J. Fluid Mech. (2006), *vol.* 568, *pp.* 77–88.
doi:10.1017/S0022112006002679 Printed in the United Kingdom

Adaptive simulation of the subcritical flow past a sphere

By JOHAN HOFFMAN†
School of Computer Science and Communication, KTH, SE-10044 Stockholm, Sweden

(Received 13 April 2005 and in revised form 20 January 2006)

Adaptive DNS/LES (direct numerical simulation/large-eddy simulation) is used to compute the drag coefficient c_D for the flow past a sphere at Reynolds number $Re = 10^4$. Using less than 10^5 mesh points, c_D is computed to an accuracy of a few percent, corresponding to experimental precision, which is at least an order of magnitude cheaper than standard non-adaptive LES computations in the literature. Adaptive DNS/LES is a General Galerkin G2 method for turbulent flow, where a stabilized Galerkin finite element method is used to compute approximate solutions to the Navier–Stokes equations, with the mesh being adaptively refined until a stopping criterion is reached with respect to the error in a chosen output of interest, in this paper c_D. Both the stopping criterion and the mesh refinement strategy are based on *a posteriori* error estimates, in the form of a space–time integral of residuals multiplied by derivatives of the solution of an associated dual problem, linearized at the approximate solution, and with data coupling to the output of interest. There is no filtering of the equations, and thus no Reynolds stresses are introduced that need modelling. The stabilization in the numerical method is acting as a simple turbulence model.

1. Introduction

We consider the flow of an incompressible fluid past a sphere at a high Reynolds number. Characteristic of this flow is a laminar boundary layer that separates close to the equator to form a turbulent wake of approximately the same length as the diameter of the sphere. For very high Reynolds numbers the boundary layer undergoes transition to turbulence leading to a delayed separation and a much smaller wake, corresponding to a drastic decrease of the drag, the so-called *drag crisis*. In this paper we consider the subcritical flow with laminar separation of the boundary layer.

The Navier–Stokes equations (NSE) seem to be able to model both laminar and turbulent flow in a wide range of applications. The number of degrees of freedom needed to represent all the small scales in the turbulent flow in a direct numerical simulation (DNS) may be estimated to be of order Re^3 in space–time. In many applications of industrial importance the Reynolds number Re is larger than 10^6, and thus full resolution of all physical scales is impossible using the computers of today.

Not only is DNS for high Reynolds numbers very expensive; even if we were able to resolve all physical scales in a DNS we should not expect to be able to compute a pointwise accurate solution. Theoretically, such an ‘exact solution’ could exist in a computation completely free from noise and with zero numerical error, but the

† Present address: School of Computer Science and Communication, KTH, SE-10044 Stockholm, Sweden.

smallest perturbation in data, model, or numerics would result in a solution which locally could differ significantly from this ‘exact solution’. In addition, the value of such an ‘exact solution’ would be questionable from an application point of view, where perturbations always are present.

On the other hand, both experiments and computations indicate that certain mean values are less sensitive to perturbations than more local quantities. That is, in general we expect a smaller variation in mean value output than in pointwise output. It seems reasonable to aim for an accuracy in computations similar to the best possible accuracy in experiments. For the turbulent flow over a sphere, mean value outputs such as drag are experimentally determined to an accuracy of a few percent, and thus we aim for a similar accuracy in computations.

The traditional approach to get around the problems of DNS is to use some sort of turbulence modelling, where one seeks new equations satisfied by some average of the solutions to the NSE. These new equations are constructed by averaging, or filtering, NSE, introducing so-called Reynolds stresses, representing the influence of unresolved scales on resolved scales. In a Reynolds-averaged Navier–Stokes (RANS) model the filter corresponds to a global average or an ensemble average, whereas in a large eddy simulation (LES) the average is local in space and time, see Sagaut (2001) for an overview of LES. The Reynolds stresses are given in terms of the unfiltered velocity, and thus needs to be modelled in terms of the filtered velocity in a subgrid model, or turbulence model, which is referred to as the problem of closure. The closure problem is a main unsolved problem of turbulence research today, with the existing turbulence models being problem dependent and highly sensitive to the numerical method being used to solve the averaged equations.

An alternative approach to filtering is to seek functions that satisfy the NSE only in an approximate weak sense. Using stabilized Galerkin finite element methods, here referred to as General Galerkin G2 methods, we can construct such solutions with the corresponding NSE residuals being small in a weak norm. A G2 method is a combination of a Galerkin method, ensuring the residual to be small in average, and a weighted least-squares stabilization, corresponding to a certain strong control of the residual.

For a G2 solution, one can derive *a posteriori* error estimates for the error in a quantity of interest, or output. Such *a posteriori* error estimates take the form of a space–time integral of a residual times a dual weight, where the dual weight characterizes the sensitivity in a chosen output with respect computational errors. Within the same framework it is possible to study the sensitivity in output error with respect errors in data, but in this paper we focus on computational errors. The dual weight is obtained from computational approximation of an associated dual problem linearized at a G2 solution with data coupling to the chosen output. In particular, the *a posteriori* error estimates both control the numerical error from the Galerkin discretization in G2, and the modelling error from the stabilization in G2.

Based on the *a posteriori* error estimates we construct an algorithm for adaptive mesh refinement with respect to the error in the chosen output. For turbulent flow we also refer to this method as Adaptive DNS/LES, where part of the flow is being resolved in a DNS and part of the flow is being left under-resolved in an LES, with the stabilization in G2 acting as a type of turbulence model.

For an overview of adaptive finite element methods including references, we refer to the survey articles by Eriksson *et al.* (1995), Becker & Rannacher (2001) and Giles & Süli (2002). The extension of this framework to LES is investigated in Hoffman (2004). The generalization to Adaptive DNS/LES is first presented in Hoffman & Johnson

(2006) and Hoffman (2005), with applications to flow around a surface-mounted cube and a square cylinder.

For the bluff body problems considered in Hoffman & Johnson (2006) and Hoffman (2005), we find that using Adaptive DNS/LES we are able to compute mean value output, such as drag, using of order 10–100 times fewer mesh points in space than typical LES computations for the corresponding problems in the literature. This is a dramatic cut of the computational cost, and it is of great importance to further investigate the properties of Adaptive DNS/LES applied to basic benchmark problems of turbulence.

In this paper we use Adaptive DNS/LES to compute drag for the subcritical flow past a sphere at $Re = 10^4$, and again we find that Adaptive DNS/LES is significantly cheaper than corresponding non-adaptive LES computations in the literature.

First we recall details of Adaptive DNS/LES as a computational method for turbulence simulation, and then present results from computing the drag of a sphere at Reynolds number 10^4.

2. The Navier–Stokes equations

The incompressible Navier–Stokes equations expressing conservation of momentum and incompressibility of a unit-density Newtonian fluid with constant kinematic viscosity $\nu > 0$ enclosed in a volume Ω in $\mathbb{R}^3$ (where we assume that Ω is a polygonal domain) with homogeneous Dirichlet boundary conditions take the form: find $\hat{\boldsymbol{u}} = (\boldsymbol{u}, p)$ such that

$$\left.\begin{aligned} \dot{\boldsymbol{u}} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad & \text{in } \Omega \times I, \\ \nabla \cdot \boldsymbol{u} = 0 \qquad & \text{in } \Omega \times I, \\ \boldsymbol{u} = 0 \qquad & \text{on } \partial\Omega \times I, \\ \boldsymbol{u}(\cdot, 0) = \boldsymbol{u}_0 \qquad & \text{in } \Omega, \end{aligned}\right\} \tag{2.1}$$

where $\boldsymbol{u}(x,t) = (u_i(x,t))$ is the velocity vector and $p(x,t)$ the pressure of the fluid at (x,t), and $\boldsymbol{f}$, u_0, $I = (0,T)$, are a given driving force, initial data and time interval, respectively. The quantity $\nu \Delta \boldsymbol{u} - \nabla p$ represents the total fluid force, and may alternatively be expressed as

$$\nu \Delta \boldsymbol{u} - \nabla p = \operatorname{div} \boldsymbol{\sigma}(\boldsymbol{u}, p), \tag{2.2}$$

where $\boldsymbol{\sigma}(\boldsymbol{u}, p) = (\sigma_{ij}(u,p))$ is the stress tensor, with components $\sigma_{ij}(u,p) = 2\nu\epsilon_{ij}(\boldsymbol{u}) - p\delta_{ij}$, composed of the stress deviatoric $2\nu\epsilon_{ij}(\boldsymbol{u})$ with zero trace and an isotropic pressure: here $\epsilon_{ij}(\boldsymbol{u}) = (u_{i,j} + u_{j,i})/2$ 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 typically assume that (2.1) 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}.

3. Discretization: cG(1)cG(1)

The cG(1)cG(1) method is a General Galerkin G2 method, see Hoffman (2005), using the continuous Galerkin method cG(1) in space and time. 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 $0 = t_0 < t_1 < \ldots < t_N = T$ 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 $\mathscr{T}_n = \{K\}$ of mesh size $h_n(x)$, with W^n_w the functions $v \in W^n$ satisfying the Dirichlet boundary condition $v|_{\partial\Omega} = w$.

We seek $\hat{\boldsymbol{U}} = (\boldsymbol{U}, P)$, continuous piecewise linear in space and time, and the cG(1)cG(1) method for the Navier–Stokes equations (2.1) with homogeneous Dirichlet boundary conditions reads: for $n = 1, \ldots, N$, find $(\boldsymbol{U}^n, P^n) \equiv (\boldsymbol{U}(t_n), P(t_n))$ with $\boldsymbol{U}^n \in V^n_0 \equiv [W^n_0]^3$ and $P^n \in W^n$, such that

$$
\begin{aligned}
((\boldsymbol{U}^n - \boldsymbol{U}^{n-1})k_n^{-1} + \bar{\boldsymbol{U}}^n \cdot \nabla \bar{\boldsymbol{U}}^n, \boldsymbol{v}) + (2\nu\epsilon(\bar{\boldsymbol{U}}^n), \epsilon(\boldsymbol{v})) - (P^n, \nabla \cdot \boldsymbol{v}) \\
+ (\nabla \cdot \bar{\boldsymbol{U}}^n, q) + SD_\delta(\bar{\boldsymbol{U}}^n, P^n; \boldsymbol{v}, q) = (\boldsymbol{f}, \boldsymbol{v}) \quad \forall \hat{v} = (\boldsymbol{v}, q) \in V^n_0 \times W^n,
\end{aligned}
\tag{3.1}
$$

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

$$
SD_\delta(\bar{\boldsymbol{U}}^n, P^n; \boldsymbol{v}, q) \equiv (\delta_1(\bar{\boldsymbol{U}}^n \cdot \nabla \bar{\boldsymbol{U}}^n + \nabla P^n - \boldsymbol{f}), \bar{\boldsymbol{U}}^n \cdot \nabla \boldsymbol{v} + \nabla q) + (\delta_2 \nabla \cdot \bar{\boldsymbol{U}}^n, \nabla \cdot \boldsymbol{v}); \tag{3.2}
$$

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

$$
\left.
\begin{aligned}
(\boldsymbol{v}, \boldsymbol{w}) &= \sum_{K \in \mathscr{T}_\backslash} \int_K \boldsymbol{v} \cdot \boldsymbol{w} \, \mathrm{d}x, \\
(\boldsymbol{\epsilon}(\boldsymbol{v}), \boldsymbol{\epsilon}(\boldsymbol{w})) &= \sum_{i,j=1}^{3} (\epsilon_{ij}(\boldsymbol{v}), \epsilon_{ij}(\boldsymbol{w})).
\end{aligned}
\right\}
\tag{3.3}
$$

4. Computation of drag

Using partial integration, the mean value in time of the drag of a body may be expressed as (Giles *et al.* 1997)

$$
N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}})) = \frac{1}{|I|} \int_I (\dot{\boldsymbol{u}} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{f}, \boldsymbol{\Phi}) - (p, \nabla \cdot \boldsymbol{\Phi}) + (2\nu\boldsymbol{\epsilon}(\boldsymbol{u}), \boldsymbol{\epsilon}(\boldsymbol{\Phi})) + (\nabla \cdot \boldsymbol{u}, \Theta) \, \mathrm{d}t, \tag{4.1}
$$

where $\boldsymbol{\Phi}$ is a function defined in the fluid volume Ω, and is equal to a unit vector in the direction of the flow on Γ_0, the surface of the body in contact with the fluid, and zero on the remaining part of the boundary of the fluid volume $\Gamma_1 = \partial\Omega \backslash \Gamma_0$. The representation (4.1) is independent of Θ, and the particular extension of $\boldsymbol{\Phi}$ away from the boundary. Here $\hat{\boldsymbol{u}} = (\boldsymbol{u}, p)$ is a solution to (2.1) in the fluid volume Ω surrounding the body (using suitable boundary conditions as specified below) defining the target output $N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}}))$, with sufficient regularity for (4.1) to be well defined.

We compute an approximation of the drag $N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}}))$ from a cG(1)cG(1) solution $\hat{\boldsymbol{U}} = (\boldsymbol{U}, P)$, using the formula

$$
\begin{aligned}
N^h(\sigma(\hat{\boldsymbol{U}})) = \frac{1}{|I|} \int_I (\dot{\boldsymbol{U}} + \boldsymbol{U} \cdot \nabla \boldsymbol{U} - \boldsymbol{f}, \boldsymbol{\Phi}) - (P, \nabla \cdot \boldsymbol{\Phi}) \\
+ (2\nu\boldsymbol{\epsilon}(\boldsymbol{U}), \boldsymbol{\epsilon}(\boldsymbol{\Phi})) + (\nabla \cdot \boldsymbol{U}, \Theta) + SD_\delta(\boldsymbol{U}, P; \boldsymbol{\Phi}, \Theta) \, \mathrm{d}t,
\end{aligned}
\tag{4.2}
$$

where now $\boldsymbol{\Phi}$ and Θ are finite element functions (with, as before, $\Phi = \phi$ on Γ_0 and $\Phi = 0$ on Γ_1), and where $\dot{\boldsymbol{U}} = (\boldsymbol{U}^n - \boldsymbol{U}^{n-1})/k_n$ on I_n. We note the presence of the stabilizing term SD_δ in (4.2), compared to (4.1), which is added in order to obtain the independence of $N^h(\boldsymbol{\sigma}(\hat{\boldsymbol{U}}))$ from the choice of $(\boldsymbol{\Phi}, \Theta)$, given by (3.1).

We define the drag coefficient c_D as a global average of a normalized drag force on the sphere from the flow, and we seek to approximate c_D by $\bar{c}_D$, a normalized drag force averaged over a finite time interval I in fully developed flow, defined by

$$\bar{c}_D \equiv \frac{1}{\frac{1}{2}U_\infty^2 A} \times N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}})), \tag{4.3}$$

where $U_\infty = 1$ is the free-stream velocity, $A = 0.25 \times \pi D^2$ is the sphere section area facing the mean flow, with D the diameter of the sphere, and $N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}}))$ is defined by (4.1). In computations we approximate $\bar{c}_D$ by $\bar{c}_D^h$, defined by

$$\bar{c}_D^h = \frac{1}{\frac{1}{2}U_\infty^2 A} \times N^h(\boldsymbol{\sigma}(\hat{\boldsymbol{U}})), \tag{4.4}$$

with $N^h(\boldsymbol{\sigma}(\hat{\boldsymbol{U}}))$ being defined by (4.2). Below we present an *a posteriori* error estimate (5.2) for the error $|N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}})) - N^h(\boldsymbol{\sigma}(\hat{\boldsymbol{U}}))|$. A estimate of the error $|\bar{c}_D - \bar{c}_D^h|$ is obtained by scaling (5.2).

5. Adaptive DNS/LES

Adaptive DNS/LES may be thought of as an algorithm for solving the minimization problem: minimize the number of degrees of freedom, under the contraint that $|\bar{c}_D - \bar{c}_D^h| < TOL$, where TOL is a given tolerance typically of the same size as the experimental precision in c_D. The *a posteriori* error estimate underlying Adaptive DNS/LES is based on duality. We introduce the following dual problem: find $\hat{\boldsymbol{\varphi}} = (\boldsymbol{\varphi}, \theta)$ with $\boldsymbol{\varphi} = \boldsymbol{\Phi}$ on Γ_0 and $\varphi = 0$ on Γ_1, such that

$$\left.\begin{array}{rl} -\dot{\boldsymbol{\varphi}} - (\boldsymbol{u}\cdot\nabla)\boldsymbol{\varphi} + \nabla\boldsymbol{U}\cdot\boldsymbol{\varphi} - \nu\Delta\boldsymbol{\varphi} + \nabla\theta = 0 & \text{in } \Omega\times I, \\ \nabla\cdot\boldsymbol{\varphi} = 0 & \text{in } \Omega\times I, \\ \boldsymbol{\varphi}(\cdot, T) = 0 & \text{in } \Omega, \end{array}\right\} \tag{5.1}$$

where $(\nabla\boldsymbol{U}\cdot\boldsymbol{\varphi})_j = (\boldsymbol{U})_{,j}\cdot\boldsymbol{\varphi}$.

Replacing the exact dual solution $\hat{\boldsymbol{\varphi}}$ by a computed approximation $\hat{\boldsymbol{\varphi}}_h = (\boldsymbol{\varphi}_h, \theta_h)$, we are led to the following *a posteriori* output error estimate, see Hoffman & Johnson (2006) and Hoffman (2005), and assuming sufficient regularity of $\hat{\boldsymbol{\varphi}}_h$:

$$|N(\boldsymbol{\sigma}(\hat{\boldsymbol{u}})) - N^h(\boldsymbol{\sigma}(\hat{\boldsymbol{U}}))| \approx \left|\sum_{K\in\mathscr{T}_h} \mathscr{E}_{K,h}\right|, \tag{5.2}$$

where $\mathscr{E}_{K,h} = e_{D,h}^K + e_{M,h}^K$ is an error indicator for element K in the mesh $\mathscr{T}_h$, and

$$\begin{aligned} e_{D,h}^K = {} & \frac{1}{|I|}\int_I ((|R_1(\boldsymbol{U},P)|_K + |R_2(\boldsymbol{U},P)|_K)\cdot(C_h h^2|D^2\boldsymbol{\varphi}_h|_K + C_k k|\dot{\boldsymbol{\varphi}}_h|_K) \\ & + \|R_3(\boldsymbol{U})\|_K\cdot(C_h h^2\|D^2\theta_h\|_K + C_k k\|\dot{\theta}_h\|_K))\,\mathrm{d}t, \end{aligned} \tag{5.3}$$

$$e_{M,h}^K = \frac{1}{|I|}\int_I SD_\delta(U,P;\boldsymbol{\varphi}_h,\theta_h)_K\,\mathrm{d}t, \tag{5.4}$$

where we may view $e_{D,h}^K$ as an error contribution from the Galerkin part of the cG(1)cG(1) discretization, and $e_{M,h}^K$ a contribution from the stabilization in cG(1)cG(1), on element K, and k and h are the time step and the local mesh size, respectively. The residuals R_i are defined by

$$\left.\begin{array}{l} R_1(\boldsymbol{U},P) = \dot{\boldsymbol{U}} + \boldsymbol{U}\cdot\nabla\boldsymbol{U} + \nabla P - \boldsymbol{f} - \nu\Delta\boldsymbol{U}, \\ R_2(\boldsymbol{U},P) = \nu D_2(\boldsymbol{U}), \\ R_3(\boldsymbol{U},P) = \nabla\cdot\boldsymbol{U}, \end{array}\right\} \tag{5.5}$$

with

$$D_2(\boldsymbol{U})(x,t) = \max_{y\in\partial K}(h_n(x))^{-1}\left|\left[\frac{\partial \boldsymbol{U}}{\partial n}(y,t)\right]\right| \tag{5.6}$$

for $x \in K$, with $[\cdot]$ the jump across the element edge ∂K. D^2 denotes second-order spatial derivatives, and we write $|\boldsymbol{w}|_K \equiv (\|w_1\|_K, \|w_2\|_K, \|w_3\|_K)$, with $\|\boldsymbol{w}\|_K = (\boldsymbol{w}, \boldsymbol{w})_K^{1/2}$, and let the dot denote the scalar product in $\mathbb{R}^3$.

In the computations we drop the $R_2(\boldsymbol{U}, P)$-term since $\nu \ll h$, and we use $C_k = 1/2$ and $C_h = 1/8$ as constant approximations of the interpolation constants in (5.2), motivated by simple analysis on a reference element. Non-Dirichlet boundary conditions, such as slip conditions at lateral boundaries and transparant outflow conditions (see Hoffman 2005), introduce additional boundary terms in the *a posteriori* error estimate (5.2). But since the dual solution for the problem in this paper is small at such non-Dirichlet boundaries, we neglect the corresponding boundary terms in (5.2).

The dual problem (5.1) is a linear convection–diffusion–reaction problem where the convection acts backward in time and in the opposite direction to the exact flow velocity $\boldsymbol{u}$, which in computations is approximated by an approximate solution $\boldsymbol{U}$. The coefficient $\nabla \boldsymbol{U}$ of the reaction term is locally large in turbulent regions, and thus potentially generates rapid exponential growth. However, $\nabla \boldsymbol{U}$ is fluctuating, and the net effect of the reaction term in this paper turns out to generate slower growth, as we learn from computing approximations of the dual solution. We had the same experience when computing dual solutions related to mean value output in other turbulent flow problems, see Hoffman & Johnson (2006) and Hoffman (2005), where we also find that the dual solution is stable with respect to perturbations from computational errors and linearization errors from linearizing at the approximate convection velocity $\boldsymbol{U}$ instead of the exact velocity $\boldsymbol{u}$.

6. Computational model

In this paper, we keep the space mesh $\mathscr{T}_h$ and time step k constant in time, with the time step being equal to the smallest element diameter in the space mesh, and we use an algorithm for adaptive mesh refinement in space, based on the *a posteriori* error estimate (5.2), of the form:

ALGORITHM 1 (ADAPTIVE DNS/LES). *Given an initial coarse computational space mesh $\mathscr{T}_h^0$, start at $k=0$, then do*

(1) *Compute an approximation of the primal problem using $\mathscr{T}_h^k$.*
(2) *Compute an approximation of the dual problem using $\mathscr{T}_h^k$.*
(3) *If $|\sum_{K\in\mathscr{T}_h^k}\mathscr{E}_{K,h}^k| < TOL$ then STOP, else:*
(4) *Refine a fraction of the elements in $\mathscr{T}_h^k$ with largest $\mathscr{E}_{K,h}^k \rightarrow \mathscr{T}_h^{k+1}$.*
(5) *Set $k=k+1$, then goto* (1).

We consider the flow around a sphere at Reynolds number $Re = 10^4$, based on the unit inflow velocity in the x_1 streamwise direction and the sphere diameter $D = 0.1$. The sphere is centred at $(5.5D, 7.5D, 7.5D)$ in a channel of dimension $10D \times 15D \times 15D$. We use no-slip boundary conditions on the sphere, slip boundary conditions on the lateral walls, and a transparent (or do nothing) outflow boundary condition, see Hoffman (2005), at the end of the channel.

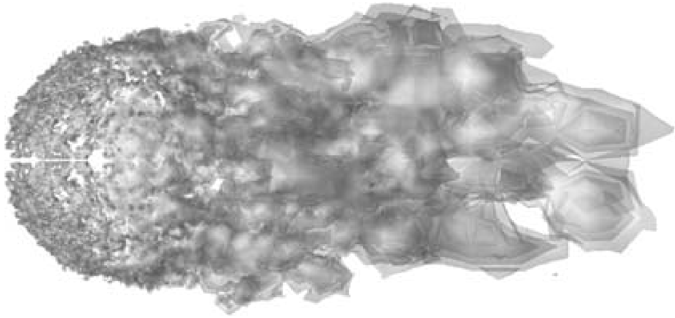

FIGURE 1. Snapshot of the out-of-plane vorticity isosurfaces for the solution on the finest computational mesh with 91 095 mesh points.

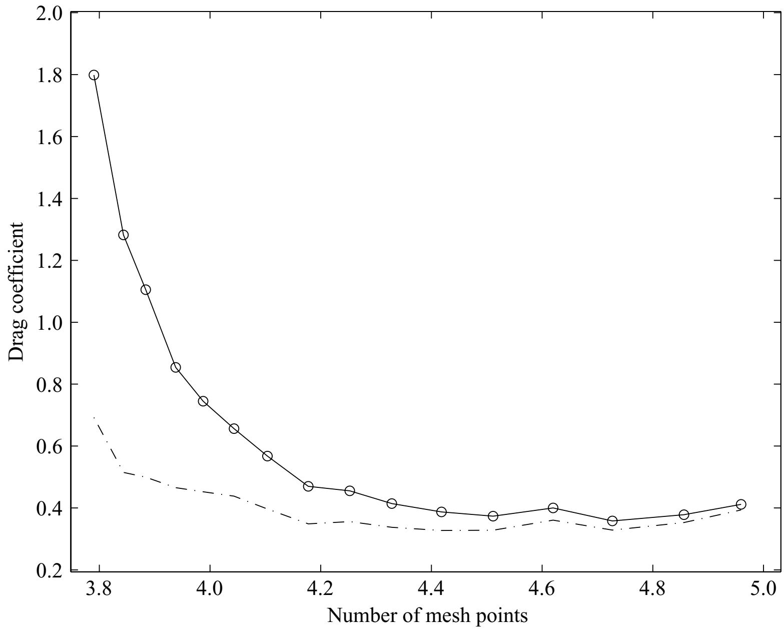

FIGURE 2. $\bar{c}_D^h$ with (—) and without (···) the contribution from the stabilizing term in (4.2) averaged over a time interval of length $25U_\infty/D$ plotted against $\log_{10}$ of the number of mesh points in the corresponding computational mesh.

7. Computational results

We use Adaptive DNS/LES to compute approximations $\bar{c}_D^h$ of the drag coefficient c_D from (4.4), on a tetrahedral computational mesh. We illustate the character of the solution in figure 1, and in figure 2 we plot $\bar{c}_D^h$ as we refine approximately 5% of the elements in each iteration of the adaptive algorithm. Experimental reference values for this problem are presented as $c_D \approx 0.40$ to an accuracy of a few percent, see Schlichting (1955) and Constantinescu & Squires (2004), and in figure 2 we find that when using less than 30 000 mesh points the approximation $\bar{c}_D^h$ is within the experimental tolerance.

In figure 3 we plot $\bar{c}_D^h$ as a function of time for the solution on the finest computational mesh with 91 095 mesh points. Here we also plot approximations of the two

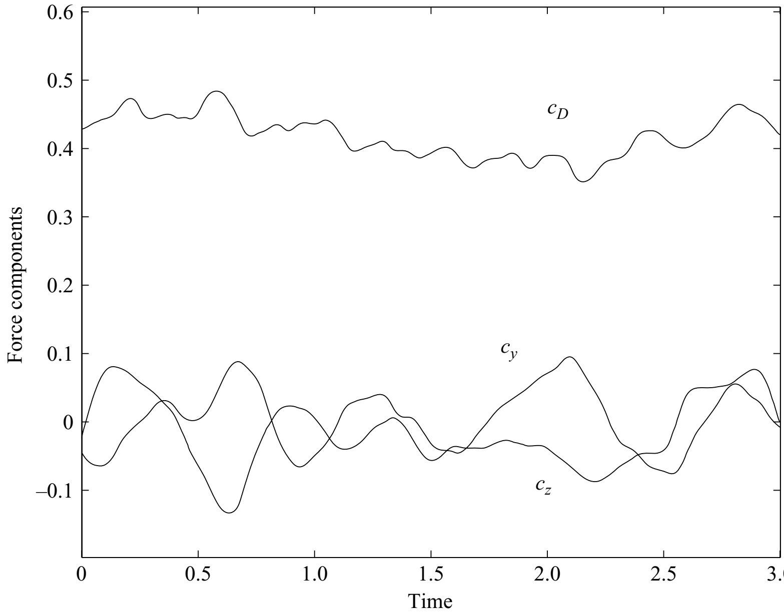

FIGURE 3. Drag coefficient $\bar{c}_D^h$ and transversal force coefficients $\bar{c}_y^h$ and $\bar{c}_z^h$ as functions of time t.

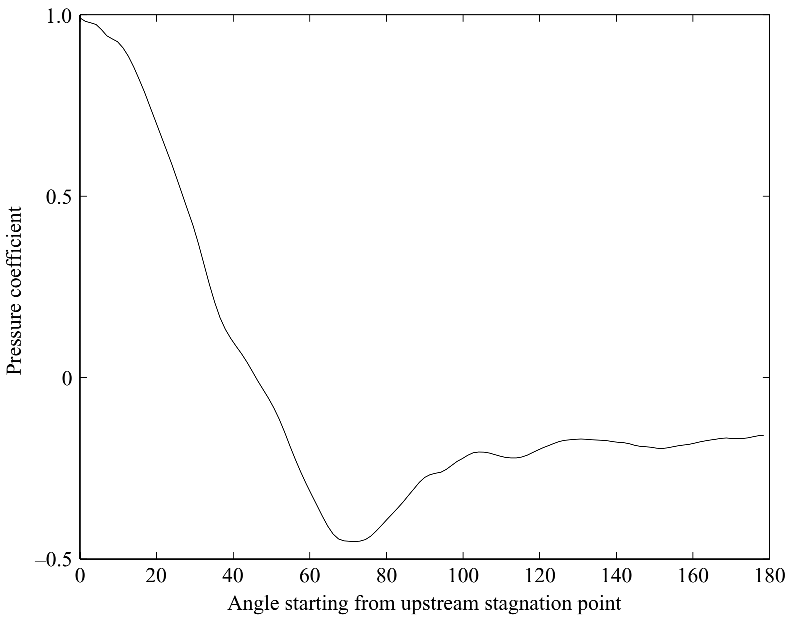

FIGURE 4. Pressure coefficient c_p averaged over a time interval of length $25U_\infty/D$, plotted against an angle starting from zero at the upstream stagnation point.

other components of the normalized force $\bar{c}_y^h$ and $\bar{c}_z^h$, and in figure 4 we plot the pressure coefficient c_p.

The adaptive algorithm is designed to compute the correct drag using a minimum number of degrees of freedom. Starting from a coarse tetrahedral mesh with 6168

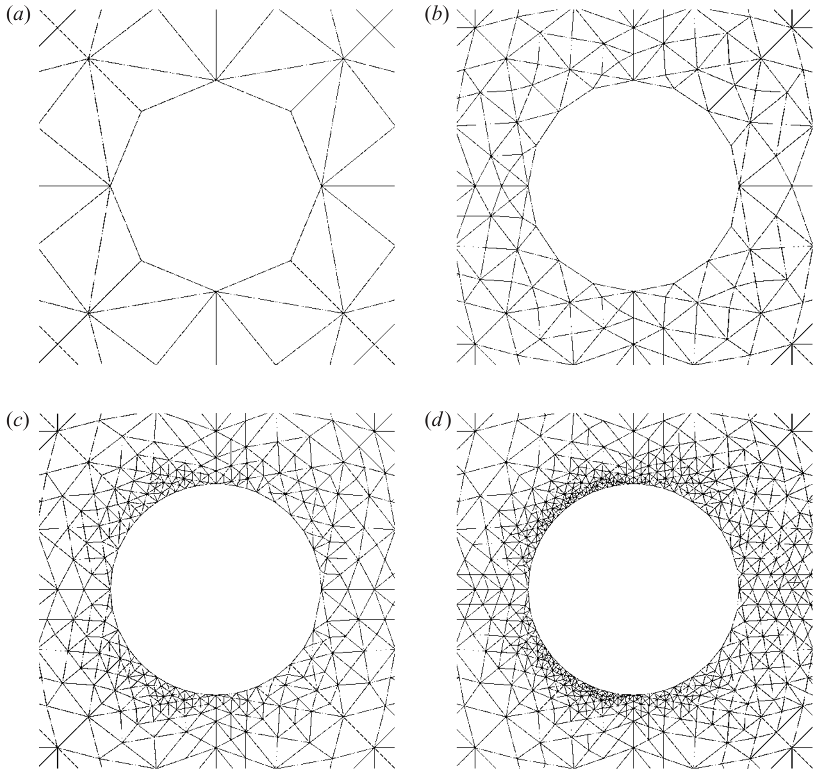

FIGURE 5. Computational mesh without refinement (*a*), and after 5 (*b*), 10 (*c*), and 15 (*d*) adaptive mesh refinements.

nodes, the mesh is adaptively refined with respect to the error in c_D (or rather $\bar{c}_D$). As the mesh is refined it is modified to fit the surface of the sphere, and the computational domain thus changes in each iteration of the adaptive algorithm, see figure 5. We find that the adaptive algorithm converges to a $\bar{c}_D^h$ corresponding to the exact geometry of the sphere.

Since the adaptive algorithm is designed to minimize computational cost for the computation of drag, the resulting computational mesh in figure 6 is optimized for the approximation of drag. We note that the mesh refinement is concentrated in boundary layers and the turbulent wake. In particular we note that the mesh is finest at the boundary layer upstream separation, so as to be able to capture the correct separation points.

Unneccessary refinement is avoided in parts of the domain not critical for the approximation of drag. For example, the mesh downstream of the wake is kept very coarse. If we were interested in an accurate approximation of the flow field further downstream we would have to change the quantity of interest in the adaptive algorithm.

The dual solution (or rather dervatives thereof) acts as a weight function for the residual in the *a posteriori* error estimate underlying the mesh refinement criterion for the adaptive algorithm. In figure 7 we plot snapshots of a dual solution for the computation of drag, and the corresponding primal solution at which the dual problem is linearized.

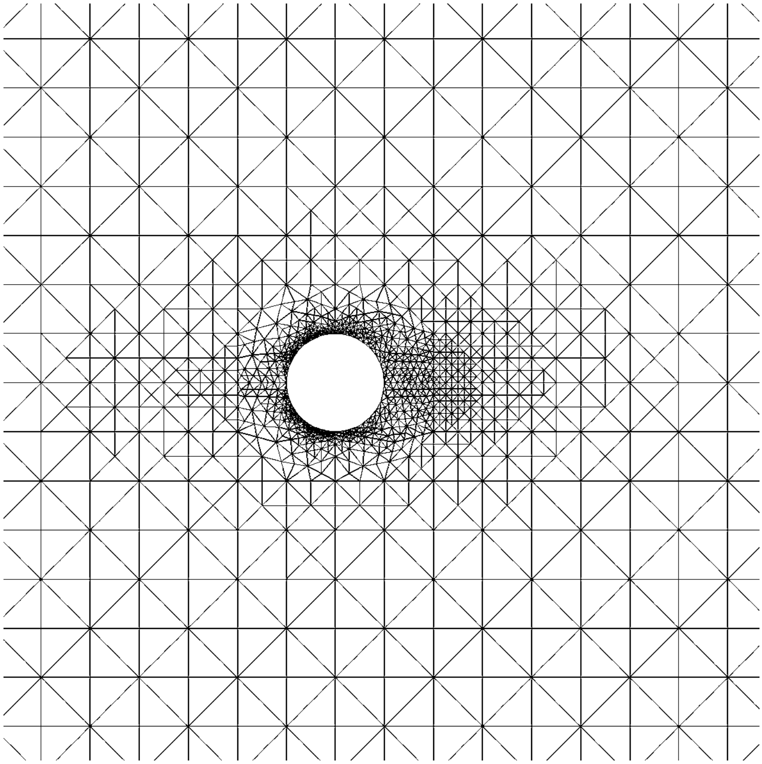

FIGURE 6. Computational mesh after 15 mesh refinements.

Refinements	Nodes	$\bar{c}_D^h$	$\bar{c}_y^h$	$\bar{c}_z^h$	St
15	91 095	0.41	0.0086	−0.025	0.20
14	71 814	0.38	−0.066	−0.051	0.20
13	53 322	0.36	−0.039	0.011	0.19
12	41 677	0.40	−0.030	0.037	0.17
11	32 450	0.37	0.029	0.025	0.18
10	26 190	0.39	−0.030	−0.029	0.15

TABLE 1. Mean value output averaged over a time interval of length $25U_\infty/D$.

The adaptive algorithm is constructed for approximation of drag, but it may be interesting also to measure other output from the resulting solutions. In table 1 we display mean value output for a set of meshes. Using less than 30 000 nodes we capture the experimental reference value $c_D \approx 0.40$, and for the finer meshes we also obtain good approximations of the other force components, which are close to zero, and the Strouhal number $St \approx 0.20$. We note that the length of the time averaging interval $25U_\infty/D$ is rather short, and on increasing the length we would expect even more stable mean value output.

Compared to typical LES computations, see e.g. Constantinescu & Squires (2004), with *ad hoc* mesh refinement, using $5.8 \times 10^5 - 1.2 \times 10^6$ mesh points, adaptive DNS/LES is about a factor 10–40 times cheaper in terms of mesh points.

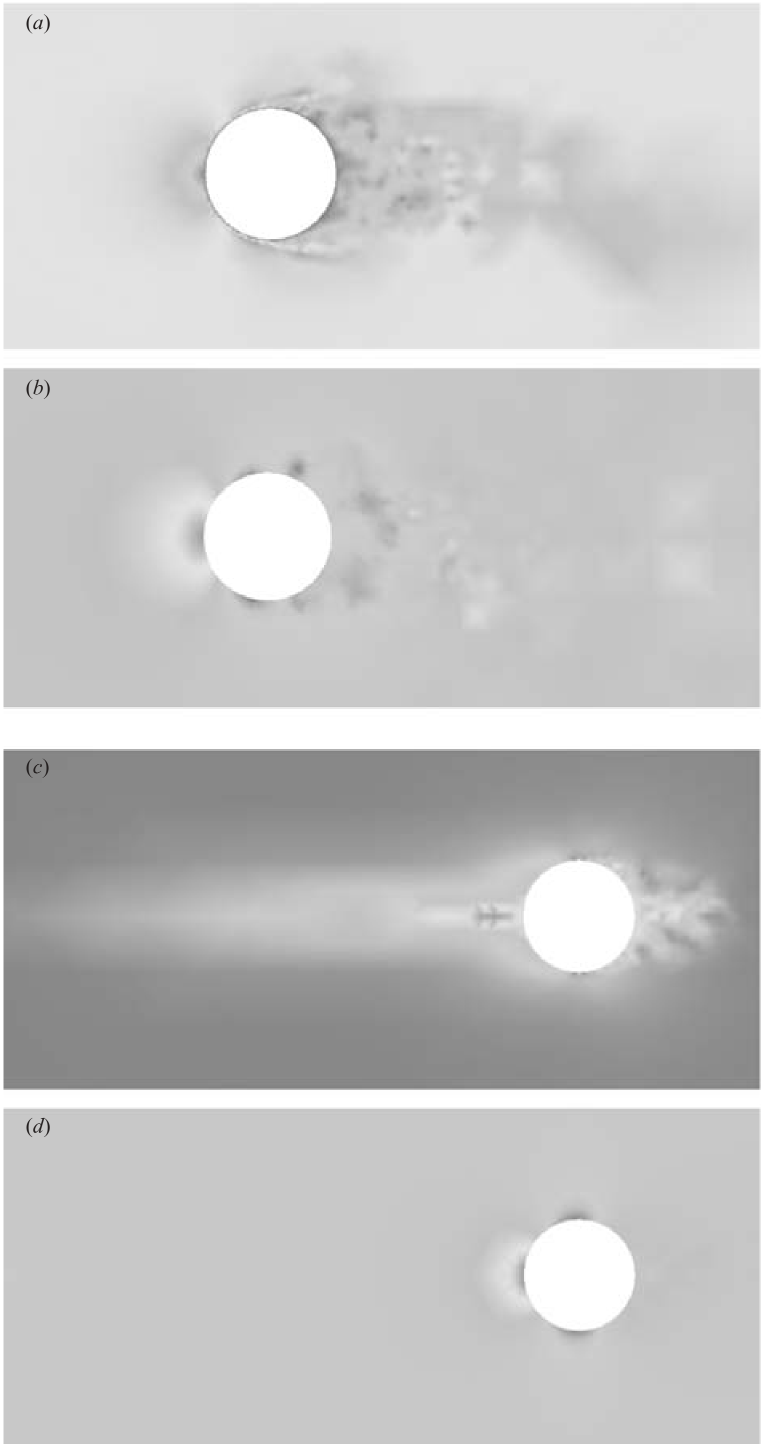

FIGURE 7. The magnitude of primal velocity (*a*) and pressure (*b*), and the magnitude of dual velocity (*c*) and pressure (*d*).

8. Summary

In this paper we have considered the flow past a sphere at $Re = 10^4$. The drag was computed using Adaptive DNS/LES, in which the computational mesh is refined adaptively until the error in a specified output, in this paper drag, is less than a given tolerance. The incompressible Navier–Stokes equations are solved using a stabilized Galerkin finite element method. Both the stopping criterion and the mesh refinement strategy are based on *a posteriori* error estimates, in the form of a space–time integral of residuals multiplied by derivatives of the solution of an associated dual problem, linearized at the approximate solution, and with data coupling to the output of interest. There is no filtering of the equations, and thus no Reynolds stresses are introduced. Instead the stabilization in the numerical method is acting as a simple turbulence model.

Using less than 10^5 mesh points, c_D is computed to an accuracy of a few percent, corresponding to experimental precision, which is at least an order of magnitude cheaper than standard non-adaptive LES computations in the literature. In addition, other output, such as transversal force components and Strouhal number, are also captured to experimental accuracy using Adaptive DNS/LES with respect to drag.

REFERENCES

Becker, R. & Rannacher, R. 2001 A posteriori error estimation in finite element methods. *Acta Numer.* **10**, 1–103.

Constantinescu, G. & Squires, K. 2004 Numerical investigations of flow over a sphere in the subcritical and supercritical regimes. *Phys. Fluids* **16**, 1449–1466.

Eriksson, K., Estep, D., Hansbo, P. & Johnson, C. 1995 Introduction to adaptive methods for differential equations. *Acta Numer.* **4**, 105–158.

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

Giles, M. & Süli, E. 2002 Adjoint methods for pdes: *a posteriori* error analysis and postprocessing by duality. *Acta Numer.* **11**, 145–236.

Hoffman, J. 2004 On duality based *a posteriori* error estimation in various norms and linear functionals for les. *SIAM J. Sci. Comput.* **26**, 178–195.

Hoffman, J. 2005 Computation of mean drag for bluff body problems using adaptive dns/les. *SIAM J. Sci. Comput.* **27**, 184–207.

Hoffman, J. & Johnson, C. 2006 A new approach to computational turbulence modeling. *Comput. Methods Appl. Mech. Engng* (in press).

Sagaut, P. 2001 *Large Eddy Simulation for Incompressible Flows*. Springer.

Schlichting, H. 1955 *Boundary Layer Theory*. McGraw-Hill.