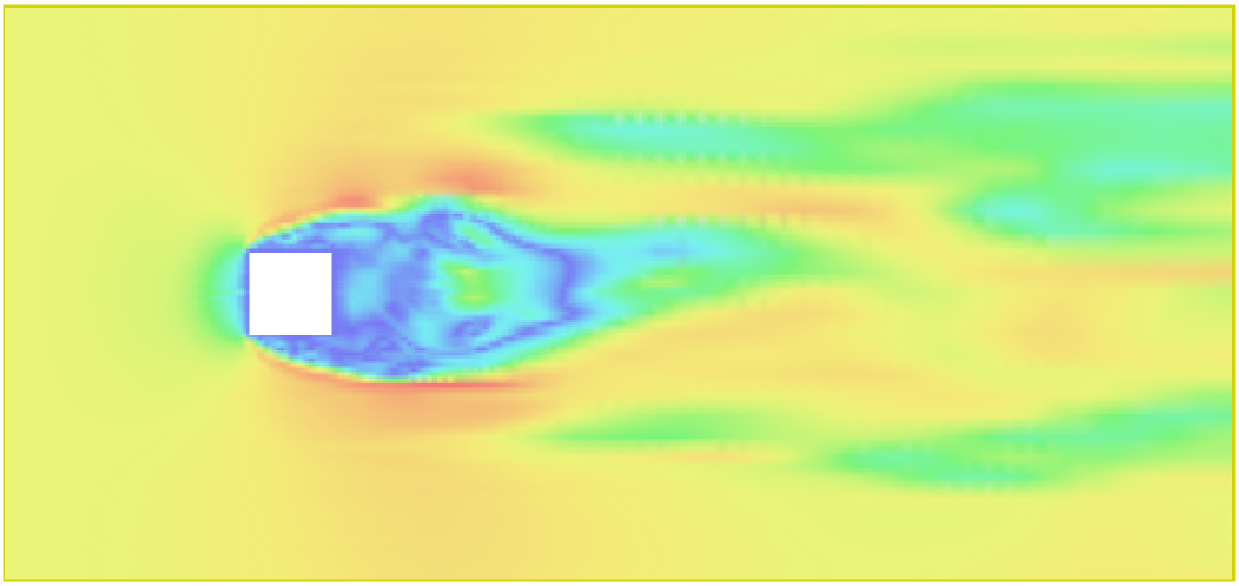
CHALMERS
FINITE ELEMENT CENTER



PREPRINT 2003–23

Adaptive DNS/LES: a New Agenda in CFD

Johan Hoffman and Claes Johnson

Chalmers Finite Element Center
CHALMERS UNIVERSITY OF TECHNOLOGY
Göteborg Sweden 2003

CHALMERS FINITE ELEMENT CENTER

Preprint 2003–23

Adaptive DNS/LES: a New Agenda in CFD

Johan Hoffman and Claes Johnson

CHALMERS

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg Sweden
Göteborg, November 2003

Adaptive DNS/LES: a New Agenda in CFD
Johan Hoffman and Claes Johnson
NO 2003–23
ISSN 1404–4382

Chalmers Finite Element Center
Chalmers University of Technology
SE–412 96 Göteborg
Sweden
Telephone: +46 (0)31 772 1000
Fax: +46 (0)31 772 3595
www.phi.chalmers.se

Printed in Sweden
Chalmers University of Technology
Göteborg, Sweden 2003

ADAPTIVE DNS/LES: A NEW AGENDA IN CFD

JOHAN HOFFMAN AND CLAES JOHNSON

Abstract. We show that using adaptive finite element methods it is possible to accurately simulate turbulent flow with the computational power of a PC. We argue that this possibility should set a new agenda in CFD. The key to this break-through is (i) application of the general approach to adaptitive error control in Galerkin methods based on duality, coupled with (ii) crucial properties of turbulent flow allowing accurate computation of mean value quantities such as drag and lift without full resolution of all scales.

1. Introduction and Perspectives

Turbulence represents an outstanding open problem of fluid mechanics. The Navier-Stokes equations formulated in 1822-45 are believed to accurately describe fluid flow over a very wide range of applications and flow characteristics including both laminar and turbulent flow. The *Reynolds number* $Re = \frac{UL}{\nu}$, where U is a characteristic flow velocity, L a characteristic length scale, and ν the viscosity of the fluid, is often used to characterize fluid flow. If $Re \sim 10$ or smaller, then the flow is viscous and the flow field is ordered and smooth or *laminar*, while if $Re \geq 100$, then the flow in general will show features of non-ordered non-smooth time-dependent or *turbulent* flow. In many applications of scientific and industrial importance Re is very large, of the order 10^6 or larger. A turbulent velocity field varies on a range of scales in space from a largest length 1 assuming $L = 1$ to a smallest scale of size $Re^{3/4}$. To accurately resolve a turbulent flow at $Re = 10^6$ in a *Direct Numerical Simulation DNS* would require of the order $Re^3 = 10^{18}$ mesh points in space-time, and thus would be impossible on any forseeable computer. On the other hand DNS at $Re = 10^2$ on a PC and at $Re = 10^3$ on a supercomputer today appears feasible.

To overcome the impossibility of DNS at higher Reynolds numbers various techniques of *turbulence modeling* have been attempted, with *Large Eddy Simulation LES* showing most promise. In a LES one resolves the coarser flow scales and seeks to model the influence of unresolved small scales on resolved larger scales in a *turbulence model* or *subgrid model*. Various subgrid models have been proposed, but no clear answer to the question of the feasibility of LES in simulation of turbulence has been given. It thus seems

Date: November 26, 2003.

Key words and phrases. adaptive finite element method, duality, a posteriori error estimate, turbulent viscous incompressible flow, direct numerical simulation, large eddy simulation, surface mounted cube.

Johan Hoffman, Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012-1185, USA *email*: hoffman@cims.nyu.edu

Claes Johnson, Department of Computational Mathematics, Chalmers University of Technology, S–412 96 Göteborg, Sweden, *email*: claes@math.chalmers.se.

fair to describe the simulation of turbulence at high Reynolds numbers as the main open problem of *Computational Fluid Dynamics CFD* today.

The purpose of this note is to present a new approach to this fundamental problem which is now being opened using *adaptive finite element methods*. The new approach may be described as *Adaptive DNS/LES* and is a direct application of the general approach to adaptive error control in Galerkin methods for differential equations based on duality developed over the last 15 years, see [5, 4, 2, 3, 12, 13], and references therein. The basic technical tool in this approach is a representation of the error in a *quantity of interest* in the form of a space-time integral of *residuals* of computed solutions multiplied by *weight functions* coupled to the solution of an associated *linearized dual problem*. From the error representation one may derive an *a posteriori error estimate* for the quantity of interest and formulate a corresponding adaptive method including both a *stopping criterion* and a *mesh modification criteria* to reach the stopping criterion. This way we are able to guarantee that a chosen quantity of interest is computed to a certain tolerance at minimal computational cost, that is, we can make the computation both *reliable* and *efficient*. For incompressible flow, applications of adaptive finite element methods based on this framework have been increasingly advanced with computation of quantities of interest such as the *drag force* for 2d stationary benchmark problems in [2, 6], and drag and *lift forces* and *pressure differences* for 3d stationary benchmark problems in [9]. In [10], time dependent problems in 3d are considered, and the extension of this framework to LES is investigated in [7, 8]. This extension is crucial and opens for a large wealth of real world applications.

Since the approach to adaptive error control via duality is very general, we may use it in particular to solve the Navier-Stokes equations at high Reynolds numbers, that is for computational simulation of turbulent flow. The key question then becomes the *computational cost*. The adaptive method will produce a reliable result at a minimal cost, and the main remaining question is then: What is then the cost? In particular, can we compute turbulent high Reynolds number flows on a PC?

The goods news is now that the cost seems reasonable, at least if the quantity of interest is a mean value such as the mean value in time of the drag force of a bluff body. Thus, to accuratetly compute the drag force of a bluff body at Reynolds number say 10^6, our experience with Adaptive DNS/LES indicates that this is possible with less than 10^9 mesh points in space-time, instead of the 10^{18} required in direct DNS. This obviously makes a huge difference and turns a problem from being uncomputable even on massive clusters of supercomputers, to become computable on a PC. We give evidence below.

So, how can it be possible to accurately compute certain aspects of turbulent flow on a PC ? To give a rough idea of the answer , which is given in more detail below, we recall that the error representation by duality describes the error in the quantity of interest, such as the drag of a bluff body, in terms of a *discretization residual* related to the Galerkin discretization of Navier-Stokes equations, and a *modeling residual* related to the linearization behind the dual problem, combined with weights depending on the dual solution. The modeling residual occurs because of in the linearization, and we compare the computed solution with a local mean value of the true solution, which may be pointwise approximated, rather than the true solution itself which may not be pointwise approximated, unless we are willing to do a full DNS, which we are not.

The crucial fact is now the size of the weights which directly couples to the size of the dual solution and its derivatives. The remarkable experience from our computations of these weights in turbulent flows by solving the linearized dual problem computationally, is now that the weights are *not as large as pessimistic predictions would indicate*, but rather much smaller. The linearized dual problem is a linear convection-diffusion-reaction problem with the crucial coefficient of the reaction term being the gradient of the computed velocity field. This coefficient is large in turbulent areas, typically of size $\sqrt{Re}$, and thus potentially generating exponential growth of the dual solution. However, the gradient of the velocity is rapidly fluctuating and the net effect of the reaction term turns out to be much smaller due to cancellation and thus the corresponding net growth of the dual solution is much slower than worst case exponential.

The net result is that the drag of a bluff body may be accurately computed using Adaptive DNS/LES without solving *all* of the fine scale features. However, the error representation also indicates that it is critical to resolve *some* of the small scale features, such as separation points which determine gross aspects of the flow pattern. The resulting method Adaptive DNS/LES thus (adaptively) uses LES without full resolution of all scales in the larger part of the turbulent region and DNS in the laminar parts of the flow.

The next pertinent question to address is then what subgrid model should be used in the LES? The good news here is that just about any turbulence model approximating the Reynolds stresses as a dissipative term will work fine, as long as the dissipation is large enough. Our implementation of Adaptive DNS/LES uses a standard least-squares stabilized Galerkin method and this simple recipee seems to be fully adequate as a turbulence model in the LES. The test of "sufficient dissipation" is very simple: if the the finite element method produces discretization residuals which are not larger than necessary, then there is enough dissipation in the numerics, or the other way around: If the numerics does not explode, then there is sufficent numerical dissipation. It appears that meaningful numerics must have more dissipation than the true flow, and therefore it seems impossible to get reasonable results with too little dissipation.

The main message here is thus that the choice of turbulence model in the LES is quite free; just about any reasonable dissipative model will work, for example the Smagorinsky model or least-squares stabilization. This couples to the following property of turbulent flow around a bluff body, the turbulent dissipation per unit volume (in the turbulent wake behind the body), is nearly constant independent of the viscosity below a certain threshold corresponding to Re of size about 100. This constancy may be observed through the refinement procedure in the adaptive method, which with increasing mesh refinement may be seen as a decrease of effective viscosity in the numerics. So the intensity of turbulent dissipation turns out to be constant once the Reynolds number is above a not so large threshold. But as the viscosity decreases the drag increases, which couples to the observation that the turbulent wake behind a bluff body increases in volume as the viscosity decreases. This increase in volume of the turbulent wake seems to be well captured by adaptively using

DNS in the surrounding laminar flow including a strong shear layer around the turbulent wake.

The constancy of the dissipation intensity couples to the observation that if the quantity of interest is a global quantity such as the drag, then "it does not matter" on what scale the turbulent dissipation actually takes place. This means that it may be perfectly possible to compute the drag of a bluff body without resolving all of the fine scales of the flow, but only some.

We sum up as follows: using Adaptive DNS/LES it seems possible to accurately compute (on a PC) the drag of a bluff body at high Reynolds number involving turbulent flow. The basic reasons for this new possibility are as follows: (i) we use DNS with full resolution in laminar parts of the flow, (ii) it turns out that LES with a simple dissipative subgrid model is adequate in the rest of the domain.

The net result is that we may accurately compute turbulent high Reynolds number flow on a PC, without having to invent sophisticated turbulence models. The DNS will then cover the laminar parts of the flow and certain critical parts of the flow including separation points and surfaces separating turbulent and laminar parts of the flow. The Adaptive DNS/LES method automatically takes care of the DNS vs LES aspect based on a posteriori error estimation.

We present below computational results for a *generic bluff body problem* in the form of the computation of the *drag coefficient* (normalized drag force) c_D at $Re = 40.000$ for a surface mounted cube. We find that $c_D = 1.48$ using about 400.000 mesh points in space. The only other result (!) available in the literature seems to be that of [14], where values in the range $c_D = 1.12 - 1.24$ are presented. Surprisingly, no measurements seem to be available, maybe because of lacking motivation because of lacking computations and little interest from the car industry in cubic vehicles. A measured value of $c_D = 1.3$ is reported [1] for a box with height-width ratio $5 : 1$. It seems very likely that the c_D for a cube would be larger than that for a more slender box.

1.1. **The novelty of Adaptive DNS/LES.** We believe the Adaptive DNS/LES introduces a true new element in CFD, specifically represented by the computation of the dual solution, which carries the coupling between residuals and output quantities. To compute dual solutions has been a trademark of our approach to adaptivity since the early 90s, with full time-dependent 3d problems including the dual linearized Navier-Stokes equations first being realized in [10]. The break-through experience was that, in contrast to early pessimistic predictions, the cost of solving the dual problem was very well affordable, and with the information from the dual problem available the computation of a quantity of interest could be made both relaible and efficient. Further, with a global quantity of interest such as the drag of a bluff body, the dual solution turns out to be relatively smooth and thus the drag is computable without excessive computational effort. So there are here two key elements: first we have to solve the dual problem, and then make the observation that the dual solution is reasonably smooth. If we can do this, then we can compute aspects of turbulent flow on a PC.

Note the close coupling between the quantity of interest (or output) and the dual solution: the data to the dual problem directly reflects the output, with typically the data being smoother for a global output than a local one. Again, this means that we may compute with Adaptive DNS/LES a global quantity such as drag on a PC, while if we would seek to compute the local flow velocity in space-time, then we would need (at least) DNS (which is not affordable).

Thus, there are aspects of turbulent flow which we cannot compute, such as the flow velocity at a specific point in the turbulent wake behind a bluff body, while a global quantity such as the drag may be computable. Of course this seems to match with our experience: we may believe that the drag of our car can be computed, but not the exact location in space-time where a certain dust particle will end up on our rear wind shield.

1.2. **On Turbulence Modeling.** Our results indicate that, in a certain sense, the computational modeling of turbulence is easy: just use any reasonable dissipative model, combined with selected sparse use of DNS.

On the other hand, there may be cases where the DNS (with such a simple LES turbulence model) may be required in larger portions of the domain, and then again the cost would become prohibitive. Such cases may occur e.g. in reactive flow, where the turbulent fine scale flow features may influence the reaction rates or effective reaction diffusivities. In such cases, one may hope that again the DNS may be used sparsely, in this case by local DNS in space-time combined with extrapolation. More precisely, in a local DNS over a small volume, one would be able to resolve the fine scales and thereby determine effective reaction diffusivities relating to the turbulent mixing, and then export these effective diffusivities to other parts of the domain with similar flow patterns.

Further, through the error representation we are able to quantitively compare different turbulence models in LES, and thus adaptively choose the best from a given set of models or get clues on how to develop improved models. The turbulence modeling problem is a truely "open" problem: the turbulence model may be a free invention or artifice without clear coupling to known physics.

1.3. **On Classical Turbulence Models such as the $k - \epsilon$ model.** The classical approach to turbulence, not including LES, is to seek transport equations for quantities like the *turbulent kinetic energy* k and *turbulent dissipation* ϵ. The difficulty lies in the derivation of such equations from the Navier-Stokes equations, where no rational analytical procedure (or so called *closure*) seems to be available, and therefore the modeling seems to have very limited success, unless possibly in some cases where a model may be fitted to a particular flow based on experimental data, without being able to fit other cases. There is a lot of critics of $k - \epsilon$ models and the hopes of great improvement today seem to be small.

1.4. **On Dynamic LES.** The classical LES turbulence model is the Smagorinsky model introducing a turbulent viscosity of the form $CH^\alpha|\nabla u|$, where H is a parameter representing the *smallest scale* of the flow, α is a positive number (normally $\alpha = 2$) and C is a positive constant or coefficient depending on space and time. In *Dynamic LES* one seeks to determine the α and C that would best model a particular flow. The term *dynamic* refers to the fact that one seeks to determine these best values based on computation and some form of extrapolation. There are also *mixed* models where in addition to Smagorinsky viscosity one seeks to use scale similarity to capture whatever is not captured in Smagorinsky viscosity. Such effects are referred to as *backscattering* and are supposed to model a possible influence on large scales from small scales. Again the success with such mixed models has been limited and no conclusive experience seems to exist.

As already indicated, our experience is that LES, when adaptively coupled with DNS, may work well with e.g. a standard Smagorinsky model or numerical viscosity, and that the further elaboration in dynamic mixed models may not be worthwhile.

1.5. **What numerical viscosity do we use?** We remark that in our implementation we use a specific numerical viscosity arising from a least-squares stabilization of the Galerkins method. This numerical viscosity seems to be close to a standard Smagorinsky model concerning its action on the finest scales of the turbulent flow, where most of the turbulent dissipation takes place, while the stabilization term introduces little dissipation on coarser scales. In particular, our viscosity as well as the standard Smagorinsky model, seems to be consistent with the Kolmogorov similarity principles for turbulent flow.

1.6. **On Understanding Turbulence.** We claim that using Adaptive DNS/LES one may accurately compute (certain aspects of) turbulent flow. In principle this is a fully automatic procedure for which no "understanding" of turbulence seems to be needed. We just compute and then observe what we have computed. Of course after having made observations based on computation we may seek to bring some order and find some structure in what we see, possibly using some simple mental models connected to the flow. This is analogous to the observation that (i) we are able to live (or "compute") our lives without too much "understanding" (which is particularly evident when we observe other people). On the other hand, sometimes we seem to be able to collect some "understanding"of our own lives from the experience we get by living (computing) our lives. Thus, computing and "understanding"are not in opposition; from computing we may get "insight". However, to gain insight without ever computing or making practical experiments may be very difficult, at least for human beings.

1.7. **Turbulent Compressible Flow.** In this note we focus on turbulent incompressible flow. The general scope presented applies as well to compressible flow. The typical flow feature of compressible flow as shock waves may be expected to involve turbulent flow: most likely the flow inside a shock on a the wing of a 777 will be turbulent, as well as in the wake behind the wing. Again Adaptive DNS/LES should allow accurate computation without invoking RANS models as is now standard. In a RANS model one seeks to find and solve equations for global mean values in space-time. It seems that today few advocate RANS as a method to compute turbulent flow.

1.8. **Further applications.** There is a large variety of bluff body problems, which may be viewed as variants of the generic bluff body problem of the surface mounted cube, including sails, ship hulls, cars, trains, aeroplanes, turbomachines, and many more... It is clear that a possibility of accurate computation turbulent flow may strongly influence the development in these fields.

1.9. **The Clay Prize.** One of the 10 Clay $\$10^6$ Prize problems concerns a mathematical proof of existence of a (smooth) solution to the Navier-Stokes equations. Despite strong efforts for over 70 years by very clever mathematicians little progress has been made towards a solution. It is possible that computational evidence on the nature of (turbulent) solutions may give added insight and open for a solution. Maybe.

2. A New Agenda in CFD?!

Adaptive DNS/LES introduces a true new element in CFD, and thereby opens new possibilities of computational simulation of complex flows including turbulent flows, with duality playing the key role. We strongly believe that now the agenda in CFD will have to be reformulated, and computation of dual solutions to gain sensitivity information relating to discretization and modeling, should be the new standard.

To motivate our stand point we now present results obtained by applying Adaptive DNS/LES to a generic bluff body problem in the form a surface mounted cube at $Re =$ 40.000.

3. A generic bluff body problem: surface mounted cube

We consider the problem of a turbulent flow around a surface mounted cube, investigated in [15, 16, 14, 8]. In our computational model we use the Navier-Stokes equations to model the incompressible fluid around a cubic body of dimension $H \times H \times H$ that sits on the floor of a rectangular channel of length $15H$, height $2H$, and width $7H$, centered at $(3.5H, 0.5H, 3.5H)$. At the inlet we use a velocity profile interpolated from experiments, we use no slip boundary conditions on the body and the vertical boundaries, slip boundary conditions on the lateral boundaries, and a transparent outflow boundary condition. The viscosity ν is chosen to give a Reynolds number $Re = U_b H/\nu = 40.000$, where we have used $U_b = 1.0$. We compute on tetrahedral meshes using the cG(1)cG(1), see Section 4.2, for both the primal and the dual problem.

In Figure 1 we show a snapshot of the solution and the corresponding computational mesh after 13 adaptive mesh refinements, using Algorithm 1 stated below, to compute an approximation of the mean drag coefficient $\bar{c}_D$ over a time interval $I = [T_0, T]$, defined by

$$\bar{c}_D = \frac{1}{|T - T_0|} \int_{T_0}^{T} c_D(t)\, dt, \tag{3.1}$$

where $c_D(t)$ is the drag coefficient at time t. In Figure 2 we show the computed values of $\bar{c}_D$ (for a time interval of length $40H$). The approximations of $\bar{c}_D$ approaches 1.45-1.5, a value that is well captured already using less than 10^5 mesh points.

We know of no experimental reference values of $\bar{c}_D$, but in [14] $\bar{c}_D$ is approximated computationally. The computational setup is similar to the one in this paper except the numerical method, a different length of the time interval, and that we in this paper use a channel of length $15H$, compared to a channel of length $10H$ in [14]. Using different

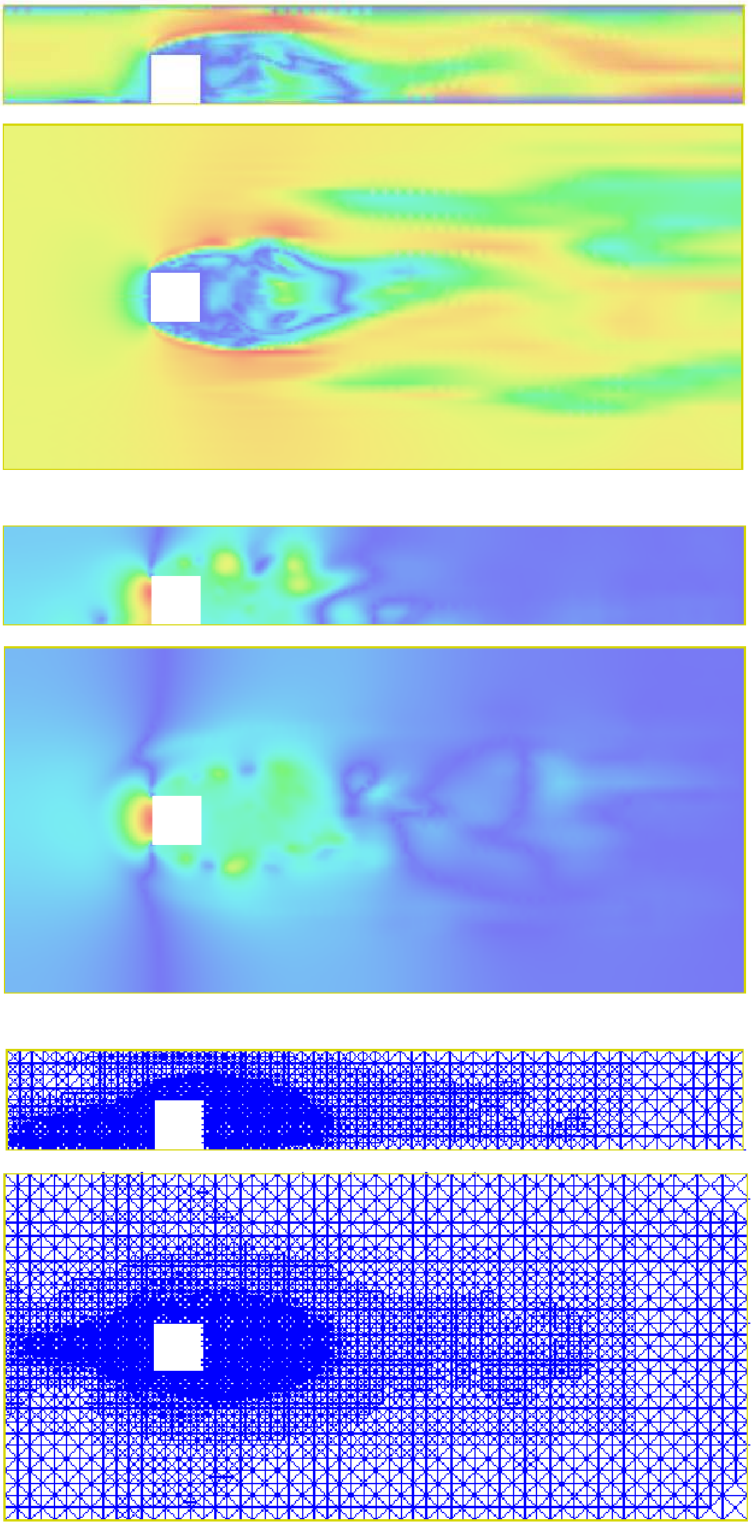


FIGURE 1. Velocity $|u|$ (upper), pressure $|p|$ (middle), and computational mesh (lower), after 13 adaptive mesh refinements at $z = 3.5H$ and $y = 0.5H$.

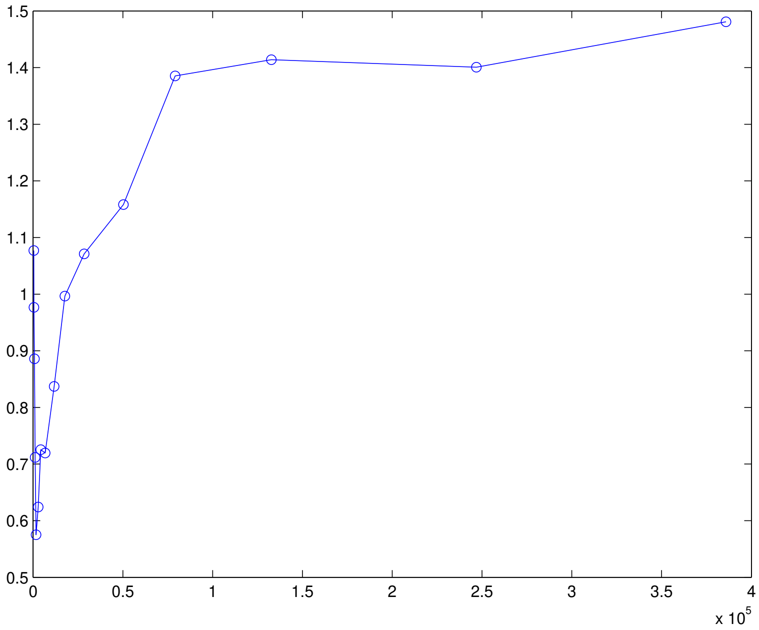


FIGURE 2. Mean drag coefficient $\bar{c}_D$ as a function of the number of mesh points.

meshes and subgrid models, approximations of $\bar{c}_D$ in the interval $[1.14, 1.24]$ are presented in [14].

4. Adaptive DNS/LES

To compute the drag force we solve 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 $\mathbb{R}^3$ with homogeneous Dirichlet boundary conditions, given by: Find (u, p) such that

$$
\begin{aligned}
\dot{u} + (u\cdot\nabla)u - \nu\Delta u + \nabla p &= f && \text{in } \Omega\times I,\\
\nabla\cdot u &= 0 && \text{in } \Omega\times I,\\
u &= 0 && \text{on } \partial\Omega\times I,\\
u(\cdot,0) &= u_0 && \text{in } \Omega,
\end{aligned}
\tag{4.1}
$$

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, u_0, $I = (0,T)$, is a given driving force, initial data and time interval, respectively. The quantity $\nu\Delta u - \nabla p$ represents the total fluid force, and may alternatively be expressed as

$$\nu\Delta u - \nabla p = \operatorname{div}\sigma(u,p), \tag{4.2}$$

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. We assume that (4.1) 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}.

4.1. **LES: The averaged Navier-Stokes equations.** In a turbulent flow we are typically not able to resolve all scales of motion computationally. We may instead aim at computing a *running average* u^h of u on a scale h, defined by

$$u^h(x,t) = \frac{1}{h^3}\int_{Q_h} u(x+y,t)\; dy, \tag{4.3}$$

where $h = h(x,t)$ is a parameter related to the local resolution of the problem and $Q_h = \{y \in \mathbb{R}^3 : |y_i| \leq h/2\}$. In the LES literature it is common to define the averaging operator through convolution by a certain filter function, and there is a multitude of filter functions being used. Though we only consider the case of the filter corresponding to (4.3) in this paper, the teqniques for a posteriori error estimation are general and apply to other filters, possibly with modifications for commutation errors associated with such filters.

By an extension of (u, p, u_0, f) to $\mathbb{R}^3$ by reflection for all $x \notin \overline{\Omega}$, the averaging operator (4.3) commutes with space and time differentiation. If we take the running average of the equations (4.1), corresponding to a LES, we obtain the following equations for u^h:

$$\begin{aligned} \dot{u}^h + (u^h\cdot\nabla)u^h - \nu\Delta u^h + \nabla p^h + F_h(u) = f^h \qquad & \text{in } \Omega\times I,\\ \nabla\cdot u^h = 0 \qquad & \text{in } \Omega\times I,\\ u^h = 0 \qquad & \text{on } \partial\Omega\times I,\\ u^h(\cdot,0) = u_0 \qquad & \text{in } \Omega, \end{aligned} \tag{4.4}$$

where we choose homogeneous Dirichlet boundary conditions for u^h, and $F_h(u) = \nabla\cdot\tau^h(u)$, where $\tau^h_{ij}(u) = (u_iu_j)^h - u^h_iu^h_j$ is the *Reynolds stress tensor*. The closure problem of LES is how 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 $\hat{\tau}^h(u^h)$.

A weak formulation of (4.4) reads: find $(u^h, p^h) \in L_2(I; [H^1_0(\Omega)]^3 \times L_2(\Omega))$, with $\dot{u}^h \in L_2(I; [L_2(\Omega)]^3)$ and $u^h(\cdot,0) = u^h_0$, such that

$$\begin{aligned} (\dot{u}^h + u^h\cdot\nabla u^h, v) + (2\nu\epsilon(u^h), \epsilon(v)) - (p^h, \nabla\cdot v)&\\ -(\tau^h(u), \nabla v) + (\nabla\cdot u^h, q) = (f^h, v),& \end{aligned} \tag{4.5}$$

for all $(v,q) \in L_2(I; [H^1_0(\Omega)]^3 \times L_2(\Omega))$, where we assume that $f^h \in L_2(I; [L_2(\Omega)]^3)$.

Here $L_2(\Omega)$ is the Hilbert space of Lebesgue square integrable functions on Ω, with scalar product $(\cdot,\cdot)$ and norm $\|\cdot\|$, and $H^s(\Omega)$ is the standard Hilbert space of functions in $L_2(\Omega)$ with also partial derivatives of order $\leq s$ in $L_2(\Omega)$. $H^s_w(\Omega)$ denotes the functions $v \in H^s(\Omega)$ that satisfies the Dirichlet boundary condition $v|_{\partial\Omega} = w$ (in the sense of traces), and in particular $H^s_0(\Omega)$ denotes the functions in $H^s(\Omega)$ that vanish on $\partial\Omega$. We let $\mathcal{C}(I;X)$ denote the space of all continuous functions $v : I \to X$ with $\max_{t\in I}\|v(t)\|_X < \infty$, where X denotes a Banach space with norm $\|\cdot\|_X$.

Assuming we have also Neumann boundary conditions, we denote Γ_D the part of the boundary $\partial\Omega$ where Dirichlet boundary conditions are specified, and $\Gamma_N = \partial\Omega\setminus\Gamma_D$ the part with Neumann boundary conditions. Now $H^s_w(\Omega)$ and $H^s_0(\Omega)$ denote the spaces of functions in $H^s(\Omega)$ that satisfies the Dirichlet boundary conditions on Γ_D.

4.2. **Discretization: cG(1)cG(1).** The cG(1)cG(1) method is a variant of the G^2 method [11, 10, 7] using the continuous Galerkin method cG(1) in time instead of a discontinuous Galerkin method. With cG(1) in time the trial functions are continuous piecewise linear and the test functions piecewise constant. cG(1) in space corresponds to both test functions and trial functions being continuous piecewise linear. 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 linear functions on a mesh $\mathcal{T}_n = \{K\}$ of mesh size $h_n(x)$ with W^n_w the functions $v \in W^n$ satisfying the Dirichlet boundary condition $v|_{\Gamma_D} = w$.

We now seek functions (U_h, P_h), continuous piecewise linear in space and time, and the cG(1)cG(1) method for the averaged Navier-Stokes equations (4.4), with homogeneous Dirichlet boundary conditions reads: For $n = 1, ..., N$, find $(U^n_h, P^n_h) \equiv (U_h(t_n), P_h(t_n))$ with $U^n_h \in V^n_0 \equiv [W^n_0]^3$ and $P^n_h \in W^n$, such that

$$\begin{aligned} &((U^n_h - U^{n-1}_h)k_n^{-1} + \hat{U}^n_h \cdot \nabla \hat{U}^n_h, v) + (2\nu\epsilon(\hat{U}^n_h), \epsilon(v)) - (P^n_h, \nabla \cdot v) \\ &+ (\nabla \cdot \hat{U}^n_h, q) + SD(\delta, \hat{U}^n_h, \hat{P}^n_h, v, q) = (f^h, v) \quad \forall (v, q) \in V^n_0 \times W^n, \end{aligned} \tag{4.6}$$

where $\hat{U}^n_h = \frac{1}{2}(U^n_h + U^{n-1}_h)$, with the stabilizing term

$$SD(\delta, \hat{U}^n_h, \hat{P}^n_h, v, q) \equiv \delta_1(\hat{U}^n_h \cdot \nabla \hat{U}^n_h + \nabla P^n_h - f^h, \hat{U}^n_h \cdot \nabla v + \nabla q) + \delta_2(\nabla \cdot \hat{U}^n_h, \nabla \cdot v),$$

and $\delta_1 = \frac{1}{2}(k_n^{-2} + |U|^2 h_n^{-2})^{-1/2}$ in the convection-dominated case $\nu < \hat{U}^n_h h_n$ and $\delta_1 = \kappa_1 h^2$ otherwise, $\delta_2 = \kappa_2 h$ if $\nu < \hat{U}^n_h h_n$ and $\delta_2 = \kappa_2 h^2$ otherwise, with κ_1 and κ_2 positive constants of unit size, and

$$(v, w) = \sum_{K \in \mathcal{T}_n} \int_K v \cdot w \, dx = \sum_{K \in \mathcal{T}_n} (v, w)_K,$$

$$(\epsilon(v), \epsilon(w)) = \sum_{i,j=1}^{3} (\epsilon_{ij}(v), \epsilon_{ij}(w)).$$

4.3. **Computation of the mean drag force.** We want to compute an approximation of the quantity

$$N(\sigma(u, p)) \equiv \frac{1}{|I|} \int_I \int_{\Gamma_0} \sum_{i,j=1}^{3} \sigma_{ij}(u, p) n_j \phi_i \; ds \; dt, \tag{4.7}$$

where (u, p) solves (4.1), ϕ is the trace on Γ_0 of a function in $H^1(\Omega)$, and $\Gamma_0 \subset \Gamma_D$ is a closed surface representing the boundary of a body immersed in the flow. If ϕ is a unit vector in the direction of the mean flow, (4.7) represents the mean of the drag force due to (u, p) on Γ_0 over a time interval I, and if ϕ is a unit vector in a direction perpendicular to the mean flow, (4.7) is the temporal mean of the lift force on Γ_0 due to (u, p) in that direction. With the idea of increasing the precision, see [6], we may use (4.1) and integration by parts to rewrite the surface integral in (4.7) as a volume integral, leading to the following expression for (4.7):

$$\begin{aligned} N(\sigma(u,p)) &= \frac{1}{|I|}\int_I (\dot{u} + u\cdot\nabla u - f^h, \varphi) - (p, \nabla\cdot\varphi) \\ &\quad + (2\nu\epsilon(u), \epsilon(\varphi)) + (\nabla\cdot u, \theta)\ dt, \end{aligned} \tag{4.8}$$

for any $\varphi \in L_2(I; [H^1_{\phi,0}(\Omega)]^3)$, where $H^1_{\phi,0}(\Omega) = \{v \in H^1(\Omega) : v|_{\Gamma_0} = \phi, v|_{\Gamma_1} = 0\}$, $\Gamma_1 = \Gamma_D \setminus \Gamma_0$, and $\theta \in L_2(I; L_2(\Omega))$. We note that due to (4.1), this representation does neither depend on the choice of θ, nor the particular extension φ of ϕ being used. We are thus led to approximate $N(\sigma(u,p))$ by the quantity

$$\begin{aligned} N^h(\sigma(U_h,P_h)) &= \frac{1}{|I|}\int_I (\dot{U}_h + U_h\cdot\nabla U_h - f^h, \Phi) - (P_h, \nabla\cdot\Phi) \\ &\quad + (2\nu\epsilon(U_h), \epsilon(\Phi)) + (\nabla\cdot U_h, \Theta)\ dt, \end{aligned} \tag{4.9}$$

where $(U_h,P_h) \in L_2(I; V_0^n \times W^n)$ and $(\Phi,\Theta) \in L_2(I; V^n_{\phi,0} \times W^n)$, with $V^n_{\phi,0} = \{v \in [W^n]^3 : v|_{\Gamma_0} = \phi, v|_{\Gamma_1} = 0\}$.

Below we will replace $N(\sigma(u,p))$ by $N(\sigma(u^h,p^h))$, with the motivation that a global mean value such as $N(\sigma(u,p))$ should be insensitive to taking local mean values.

4.4. **Adaptive algorithm.** An adaptive algorithm includes feed-back from computation to achieve the computational goal with minimal computational cost. In an adaptive finite element method this feed-back from computation relies on a posteriori error estimates. In Algorithm 1, an adaptive algorithm for the computation of the mean drag force $N(\sigma(u^h,p^h))$ is presented, which is based on a posteriori error estimates of the form

$$|N(\sigma(u^h,p^h)) - N^h(\sigma(U_h,P_h))| \leq \sum_{K\in\mathcal{T}_n^k} \mathcal{E}_K^k, \tag{4.10}$$

where $\mathcal{E}_K^k$ is an *error indicator* for element K. We have here chosen the computational mesh $\mathcal{T}_n^k$ to be constant in time for each iteration k in the adaptive algorithm, and we have also chosen the time step length k_n to be constant in time, namely

$$k_n = \min_{K\in\mathcal{T}_n^k} \text{diam}(K), \tag{4.11}$$

where diam(K) is the diameter of element K. In Figure 4 we plot a snapshot of the dual solution after 14 mesh refinements, and in Figure 3 we plot a posteriori error estimates of the error in $\bar{c}_D$ for the 15 first refinements (vs. the estimated reference value 1.48).

Algorithm 1 (Adaptive DNS/LES)**.** *Start at $k = 0$, then do*

(1) *compute approximation to the primal problem on $\mathcal{T}_n^k$*
(2) *compute approximation to the dual problem on $\mathcal{T}_n^k$*
(3) *if $\sum_{K\in\mathcal{T}_k} \mathcal{E}_K^k < TOL$ then STOP, else*
(4) *refine a fixed fraction of the elements in $\mathcal{T}_n^k$ with largest $\mathcal{E}_K^k \to \mathcal{T}_n^{k+1}$*
(5) *set $k = k+1$, then goto (1)*

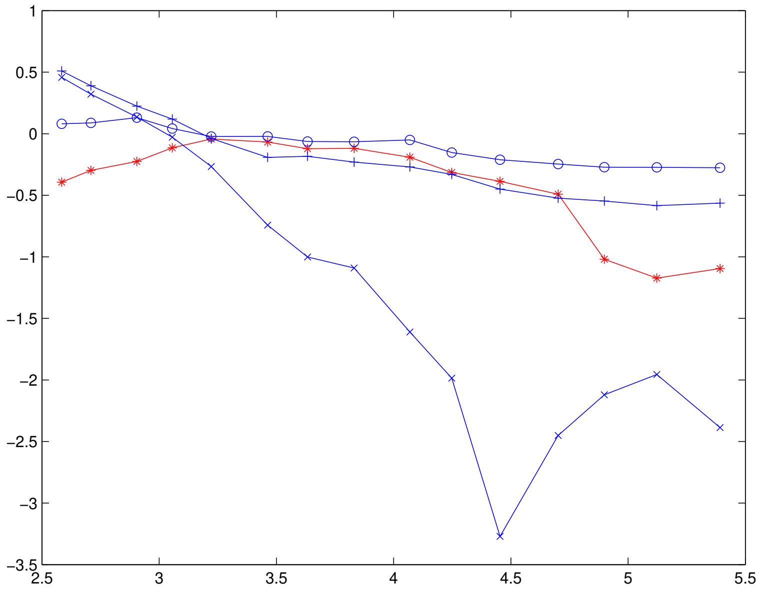


FIGURE 3. Various evaluations of the a posteriori error estimates in Theorem 2, with ('o' and '+') or without ('×') absolute values inside the summation over the elements, approximating the error ('*') in $\bar{c}_D$, versus number of mesh points.

4.5. **A posteriori error estimation.** Algorithm 1 is based on a posteriori error estimates of the form (4.10), which we derive by introducing the following linearized dual problem: Find $(\varphi,\theta)\in L_2(I;[H^1_{\psi_3}(\Omega)]^3\times L_2(\Omega))$ with $\dot\varphi\in L_2(I;[L_2(\Omega)]^3)$ and $\varphi(T)=0$, such that

$$\int_I -(v,\dot\varphi)+((u^h\cdot\nabla)v+(v\cdot\nabla)U_h,\varphi) + (2\nu\epsilon(v),\epsilon(\varphi))-(q,\nabla\cdot\varphi)+(\nabla\cdot v,\theta)\;dt=0, \tag{4.12}$$

for all $(v,q)\in L_2(I;[H^1_0(\Omega)]^3\times L_2(\Omega))$ with $v(0)=0$, given the data $\psi_3\in L_2(I;[L_2(\partial\Omega)]^3)$ defined below, and where $(\nabla U_h\cdot\varphi)_j=(U_h)_{,j}\cdot\varphi$.

Theorem 2. *If u^h solves (4.4), (U_h,P_h) solves (4.6), and (φ,θ) solves (4.12), then*

$$|N(\sigma(u^h,p^h))-N^h(\sigma(U_h,P_h))| = |\sum_{K\in\mathcal{T}_n}\mathcal{E}_K| \le \sum_{K\in\mathcal{T}_n}|\mathcal{E}_K|,$$

where $\mathcal{E}_K=e^K_D+e^K_M$ is an error indicator for element K, and

$$e^K_D=\frac{1}{|I|}\int_{I_n}((\dot U_h+U_h\cdot\nabla)U_h-f^h,\varphi-\Phi)_K-(P_h,\nabla\cdot(\varphi-\Phi))_K+(\nabla\cdot U_h,\theta-\Theta)_K$$
$$+\,(2\nu\epsilon(U_h),\nabla(\varphi-\Phi))_K+SD(\delta,U_h,P_h,\varphi-\Phi,\theta-\Theta)_K\;dt,$$

$$e^K_M=\frac{1}{|I|}\int_{I_n}(\tau^h(u),\nabla\varphi-\Phi)_K-SD(\delta,U_h,P_h,\varphi-\Phi,\theta-\Theta)_K\;dt,$$

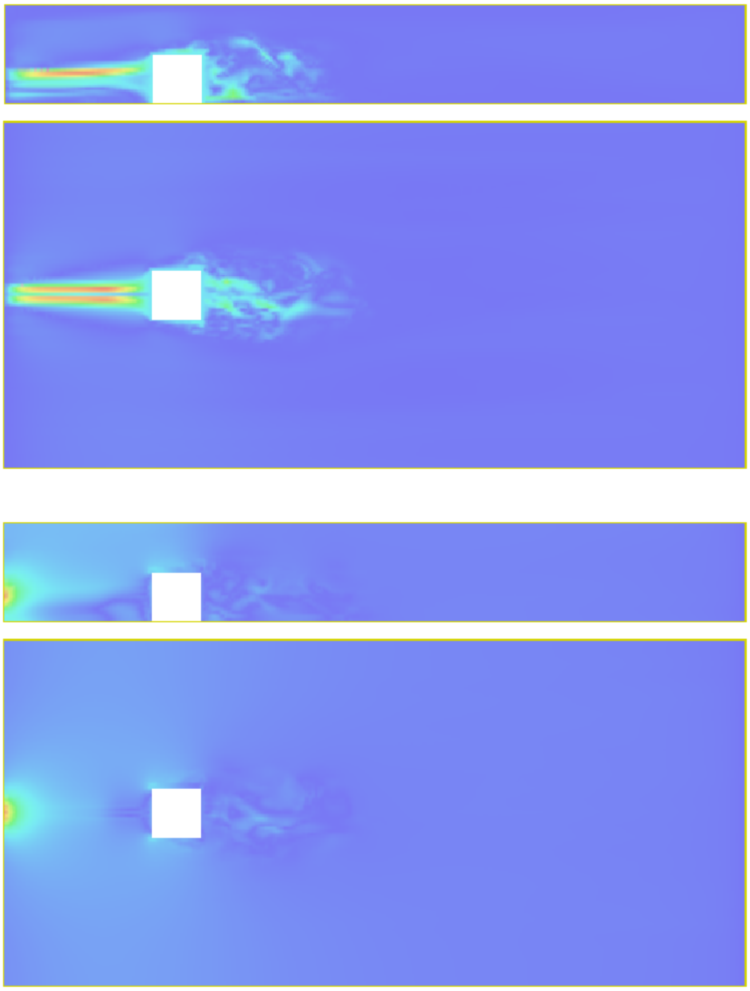


FIGURE 4. Dual velocity $|\varphi|$ (upper) and dual pressure $|\theta|$ (lower), after 13 adaptive mesh refinements at $z = 3.5H$ (upper) and $y = 0.5H$ (lower).

Proof. To derive a posteriori error estimates for $N(\sigma(u^h, p^h))$, the natural quantity to consider is the difference between (4.8) and (4.9), see [6, 9]. If we set $(\varphi, \theta) = (\Phi, \Theta) \in L_2(I; V^n_{\phi,0} \times W^n)$ in (4.8) and then subtract (4.9), we get

$$
\begin{aligned}
(4.13) \qquad & N(\sigma(u^h, p^h)) - N^h(\sigma(U_h, P_h)) \\
= \quad & \frac{1}{|I|}\int_I (\dot{u}^h + u^h \cdot \nabla u^h, \Phi) - (p^h, \nabla \cdot \Phi) + (2\nu\epsilon(u^h), \epsilon(\Phi)) + (\nabla \cdot u^h, \Theta) \\
& -((\dot{U}_h + U_h \cdot \nabla U_h, \Phi) - (P_h, \nabla \cdot \Phi) + (2\nu\epsilon(U_h), \epsilon(\Phi)) + (\nabla \cdot U_h, \Theta))\, dt.
\end{aligned}
$$

The dual problem (4.12) with data

$$
(4.14) \qquad \psi_3|_{\Gamma_1} = 0, \quad \psi_3|_{\Gamma_0} = \phi,
$$

and ϕ from (4.7), gives that

$$
\begin{aligned}
& \frac{1}{|I|}\int_I (\dot{u}^h + u^h\cdot\nabla u^h,\varphi) - (p^h,\nabla\cdot\varphi) + (2\nu\epsilon(u^h),\epsilon(\varphi)) + (\nabla\cdot u^h,\theta) \\
& -((\dot{U}_h + U_h\cdot\nabla U_h,\varphi) - (P_h,\nabla\cdot\varphi) + (2\nu\epsilon(U_h),\epsilon(\varphi)) + (\nabla\cdot U_h,\theta)\ dt \\
(4.15)\quad = \quad & \frac{1}{|I|}\int_I -(\dot{\varphi},e) + (u^h\cdot\nabla e + e\cdot\nabla U_h,\varphi) - (p^h - P_h,\nabla\cdot\varphi) \\
& +(2\nu\epsilon(e),\epsilon(\varphi)) + (\nabla\cdot e,\theta)\ dt = 0,
\end{aligned}
$$

using partial integration with $\varphi(T) = e(0) = 0$, where $e = u^h - U_h$, and that $(u^h\cdot\nabla)u^h - (U_h\cdot\nabla)U_h = (u^h\cdot\nabla)e + (e\cdot\nabla)U_h$. By (4.5), (4.13), and (4.15), we then have that

$$
\begin{aligned}
& N(\sigma(u^h,p^h)) - N^h(\sigma(U_h,P_h)) = \frac{1}{|I|}\int_I ((\dot{U}_h + U_h\cdot\nabla)U_h - f^h,\varphi-\Phi) \\
(4.16)\quad & -(P_h,\nabla\cdot(\varphi-\Phi)) + (\nabla\cdot U_h,\theta-\Theta) + (2\nu\epsilon(U_h),\nabla(\varphi-\Phi)) \\
& +SD(\delta,U_h,P_h,\varphi-\Phi,\theta-\Theta) - SD(\delta,U_h,P_h,\varphi-\Phi,\theta) - (\tau^h(u),\nabla(\varphi-\Theta))\ dt.
\end{aligned}
$$

□

From this *error representation formula* there are various possibilities to estimate the integrals in (4.16), see Figure 3. We notice that in the initial stages of the adaptive refinement, the estimators with the absolute values "inside" closely follow the actual error (vs. the estimated reference value 1.48 from the finest computation), while the estimator with absolute value "outside" seems to under estimate the error. For the fine part of the process the quality of the estimates is somewhat unclear since we do not know the true drag coefficient with sufficient precision.

Acknowledgments

The first author would like to aknowledge the support by DOE grant DE-FG02-88ER25053.

References

[1] *Aerodynamic loads database*, `http://www.nd.edu/~nathaz/database/`.
[2] R. Becker and R. Rannacher, *A feed-back approach to error control in adaptive finite element methods: Basic analysis and examples*, East-West J. Numer. Math., 4 (1996), pp. 237–264.
[3] ——, *A posteriori error estimation in finite element methods*, Acta Numerica, 10 (2001), pp. 1–103.
[4] K. Eriksson, D. Estep, P. Hansbo, and C. Johnson, *Introduction to adaptive method for differential equations*, Acta Numerica, 4 (1995), pp. 105–158.
[5] K. Eriksson and C. Johnson, *An adaptive finite element method for linear elliptic problems*, Math. Comp., 50 (1988), pp. 361–383.
[6] M. Giles, M. Larson, M. Levenstam, and E. Süli, *Adaptive error control for finite element approximations of the lift and drag coefficients in viscous flow*, Technical Report NA-76/06, Oxford University Computing Laboratory, (1997).
[7] J. Hoffman, *Duality based a posteriori error estimation in various norms and linear functionals for les*, submitted to SIAM Journal of Scientific Computing, (2002).
[8] ——, *Adaptive finite element methods for les: Computation of the drag coefficient in a turbulent flow around a surface mounted cube*, submitted to SIAM Journal of Scientific Computing, (2003).

[9] ———, *Computation of functionals in 3d incompressible flow for stationary benchmark problems using adaptive finite element methods*, submitted to Mathematical Models and Methods in Applied Sciences (M3AS), (2003).
[10] J. Hoffman and C. Johnson, *Adaptive finite element methods for incompressible fluid flow*, Error Estimation and Solution Adaptive Discretization in Computational Fluid Dynamics (Ed. T. J. Barth and H. Deconinck), Lecture Notes in Computational Science and Engineering, Springer-Verlag Publishing, Heidelberg, 2002.
[11] C. Johnson, *Adaptive Finite Element Methods for Conservation Laws*, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Springer Lecture Notes in Mathematics, Springer Verlag, 1998.
[12] C. Johnson and R. Rannacher, *On error control in CFD*, Int. Workshop Numerical Methods for the Navier-Stokes Equations (F.K. Hebeker et.al. eds.), Vol. 47 of *Notes Numer. Fluid Mech.*, Vierweg, Braunschweig, 1994, pp. 133–144.
[13] C. Johnson, R. Rannacher, and M. Boman, *Numerics and hydrodynamic stability: Toward error control in cfd*, SIAM J. Numer. Anal., 32 (1995), pp. 1058–1079.
[14] S. Krajnović and L. Davidson, *Large-eddy simulation of the flow around a bluff body*, AIAA Journal, 40 (2002), pp. 927–936.
[15] R. Martinuzzi and C. Tropea, *Flow around a surface mounted cube*, in selected papers of the 1992 Annual Conference on LDA Applications, Lisbon, 1993.
[16] ———, *The flow around a surface mounted, prismatic obstacles placed in a fully developed channel flow*, Trans. ASME, J. Fluids Eng., 115(85) (1993).

Chalmers Finite Element Center Preprints

2001–01 *A simple nonconforming bilinear element for the elasticity problem*
Peter Hansbo and Mats G. Larson

2001–02 *The $\mathcal{LL}^*$ finite element method and multigrid for the magnetostatic problem*
Rickard Bergström, Mats G. Larson, and Klas Samuelsson

2001–03 *The Fokker-Planck operator as an asymptotic limit in anisotropic media*
Mohammad Asadzadeh

2001–04 *A posteriori error estimation of functionals in elliptic problems: experiments*
Mats G. Larson and A. Jonas Niklasson

2001–05 *A note on energy conservation for Hamiltonian systems using continuous time finite elements*
Peter Hansbo

2001–06 *Stationary level set method for modelling sharp interfaces in groundwater flow*
Nahidh Sharif and Nils-Erik Wiberg

2001–07 *Integration methods for the calculation of the magnetostatic field due to coils*
Marzia Fontana

2001–08 *Adaptive finite element computation of 3D magnetostatic problems in potential formulation*
Marzia Fontana

2001–09 *Multi-adaptive galerkin methods for ODEs I: theory & algorithms*
Anders Logg

2001–10 *Multi-adaptive galerkin methods for ODEs II: applications*
Anders Logg

2001–11 *Energy norm a posteriori error estimation for discontinuous Galerkin methods*
Roland Becker, Peter Hansbo, and Mats G. Larson

2001–12 *Analysis of a family of discontinuous Galerkin methods for elliptic problems: the one dimensional case*
Mats G. Larson and A. Jonas Niklasson

2001–13 *Analysis of a nonsymmetric discontinuous Galerkin method for elliptic problems: stability and energy error estimates*
Mats G. Larson and A. Jonas Niklasson

2001–14 *A hybrid method for the wave equation*
Larisa Beilina, Klas Samuelsson and Krister Åhlander

2001–15 *A finite element method for domain decomposition with non-matching grids*
Roland Becker, Peter Hansbo and Rolf Stenberg

2001–16 *Application of stable FEM-FDTD hybrid to scattering problems*
Thomas Rylander and Anders Bondeson

2001–17 *Eddy current computations using adaptive grids and edge elements*
Y. Q. Liu, A. Bondeson, R. Bergström, C. Johnson, M. G. Larson, and K. Samuelsson

2001–18 *Adaptive finite element methods for incompressible fluid flow*
Johan Hoffman and Claes Johnson

2001–19 *Dynamic subgrid modeling for time dependent convection–diffusion–reaction equations with fractal solutions*
Johan Hoffman

2001–20 *Topics in adaptive computational methods for differential equations*
Claes Johnson, Johan Hoffman and Anders Logg

2001–21 *An unfitted finite element method for elliptic interface problems*
Anita Hansbo and Peter Hansbo

2001–22 *A P^2–continuous, P^1–discontinuous finite element method for the Mindlin-Reissner plate model*
Peter Hansbo and Mats G. Larson

2002–01 *Approximation of time derivatives for parabolic equations in Banach space: constant time steps*
Yubin Yan

2002–02 *Approximation of time derivatives for parabolic equations in Banach space: variable time steps*
Yubin Yan

2002–03 *Stability of explicit-implicit hybrid time-stepping schemes for Maxwell's equations*
Thomas Rylander and Anders Bondeson

2002–04 *A computational study of transition to turbulence in shear flow*
Johan Hoffman and Claes Johnson

2002–05 *Adaptive hybrid FEM/FDM methods for inverse scattering problems*
Larisa Beilina

2002–06 *DOLFIN - Dynamic Object oriented Library for FINite element computation*
Johan Hoffman and Anders Logg

2002–07 *Explicit time-stepping for stiff ODEs*
Kenneth Eriksson, Claes Johnson and Anders Logg

2002–08 *Adaptive finite element methods for turbulent flow*
Johan Hoffman

2002–09 *Adaptive multiscale computational modeling of complex incompressible fluid flow*
Johan Hoffman and Claes Johnson

2002–10 *Least-squares finite element methods with applications in electromagnetics*
Rickard Bergström

2002–11 *Discontinuous/continuous least-squares finite element methods for elliptic problems*
Rickard Bergström and Mats G. Larson

2002–12 *Discontinuous least-squares finite element methods for the Div-Curl problem*
Rickard Bergström and Mats G. Larson

2002–13 *Object oriented implementation of a general finite element code*
Rickard Bergström

2002–14 *On adaptive strategies and error control in fracture mechanics*
Per Heintz and Klas Samuelsson

2002–15 *A unified stabilized method for Stokes' and Darcy's equations*
Erik Burman and Peter Hansbo

2002–16 *A finite element method on composite grids based on Nitsche's method*
Anita Hansbo, Peter Hansbo and Mats G. Larson

2002–17 *Edge stabilization for Galerkin approximations of convection–diffusion problems*
Erik Burman and Peter Hansbo

2002–18 *Adaptive strategies and error control for computing material forces in fracture mechanics*
Per Heintz, Fredrik Larsson, Peter Hansbo and Kenneth Runesson

2002–19 *A variable diffusion method for mesh smoothing*
J. Hermansson and P. Hansbo

2003–01 *A hybrid method for elastic waves*
L.Beilina

2003–02 *Application of the local nonobtuse tetrahedral refinement techniques near Fichera-like corners*
L.Beilina, S.Korotov and M. Křížek

2003–03 *Nitsche's method for coupling non-matching meshes in fluid-structure vibration problems*
Peter Hansbo and Joakim Hermansson

2003–04 *Crouzeix–Raviart and Raviart–Thomas elements for acoustic fluid–structure interaction*
Joakim Hermansson

2003–05 *Smoothing properties and approximation of time derivatives in multistep backward difference methods for linear parabolic equations*
Yubin Yan

2003–06 *Postprocessing the finite element method for semilinear parabolic problems*
Yubin Yan

2003–07 *The finite element method for a linear stochastic parabolic partial differential equation driven by additive noise*
Yubin Yan

2003–08 *A finite element method for a nonlinear stochastic parabolic equation*
Yubin Yan

2003–09 *A finite element method for the simulation of strong and weak discontinuities in elasticity*
Anita Hansbo and Peter Hansbo

2003–10 *Generalized Green's functions and the effective domain of influence*
Donald Estep, Michael Holst, and Mats G. Larson

2003–11 *Adaptive finite element/difference method for inverse elastic scattering waves*
L.Beilina

2003–12 *A Lagrange multiplier method for the finite element solution of elliptic domain decomposition problems using non-matching meshes*
Peter Hansbo, Carlo Lovadina, Ilaria Perugia, and Giancarlo Sangalli

2003–13 *A reduced P^1–discontinuous Galerkin method*
R. Becker, E. Burman, P. Hansbo, and M. G. Larson

2003–14 *Nitsche's method combined with space–time finite elements for ALE fluid–structure interaction problems*
Peter Hansbo, Joakim Hermansson, and Thomas Svedberg

2003–15 *Stabilized Crouzeix–Raviart element for the Darcy-Stokes problem*
Erik Burman and Peter Hansbo

2003–16 *Edge stabilization for the generalized Stokes problem: a continuous interior penalty method*
Erik Burman and Peter Hansbo

2003–17 *A conservative flux for the continuous Galerkin method based on discontinuous enrichment*
Mats G. Larson and A. Jonas Niklasson

2003–18 *CAD–to–CAE integration through automated model simplification and adaptive modelling*
K.Y. Lee, M.A. Price, C.G. Armstrong, M.G. Larson, and K. Samuelsson

2003–19 *Multi-adaptive time integration*
Anders Logg

2003–20 *Adaptive computational methods for parabolic problems*
Kenneth Eriksson, Claes Johnson, and Anders Logg

2003–21 *The FEniCS project*
T. Dupont, J. Hoffman, C. Johnson, R. Kirby, M. Larson, A. Logg, and R. Scott

2003–22 *Adaptive finite element methods for LES: Computation of the mean drag coefficient in a turbulent flow around a surface mounted cube using adaptive mesh refinement*
Johan Hoffman

2003–23 *Adaptive DNS/LES: a New Agenda in CFD*
Johan Hoffman and Claes Johnson

These preprints can be obtained from

`www.phi.chalmers.se/preprints`