05, 0.1, 0.2$ ('-') and $C_S = 0$ ('*') (right), for $d(\omega) = 0.125$, as functions of time

7.8. Adaptive strategies for turbulent flow computations

Based on the a posteriori error estimation presented, we may device adaptive methods for both discretization and modeling. For example, we may use adaptive mesh refinement only without a subgrid model (except from the artifical viscosity of the Galerkin method), where we use a subgrid model only to estimate the modeling residual. We may also adaptively choose the best of available subgrid models (different Ansatz or parameter values).

8. Computability and predictability

We now return to the basic problem of computability/predictability of fluid flow From the error representation of Theorem 1 we know that the computational error of an output quantity may be expressed as a space-time integrals of residuals times (derivatives of) the solution to an associated linearized dual problem. The residuals measure how well the computed solution satisfies the Navier-Stokes equations, and the solution of the dual problem determines how the residual influences the particular output considered. We may alternatively view the dual problem as describing how the error, produced through a non zero residual, is propagated in space-time to the output quantity. The size of (the derivatives of) the dual solution directly couples to computability: the larger these quantities are the higher is the computational cost.

The linearized dual Navier-Stokes equations are closely related to the linearized Navier-Stokes equations, where the linearized dual Navier-Stokes equations describe the propagation of errors coupling to the question of computability, and the linearized Navier-Stokes equations describe the propagation of physical perturbations coupling to predictability and hydrodynamic stability.

8.1. Computability of the Navier-Stokes equations

In this section we investigate computability of different output quantities for a set of test problems. As indicated, dual solutions carries information on the growth and propagation of perturbations from discretization and modeling, and in particular underlie the mesh selection in adaptive methods. We may evaluate the dual solution in terms of stability factors, or we may study the dual solution in more detail. The purpose of computing stability factors is to get a rough measure of relevant stability features. For more precise error estimation, the form of the a posteriori error estimates with (more or less local) stability weights, is advantageous. The (derivatives of the) dual solution may be vastly different depending on the data of the dual problem connecting to output, with smooth data corresponding to large mean values, and of course the underlying flow. We present a selection of stability factors from Hoffman and Johnson, 2002a, using for the dual solution the $L_1(0,T;L_1(\Omega))$-norm in space-time, denoted by $\|\cdot\|_I$ with $I=[0,T]$, and correspondingly using $L_\infty(0,T;L_\infty(\Omega))$ for the residuals, see Tab. I. The different stability factors measure the dual solution and first order derivatives in space-time thereof. A variety of alternative combinations of norms in space/time for the residuals and the dual solutions are possible. We now present results from Hoffman, 2002, comparing dual solutions and stability factors for laminar and turbulent flows. We generally compute on regular tetrahedral meshes with mesh size $h = 1/32 - 1/64$ using the cG(1)cG(1)-method, and the viscosity ν varies from 10^{-3} to 10^{-4}. The stabilization introduces a numerical vicosity, which may be of size $h^{3/2}$ at best, indicating that we compute with effective viscosities in the range 10^{-2} to 10^{-3}. We linearized the dual problems at computed approximations of the primal solutions averaged over a regular tetrahedral mesh of size $h = 1/16$, which in the case of turbulent solutions might lead to an under-estimation of the stability factors. We use the same computational meshes for the dual problem as for the primal problem.

In general, the stability factors increase with (i) the number of derivates of the dual solution, (ii) mean values of decreasing diameter, and (iii) the complexity/stability of the underlying flow. In the a posteriori error estimate, stability factors of derivatives are accompanied with corresponding powers of

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

Table I. Stability factors, where $\|\cdot\|_I$ corresponds to the $L_1(I; L_1(\Omega))$-norm

$S_{0,1}$	$S_{0,2}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
$\|\varphi\|_I$	$\|\theta\|_I$	$\|\nabla\varphi\|_I$	$\|\dot{\varphi}\|_I$	$\|\nabla\theta\|_I$	$\|\dot{\theta}\|_I$

Table II. Bluff body: stability factors

$d(\omega)$	$S_{0,1}$	$S_{0,2}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
1/16	6.0	3.4	121.1	33.1	16.3	57.7
1/8	4.9	3.1	78.6	19.2	12.1	35.8
1/4	3.3	2.6	46.2	11.2	7.8	18.1
1/2	1.7	1.9	20.0	5.2	3.7	8.7

the mesh size in space/time, and there is a trade-off: e.g. with a derivative in time present there is an extra factor of the time step, but the stability factor without the derivative is smaller.

8.1.1. Bluff body We consider channel flow with no slip walls and 1×1 quadratic cross-section of length 4 containing a cubic body of side length 0.25 with center at $(0.5, 0.5, 0.5)$, where $x = 0$ is the inflow boundary. We impose a parabolic inflow condition $u = (16y(1-y)z(1-z), 0, 0)$, a transparant outflow condition, and we set $\nu = 10^{-3}$. We compute on a regular tetrahedral mesh, with $h = 1/32$, using the cG(1)cG(1)-method. We start from $u = 0$ at time $t = 0$, and we compute to time $T = 20$. We consider the problem of computing the space-time average of u_1 over the domain $\omega \times [T - d(\omega), T]$, where $\omega \subset \Omega$ is a cube centered at $(2.5, 0.5, 0.5)$ with side length $d(\omega)$. To estimate this error the appropriate data to the linearized dual problem is zero final data and a force $\psi = (\chi_\omega/(d(\omega)|\omega|), 0, 0)$ acting over the time interval $[T - d(\omega), T]$, with χ_ω the characteristic function of ω and $|\omega| = \int_\omega dx$ the volume of ω. We give stability factors corresponding to a computation starting at $t = 12$ for different $d(\omega)$ in Tab. II.

The residuals are of order 1, and the product of mesh size and stability factor increases with decreasing $d(\omega)$. This example supports our belief that pointwise quantities are more difficult to compute than mean values. In Fig.16 we note the decrease in the dual solution with (backward) time for $d(\omega)$ when the initial data is convected out of the computational domain, indicating that the error in the mean value of the solution over the time interval $[20 - d(\omega), 20]$ is independent of the error in the solution for $t < 15$ in this case.

8.1.2. Step down We consider now a channel with no slip walls and 1×1 quadratic cross-section of length 4, and a step down of height and length 0.5. We have a parabolic inflow condition $u = (64(1-y)(y-0.5)z(1-z), 0, 0)$, and we use a transparent outflow condition. We set $\nu = 10^{-3}$, and we compute on a regular tetrahedral mesh, with $h = 1/32$, using the cG(1)cG(1)-method. We start from $u = 0$ at time $t = 0$, and in Fig.8.1.2 during the start up phase we can follow the formation of the recirculation zone behind the step and also the formation of a corresponding zone where the flow separates from the top boundary.

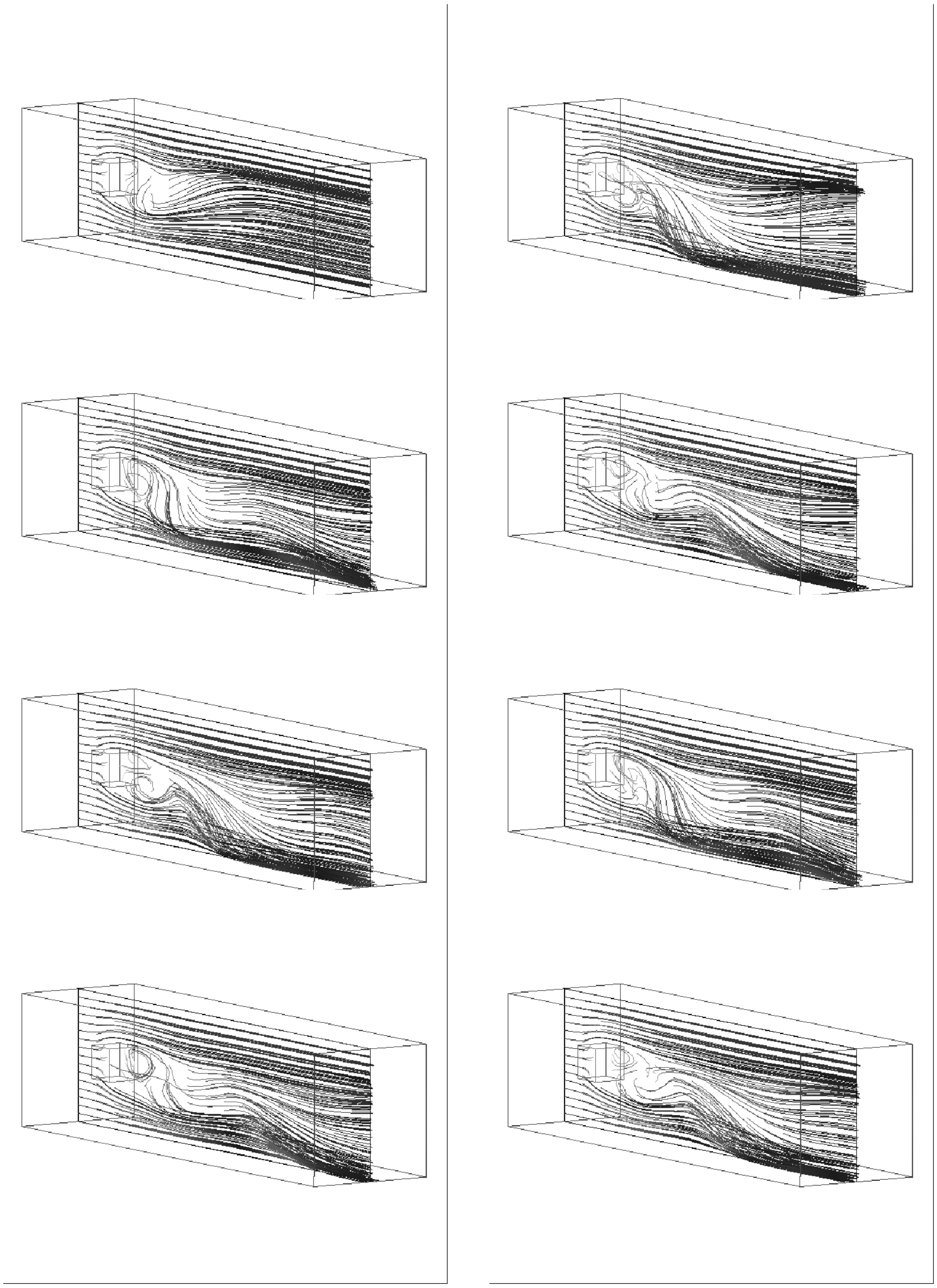

Figure 15. Time evolution for the bluff body problem with $t = 2, 4, ..., 16$

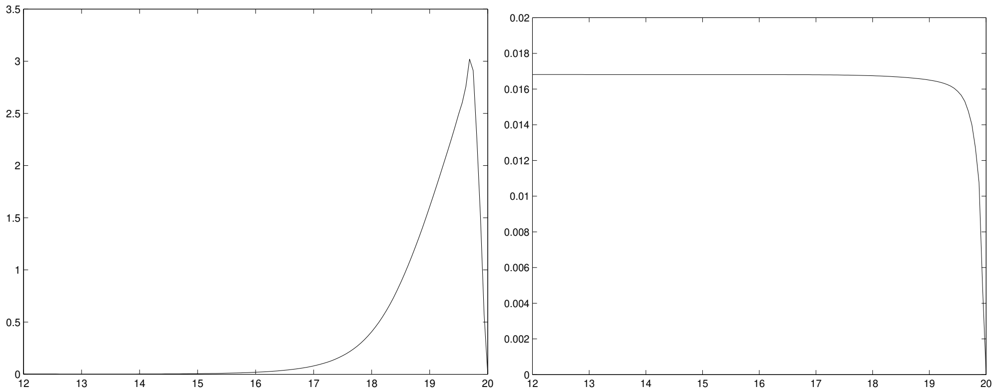

Figure 16. Bluff body: $\|\varphi(t)\|_1$ for $d(\omega) = 1/4$ (left), and for the drag problem (right), as functions of time.

Table III. Step down: stability factors

$d(\omega)$	$S_{0,1}$	$S_{0,2}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
1/8	30.9	16.2	836.8	124.0	138.4	278.4
1/4	22.9	4.2	533.4	39.0	48.9	46.8
1/2	10.8	2.4	220.3	10.5	16.1	25.2

We consider the problem of computing a space-time average over the time interval $[9, 10]$and a spatial cube $\omega \subset \Omega$ centered at $(1.5, 0.5, 0.5)$with side length $d(\omega)$, starting the computation at $t = 5$. The residuals are of order 0.1, and in Tab.III we present stablity factors corresponding to different $d(\omega)$. In Fig.8.1.2 we see how the dual "final data" is transported backwards in space-time.

8.1.3. Drag of Bluff body We reconsider the bluff body problem with now the objective of computing the average of the drag force over a time interval $[12, 20]$. The corresponding data for the dual problem is a boundary condition $u = (1, 0, 0)$ on the faces of the bluff body. We note that the norm of the dual solution after an initial (backwards in time) growth approaches a stable value, see Fig.16. The corresponding dual solution is shown in Fig.19. The dual solution does not go to zero as in the previous cases, indicating (not very surprising) that the computation of a time average is dependent of the quality of the solution during the whole time interval, in contrast to the previous cases in Section 8.1.1-8.1.2 where the dependence of the quality of the solution for previous time was decreasing with (backward) time.

We observe that the stability factors for the time mean-value of the drag are considerably smaller than those for more local space-time mean-values.

8.1.4. Jets We consider a channel with no slip walls and 1×1 rectangular cross section of length 4 with an obstacle with four quadratic holes of size 0.25 at $x_1 = 0.5$. We have a parabolic inflow condition and we use a transparant outflow condition. The inflow condition causes the flow to form 4 high velocity jets through the holes, and in the domain behind the obstacle we get a irregular unsteady

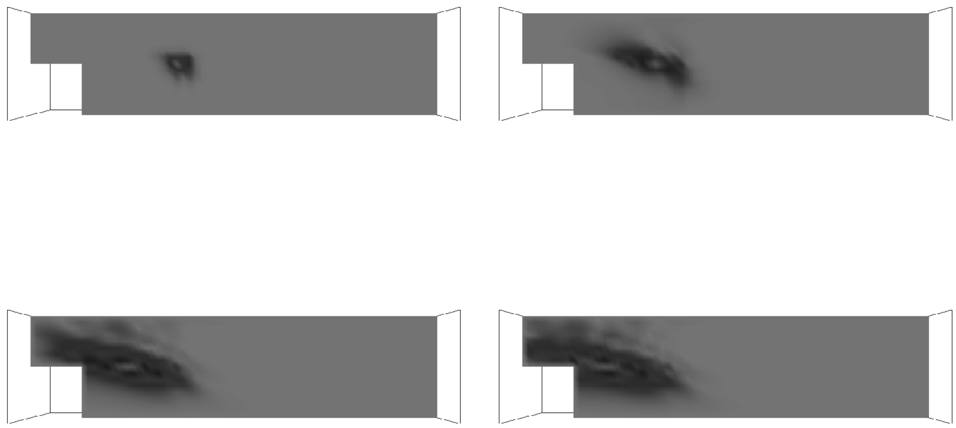

Figure 17. Dual solution for Step down problem for $d(\omega) = 1/8$, and backward time.

Table IV. Bluff body drag: stability factors

$S_{0,1}$	$S_{0,2}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
0.13	0.11	1.6	0.03	0.43	0.06

Table V. Jets: stability factors corresponding to a space-time average over $\omega \times [T_0, 10]$, with $d(\omega) = 0.125$.

T_0	$S_{0,1}$	$S_{0,2}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
1/4	4.2	4.2	189.4	28.7	31.7	39.5
1/2	3.0	2.9	101.9	8.6	14.0	12.4

flow pattern. We compute on a tetrahedral mesh, with $h = 1/32$, locally refined to $h = 1/64$ for $0.125 \leq x_1 \leq 1.125$, and $\nu = 10^{-4}$, using the cG(1)cG(1)-method. We start from $u = 0$ at time $t = 0$, and we compute to time $t = 10$. The residuals are of order 1, and in In Tab. V we consider the case of computing a space-time average over the spatial cube ω with $d(\omega) = 0.125$ for different time intervals $[T_0, 10]$, and we find that we get larger stability factors for decreasing T_0 We then fix the time interval to $[9, 10]$ and we vary the side length $d(\omega)$ of the spatial cube ω, resulting in larger stability factors for smaller $d(\omega)$, see Tab. VI.

8.1.5. Turbulent flow We now consider the case of the Couette flow from Section 7.6, with results taken from Hoffman, 2002. We consider computing the space-time average as in Section 7.6. In Fig.23 we show snapshots of the dual solutions for $d(\omega) = 0.5, 0.25, 0.125$, and in Fig.22 we plot the (backward) time evolution of the L_1-norms in space of the corresponding dual solutions. In general,

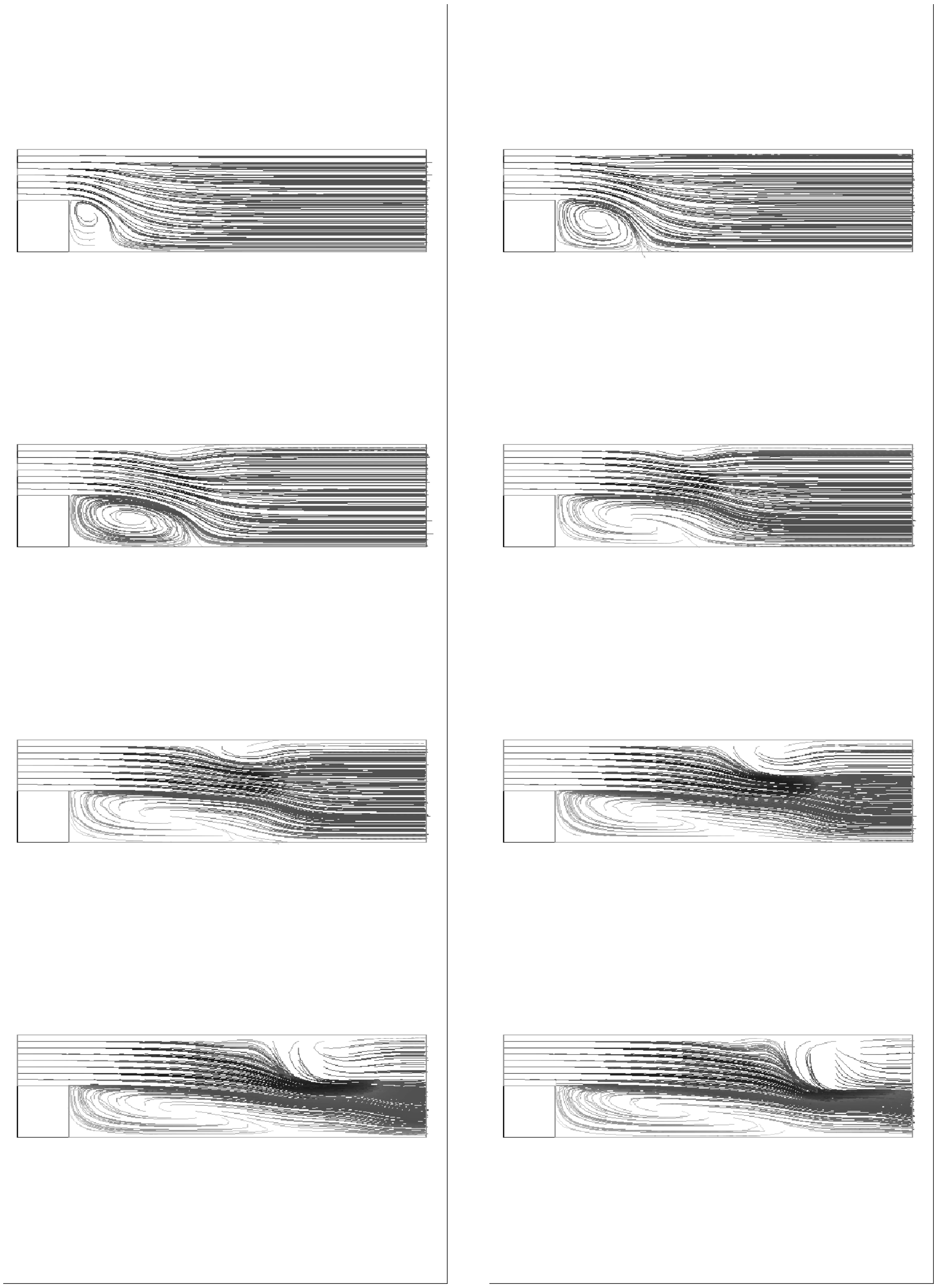

Figure 18. Time evolution for the step down problem with $t = 1, 2, ..., 8$

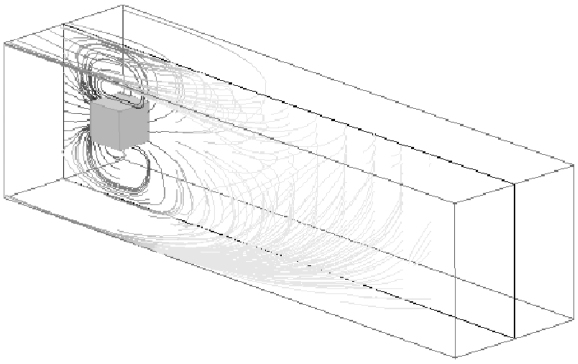

Figure 19. Dual solution for the Drag problem at $t = 19$

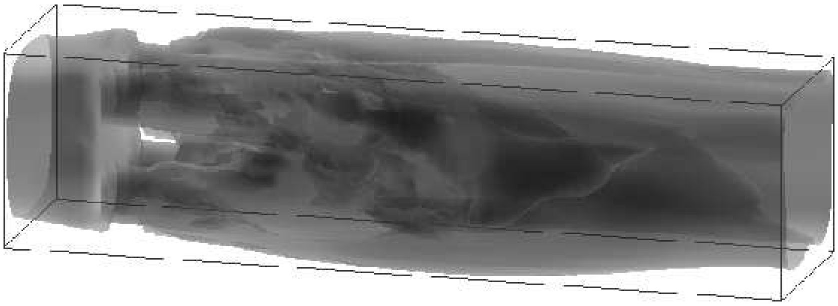

Figure 20. Jet problem isosurfaces at time $t = 10$.

Table VI. Jets: stability factors corresponding to a space-time average over $\omega \times [9, 10]$, with side length $d(\omega)$.

$d(\omega)$	$S_{0,1}$	$S_{0,2}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$	$S_{1,4}$
1/16	5.2	5.5	189.8	13.0	21.1	23.1
1/8	5.0	5.4	164.7	12.1	20.0	22.9
1/4	4.1	5.0	115.6	7.8	15.2	19.0
1/2	2.3	4.0	55.9	3.6	8.5	7.9

the area under these curves corresponds to the stability factors.

From Fig.21 we see that the residuals are fairly constant in time, whereas the solution of the dual problem is growing (backwards) in time.

In the initial phase a dual solution grows through the action of the force ψ during the time interval $[30 - d(\omega), 30]$. This initial growth is larger for small $d(\omega)$, which may be explained by the larger quotient $q(d(\omega)) = \text{surface area/volume} = 6d(\omega)^2/d(\omega)^3 = 6/d(\omega)$ for smaller $d(\omega)$. This phenomenon is also present in the laminar flows considered in the previous sections and is caused by the divergence free condition, which is active in increasing the dual solution, and depends on $q(d(\omega))$. Since this phenomena is connected to the divergence free condition, we can observe the same phenomena also in the simple problem $\dot{\varphi} + \nabla q = \psi$, $\nabla \cdot \varphi = 0$.

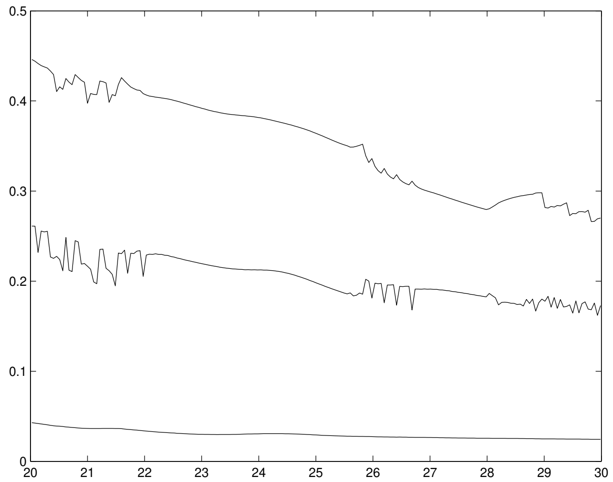

Figure 21. $\|R_4(U_h)\|$ (upper), $\|R_1(U_h, P_h)\|$ (middle), and $\|R_M(u, U_h)\|$ (lower) as functions of time.

In the next phase, when the force ψ is zero, there is a growth due to the reaction term $\nabla U_h \cdot \varphi$ in th dual problem (22). If U_h is non-smooth as in a turbulent flow, this is a reactive term with potentially strong production since ∇U_h is large, which in principle could cause the dual solution to grow very large. However, this pessimistic scenario does not take place, since the growth of the dual solution in reality is rather modest, whcih must correspond to cancelations in the action of the reaction term with alternating signs of the components of ∇U_h. Visibly, more cancellations must take place for larger mean values as the coresponding dual solution is smaller. This is good news and indicates computability of mean values in turbulent flow

In particular a time average of a certain quantity may be expected to be more computable than the same quantity at a specific time. This supported by Figure 22, where in the upper right figure now the force ψ is active over the whole time interval corresponding to a time mean-value.

The growth of the dual solution is expected to be weaker for laminar flows, for whcih the reaction term in the dual equation has smaller coefficients. As an example, we plot in Figure 22 the dual solution linearized at the laminar Couette flow $u = (2(y - 0.5), 0, 0)$, with $\nu^{-1} = 100$. We see that the dual solution is initially quickly damped followed by a slow further decrease caused by diffusive mechanisms. In Figure 22 we also plot the dual solution linearized at a laminar flow with $\nu^{-1} = 10000$, which corresponds to a more unstable flow. In this case we get an initial growth of the dual solution

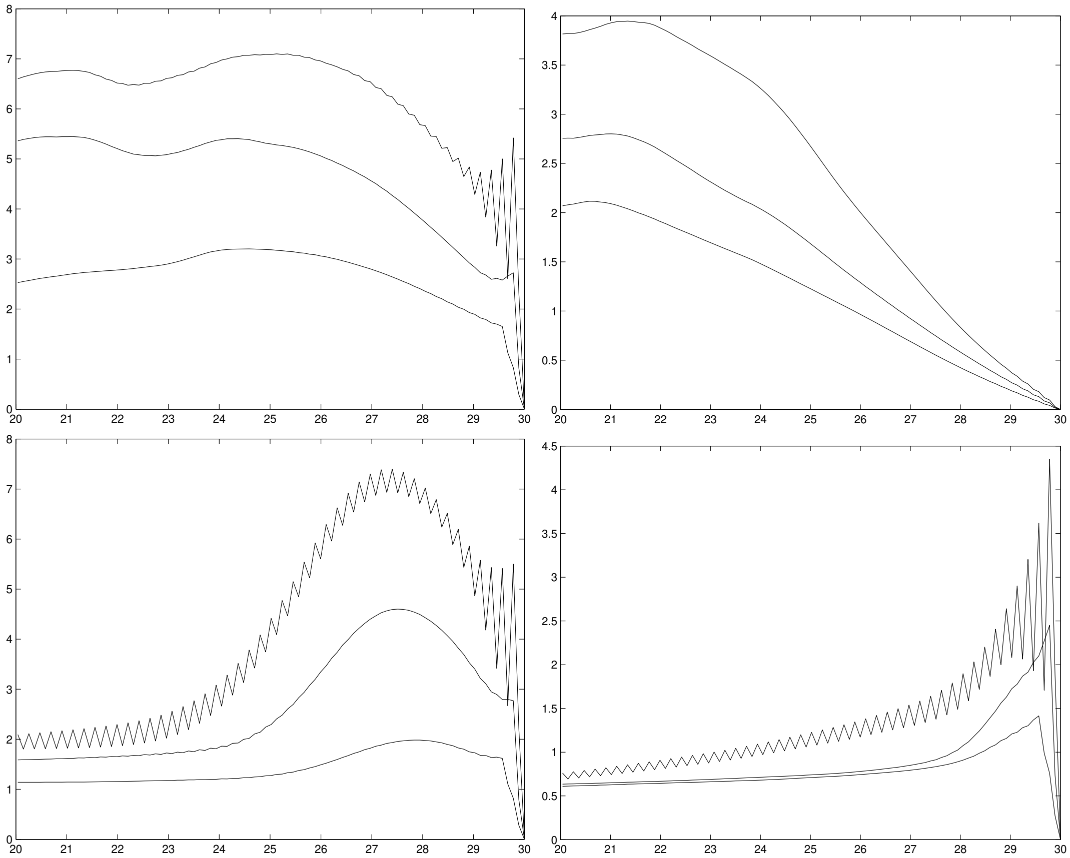

Figure 22. $\|\varphi\|_1$ for $d(\omega) = 0.5, 0.25, 0.125$ linearized at a turbulent flow (upper left), time average over $\omega \times [20, 30]$ for a turbulent flow (upper right), laminar flow with $\nu^{-1} = 10000$ (lower left), and laminar flow with $\nu^{-1} = 100$ (lower right), as functions of time.

due to the reaction term, after which we have a similar scenario as in the case of $\nu^{-1} = 100$. We thus have evidence that (as it should be) is more computationally demanding to compute a more unstable flow (with a larger Reynolds number Re), even though the exact solution is the same.

9. Summary

We have presented a general framwork for adaptive computational simulation of fluid flow based on Galerkin finite element discretization and subgrid modelling, together with a posteriori error estimation in terms of discretization and modeling residuals multiplied with stability factors/weight obtained by solving an associated linearized dual problem. We have computed stability factors/weights for different outputs in both laminar and turbulent flow, and given evidence of computability based on the size of residuals and stability factors. In particular, we have shown that stability factors for mean value outputs in turbulent flow may be of moderate size indicating computability of turbulent flow. We have also

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

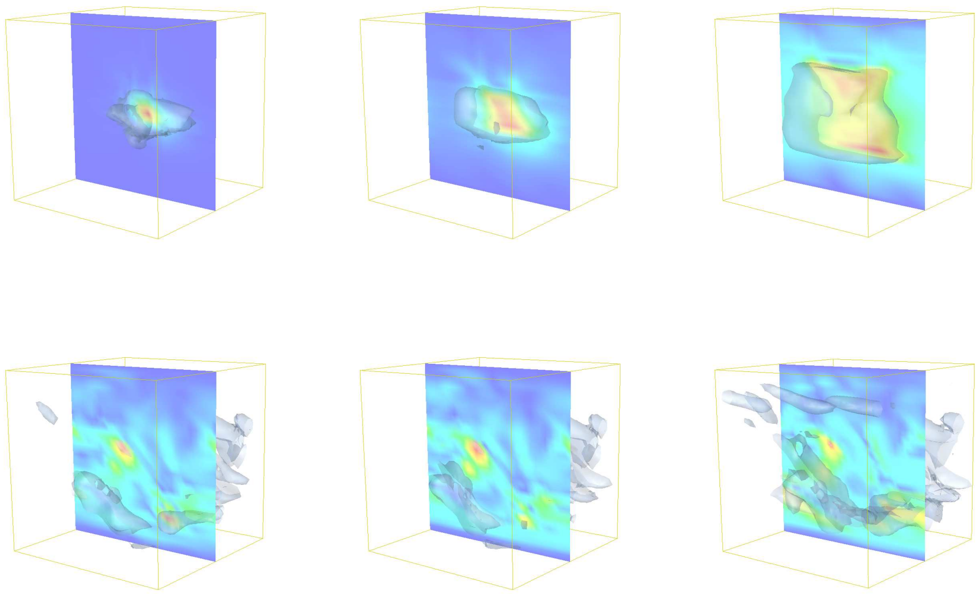

Figure 23. Streamwise high dual velocity isosurfaces: $d(\omega) = 0.125$ (left), $d(\omega) = 0.25$ (middle), $d(\omega) = 0.5$ (right), for $t = 29.5, 24$.

discussed adaptive methods with automatic selection of discretization (local size of space mesh and time steps), and subgrid modeling including various types of Ansatz for the subgrid model.

REFERENCES

Bardina J, Ferziger JH, Reynolds WC. Improved subgrid model for large-eddy simulation. *AIAA paper 80-1357*, 1980.

Becker R and Rannacher R. An optimal control approach to a posteriori error estimation in finite element methods. Acta Numerica, pp 1-102, 2001.

Eriksson K, Estep D, Hansbo P and Johnson C. Introduction to Adaptive Methods for Differential Equations. Acta Numerica, pp 105–158, 1995.

Eriksson K, Estep D, Hansbo P and Johnson C. *Computational Differential Equations*, Cambridge University Press, 1996.

Eriksson K, Estep D and Johnson C. *Applied Mathematics: Body and Soul, Vol I-III*, Springer, 2003.

Eriksson K, Johnson C and Logg. *Parabolic problems* in Encyclopedia of Computational Mechanics, 2003.

Frisch U. *Turbulence - The Legacy of A. N. Kolmogorov*, Cambridge Univ. Press, 1995.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.

Gatski TB, Hussaini MY and Lumley JL. *Simulation and Modeling of Turbulent Flow*, Oxford Univ. Press, 1996.

Germano M, Poimelli U, Moin P and Cabot W. A dynamic subgrid scale eddy-viscosity model. Phys. Fluids A 3, 1760, 1991.

Giles M, Larson M, Levenstam M and Süli E. Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow, Technical Report NA-76/06, Oxford Univ. Computing Laboratory, 1997.

Hoffman J, Johnson C and Bertoluzza S. Dynamic Subgrid Modeling I. Preprint. Chalmers Finite Element Center 2000-04a, 2000.

Hoffman J. Dynamic Subgrid Modeling II. Preprint. Chalmers Finite Element Center 2000-04b, 2000.

Hoffman J. Dynamic Subgrid Modeling for Convection-Diffusion Equations with Fractal Coefficients. *Multiscale and Multiresolution Methods: Theory and Applications*, Lecture Notes in Computational Science and Engineering, **20**, Springer Verlag, 2001.

Hoffman J. Dynamic Subgrid Modeling for Convection-Diffusion-Reaction Systems with Fractal solutions. *International Journal for Numerical Methods in Fluids*, Vol **40**, 583-592, 2002.

Hoffman J and Johnson C. Adaptive finite element methods for incompressible fluid flow. *2001 NATO-RTO Lecture Series*, Lecture Notes in Computational Science and Engineering, Springer Verlag, 2002a.

Hoffman J and Johnson C. On transition to turbulence in parallel shear flow. submitted to *Journal of Mathematical Fluid Mechanics*, 2002b.

Hoffman J. A posteriori error control in turbulent flow. submitted to *SIAM Journal of Scientific Computing*, 2002.

Hughes TJR, Mazzei L, Jansen KE. Large Eddy Simulation and the variational multiscale method. *Computing and Visualization in Science, **3**, 47-59*, 2000.

Johnson C. Adaptive Finite Element Methods for Conservation Laws. *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Springer Lecture Notes in Mathematics, Springer Verlag, 1997.

Lesieur M. *Turbulence in Fluids*. Kluwer Academic Publishers, London, England, 1997.

Liu S, Meneveau C, Katz J. On the properties of similarity subgrid-scale models as deduced from measurements in turbulent jet. *J. Fluid Mech., **275**, 83-119*, 1994.

Rannacher R. Finite element methods for the incompressible Navier Stokes equations, Preprint Intsitute of Applied Mathematics, Univ. of Heidelberg, 1999.

Scotti A, Meneveau C. Fractal dimension of velocity signal in high-Reynolds-number hydrodynamic turbulence, *Phys. Review E **51**, 5594*, 1995.

Schäfer M, Turek S. Benchmark computations of laminar flow around cylinder, in E.H. Hirschel (editor), *Flow Simulation with High-Performance Computers II*, Volume 52 of Notes on Numerical Fluid Mechanics, pp. 547-566, Vieweg, 1996.

Wagner G and Liu WK. Turbulence simulation and multiple scale subgrid models, to appear in Computational Mechanics, 1999.

Encyclopedia of Computational Mechanics. Edited by Erwin Stein, René de Borst and Thomas J.R. Hughes.