INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2005; **48**:929–945
Published online 16 March 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.960

Validation of adaptive unstructured hexahedral mesh computations of flow around a wind turbine airfoil

H. Bijl*,†, A. H. van Zuijlen and A. van Mameren

Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands

SUMMARY

In this paper we investigate local adaptive refinement of unstructured hexahedral meshes for computations of the flow around the DU91 wind turbine airfoil. This is a 25% thick airfoil, found at the mid-span section of a wind turbine blade. Wind turbine applications typically involve unsteady flows due to changes in the angle of attack and to unsteady flow separation at high angles of attack. In order to obtain reasonably accurate results for all these conditions one should use a mesh which is refined in many regions, which is not computationally efficient. Our solution is to apply an automated mesh adaptation technique.

In this paper we test an adaptive refinement strategy developed for unstructured hexahedral meshes for steady flow conditions. The automated mesh adaptation is based on local flow sensors for pressure, velocity, density or a combination of these flow variables. This way the mesh is refined only in those regions necessary for high accuracy, retaining computational efficiency. A validation study is performed for two cases: attached flow at an angle of 6° and separated flow at 12°. The results obtained using our adaptive mesh strategy are compared with experimental data and with results obtained with an equally sized non-adapted mesh. From these computations it can be concluded that for a given computing time, adapted meshes result in solutions closer to the experimental data compared to non-adapted meshes for attached flow. Finally, we show results for unsteady computations. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: computational fluid dynamics; mesh adaptation; finite volume method; wind turbine airfoil

1. INTRODUCTION

In the design process of new rotor blades for wind turbines the need for aerodynamic predictions is very important. At present it is common in the design process and in the dynamic load

*Correspondence to: H. Bijl, Faculty of Aerospace Engineering, Delft University of Technology, P.O. Box 5058, 2600 GB Delft, The Netherlands.
†E-mail: H.Bijl@lr.tudelft.nl

Contract/grant sponsor: Publishing Arts Research Council; contract/grant number: 98-1846389

Received 7 March 2004
Revised 24 January 2005
Accepted 31 January 2005

calculations for turbine certification to use the blade element method (BEM) for calculation of the aerodynamic forces, as described in the review papers [1, 2]. With this method cases can be computed fast, due to many simplifications in the flow model. In comparisons with wind tunnel measurements uncertainties of up to 20% in the BEM load calculations have been reported. For the ever-growing investment costs and risks of ever-growing wind turbines a smaller uncertainty is necessary. At present increasing effort goes into the development and validation of full Navier–Stokes codes for simulation of flows around wind turbines. Of the 21 codes that were compared in the NREL unsteady aerodynamics experiment code comparison study [3, 4], three were Navier–Stokes codes. Another example is the European VISCEL research project, which included specific tasks for the validation and assessment of existing Navier–Stokes solvers [5, 6].

At the same time it is presently not feasible to use CFD directly in design iterations or in the determination of a complete load set for certification purposes, since the computer time needed is still excessive. A typical mesh around a blade contains around 6 million points and many cases are unsteady. All CFD computations for wind turbine flows found in literature employ a non-adapted mesh, refined in a large region where most activity is expected. The refinement regions are implemented manually and are large as they have to cover the unsteady wake at all times.

We believe that the efficiency of CFD calculations for wind turbine applications can be greatly improved using automated mesh adaption algorithms. Many papers have been published on adaptive algorithms for steady flows, from a theoretical approach [7], to tests of specific adaptive algorithms. Most of these papers contain a specific choice of adaptation parameters and do not provide an evaluation of their adaptive strategy. Furthermore, to our best knowledge no papers have appeared performing adaptive computations for flow around wind turbines.

For an accurate and efficient computation of time-dependent flows mesh adaptation is especially important as the phenomenon evolves through the computational domain. In order to avoid too many fine cells or remeshing too often the evolution of the flow phenomena has to be predicted with some accuracy. This prediction should also be efficient. Not many papers have appeared so far on grid adaptation for unsteady flows. Good results have been obtained for time-dependent flows with finite element discretization by [8, 9] with an internal loop in which the transient fixed point iteration is solved using error estimate based on the Hessian of the solution. However, an estimate of the work involved, especially the work saved by the adaptation methods was not given.

Most adaptive algorithms use unstructured tetrahedral [10] or hybrid meshes [11]. A disadvantage of tetrahedral meshes for high-Reynolds number viscous simulations, such as our wind turbine application, is the relatively low accuracy in boundary layers. A few papers have appeared on adaptive algorithms for fully Cartesian meshes [12, 13]. With these meshes special attention should be paid to the cut cells that intersect the bodies. Very few publications have appeared on body-fitted unstructured all-hexahedra meshes, mainly due to the great difficulty of devising automated grid generation algorithms. Recently, a new body-conforming octree mesh generation technique proposed by several authors [14, 15] has been adopted and extended to incorporate mesh adaptation by Patel [16]. These unstructured hexahedral meshes combine the favourable accuracy of hexahedral meshes inside boundary layers with the ease of grid generation of unstructured meshes. Furthermore, hexahedral cells can be easily refined anisotropically, which greatly enhances the efficiency of computations of flows with significantly differing length scales.

In this paper we validate the adaptive algorithm using unstructured hexahedral meshes developed by Patel [16, 17], comparing with experiments and results of non-adapted computations. Questions that will be addressed are: what is the best adaptation strategy? How much does the accuracy improve using this strategy? The algorithm which adapts to local flow features for unstructured hexahedral meshes will be tested for two-dimensional flow around the DU91 wind turbine airfoil. Experimental results are taken from the experimental database for the DU91 wind turbine airfoil that is being built at the Delft University of Technology. As three-dimensional and unsteady calculations are expensive and useless if we are unable to reproduce results for two-dimensional flows, we start with low angles of attack that result in steady two-dimensional flows. Thereafter validation is performed for larger angles of attack. For unsteady flows validation of unadapted computations using various time steps is performed. The results of this study provide directions for our current development of an efficient unsteady adaptive flow solver.

2. THE FLOW SOLVER

The flow solver used is Hexstream, developed by NUMECA Int. The flow is modelled by the Reynolds-averaged Navier–Stokes equations in combination with the one-equation turbulence model of Spalart–Almaras (S–A model) [18]. Results reported for this model for three-dimensional blade simulations showed reasonable accuracy [5] even at angles of attack in the beginning of the stall regime. The tripping point is chosen to be equal to the location found from the experiments.

Space discretization is performed on a body-conforming unstructured hexahedral mesh using a cell centred, conservative, finite volume scheme. The convective flux is discretized using a second order central scheme with Jameson-type scalar artificial dissipation [19]. Time integration is performed using a standard second order backward differencing technique. Solution of the implicit system of equations is obtained using a dual time stepping scheme with an explicit four-stage Runge–Kutta scheme [20]. Convergence acceleration for steady state is obtained using local time stepping and implicit residual smoothing. The solution procedure is embedded into a sophisticated agglomeration multigrid algorithm [16]. Although the solver has a low-Mach number preconditioning capability this was not used for the cases presented.

3. THE ADAPTIVE MESH ALGORITHM

The local mesh adaptation algorithm is fully automated and is based on the detection of flow variable gradients, see also Reference [16]. Mesh adaptation based on flow feature detection consists of three consecutive steps:

1. The flow solution is calculated on the initial mesh.
2. One or more suitable *sensors* are calculated in each cell.
3. *Threshold values* for refinement and coarsening are computed and cells are flagged for coarsening and refinement.
4. *Mesh coarsening* is performed. The flow solution is transferred to the new mesh using averaging and grid hierarchy.

5. *Mesh refinement* is performed on the coarsened mesh. The flow solution is transferred to the new mesh using grid hierarchy.
6. A new flow solution is computed on the newly adapted grid.

Please note that solution transfer to the new mesh in our algorithm only provides an initial solution for the flow solve on the new mesh performed in step 6. Steps 2–5 are now discussed in more detail. For more information see References [16, 17, 20].

3.1. Sensors

For the adaptation face-centred sensor formulations are used because they naturally involve anisotropic refinement. As sensors we have chosen pressure and velocity magnitude, since most of the flow features can then potentially be captured. For the present low-speed application, density is obviously not used. Flow feature detectors are computed on face centres according to the left and right neighbouring cells. For these flow feature detectors we use a finite difference formulation:

$$S_k = \frac{|U_L - U_R|}{U_{av}|} \tag{1}$$

where U_L and U_R are the flow quantity values in the left and right cells, respectively, and U_{av} is its average value over the whole computational domain. This is a non-dimensional formulation. Next to the above finite difference formulation, divided and multiplied difference formulations can also be found in the literature [16]. In these cases Equation (1) is either divided or multiplied by the distance between the neighbouring cell centres resulting in a dimensional feature detector. In the divided difference the distance term tends to increase the detector value in small cells and therefore strengthens the sensor behaviour. The multiplied difference has an opposite behaviour and may asymptotically prevent refinement in excessively clustered regions. Their behaviour, however, is extremely case dependent. Other formulations like pure second order differences [21] or ratios of first and second order differences [22] have also been proposed and might be tested in the future.

3.2. Threshold values

Threshold values are necessary to determine which cells should be refined or coarsened. Instead of relying completely on a manual threshold determination a common statistical formulation is used. Assuming the sensor statistical distribution over the domain is Gaussian, the optimum threshold values for refinement T_r and coarsening T_c are defined by [23]:

$$T_r = S_{av} + \beta_r S_{std} \tag{2}$$

$$T_c = S_{av} - \beta_c S_{std} \tag{3}$$

In these equations S_{av} denotes the average value of the sensor in the flow field and S_{std} is the standard deviation. Furthermore, β_r and β_c are refinement and coarsening parameters respectively. In the next section the influence of these parameters on the adaptation procedure and solution is investigated.

3.3. Refinement and coarsening

When the sensor evaluation in a face exceeds the prescribed refinement threshold, the neighbouring cells are flagged for refinement in the direction parallel to the face. A balancing function is implemented to smooth refinement by propagating flags such that only one hanging node/edge is tolerated and the difference in refinement level between two neighbouring cells in the direction parallel to their common face cannot exceed one. Furthermore, holes in the mesh are prevented by refining cells of which the two neighbouring cells are flagged for refinement too.

The mesh coarsening technique is based on the hierarchical property of the mesh adaptation data structure. As the mesh entities' parents are stored during the refinement process, these parents can be recovered when their children are removed. Mesh coarsening is less local than refinement, as all siblings are considered before a cell is coarsened. Due to the following two properties coarsening is less likely to occur than refinement:

- When siblings are flagged for both coarsening and refinement, which is possible when two or more sensors are used, refinement has priority.
- At least 75% of the sibling cells must be flagged for coarsening, then the remaining sibling cells are also flagged for coarsening, even if they were flagged for refinement.

The last point can be demonstrated with the following case. For a cell to be coarsened three of its four cells should be flagged for coarsening. After coarsening of this cell the number of cells is reduced by three. So for coarsening we need three flags for three cells less. For refinement there is a case where we only need two flags for five extra cells. That is, when two faces of a cell of which one is oriented in the local ξ and one along the η axis are flagged for refinement, the cells on each side of the face will be divided in two. As a result the cell will be divided into four cells and two neighbouring cells will be refined in one direction too. In the old situation we had three cells in the new we have eight. As adaptation for an actual flow computation will consist of a variety of refinement and coarsening configurations it is hard to exactly predict the growth in number of cells. However, we can say that when the sensor has a Gaussian distribution, the number of coarsening and refinement flags will be equal. Consequently, due to the above described process the total number of cells will increase.

4. THE PROBLEM

The airfoil we selected is the 25% thick DU91 wind turbine airfoil (Figure 2), found typically at the mid-span section of a wind turbine blade. The airfoil was designed at the Institute for Wind Energy of the Delft University of Technology; its full name is DU91-W2-250 [24]. The thickness of this airfoil is quite large compared to airfoils commonly used in aeronautics, which complicates the computation of flows with flow solvers. It is therefore an ideal test case to judge the capabilities of the current flow solver and in particular the adaptive mesh strategy.

4.1. Experiments

The experiments for validation of the computational results were performed at the low-speed-low-turbulence wind tunnel of the Delft University of Technology. The wind tunnel has a

Table I. Experimental results corrected for wind tunnel walls.

Angle	*Re*	*M*	C_l	C_d	Trans.pt. u.s.	Trans.pt. l.s.
1°	3×10^6	0.21	0.525	0.00777	0.47	0.46
6.24°	1×10^6	0.174	1.132	0.01209	0.43	0.53
12.24°	1×10^6	0.174	1.216	0.05067	0.12	0.58
40°	1×10^6	0.174	0.976	0.840	0	0

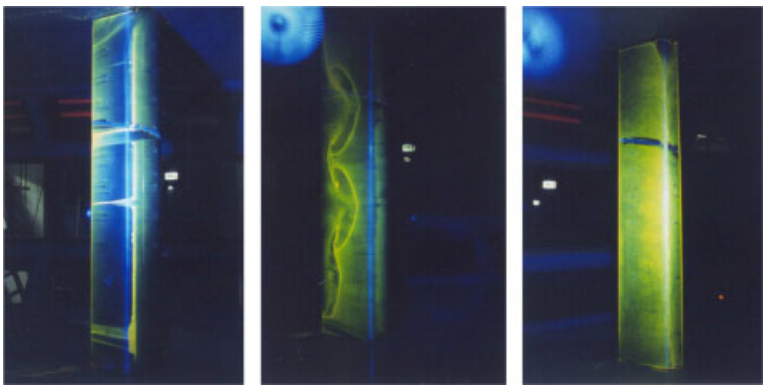

Figure 1. Flow visualization for angles of attack of 6.24°, 12.24° and 40°.

low turbulence intensity of 0.06%. Its test section has a cross section of 1.23×1.80 m. The experiments were performed on a clean-configuration without forced transition–tripping. The resulting natural transition and separation points were measured on both the upper and lower surface. Four angles of attack equal to 1°, 6.21°, 12.21°, and 40° were chosen, as each of these is representative for a specific flow regime. Reynolds numbers range from 1×10^6 to 3×10^6, Mach numbers from 0.174 to 0.21. The angles of attack mentioned earlier are the actual angles in the wind tunnel. As the computations have been performed for a free air case, corrections for the wind tunnel walls have to be made to the experimental lift and drag coefficients and angles of attack. These corrections are based on Allen and Vincenti's method [25]. The corrected results are presented in Table I.

For an angle of attack of 1° the flow is fully attached. At 6° there is a very small separation bubble, while 12° is in the stall regime. As can be seen from Figure 1 stall cells develop along the blade. At 40° the flow is massively separated. The result is an unsteady flow, where vortices are shed downstream. The oil pattern shown in Figure 1 is too slow to capture this. For this unsteady case the lift and drag coefficients shown in Table I are averages over time. Note that the flow topology for 1° and 6° is two-dimensional. The other angles of attack are included in order to determine the difference between two- and three-dimensional flow topologies for these cases. For example, for the unsteady flow at the largest angle of attack time averaged values of experiment and computation might compare well with each other.

4.2. Computations

The boundaries of the computational domain are placed at 10 chords upstream, 60 chords downstream and 30 chords above and below the airfoil. For this choice it was found that extension of the domain by a factor of 1.5 only resulted in changes in C_l of 0.05% and in C_d of 0.4%. Boundary conditions upstream and on the upper and lower boundary are freestream conditions. At the downstream boundary condition a pressure outflow condition is given. For the Spalart–Allmaras turbulence model, the transition points are given to be equal to those found in the experiments. The iterative convergence criterion is that the residuals have dropped five orders of magnitude with respect to the initial solution. The adaptation strategy will be investigated for 1° angle of attack. Hereafter validation will be performed for 6°, 12° and 40°.

5. DETERMINATION OF ADAPTIVE STRATEGY

For steady, attached flow at an angle of attack of 1° an investigation into the adaptive mesh strategy is performed. Questions are the choice of: (a) the flow sensor(s) (b) the refinement and coarsening parameters and (c) the number of mesh adaptation steps. The Reynolds number is 3×10^6, the Mach number is 0.21. The initial solution is a uniform flow field with $p = 1$ atm, $\rho = 1.225\ \mathrm{kg/m^3}$, $T = 288\ \mathrm{K}$ and $V = 71.44\ \mathrm{m/s}$. The initial mesh has a box structure with a set of uniformly refined boxes close to the airfoil, see Figure 2. This initial mesh was chosen after comparison of lift and drag coefficients on non-adapted meshes of similar size. The initial mesh has approximately 41 k cells.

In order to assess the accuracy we have to compare our results to a reference or true solution. As true solution we take in this case the experimental lift and drag coefficient. These experimental values are not the true solution of our model equations, as the difference is equal to the model error. However, as can be seen from Table II for this case the lift and drag coefficient converge to the experimental values as the number of adaptations is increased, so we can conclude that the numerical error on the present meshes is dominant. Furthermore, due to its small value for this case the relative error in drag ($\%\Delta C_d$) is much harder to predict correctly, than that in lift.

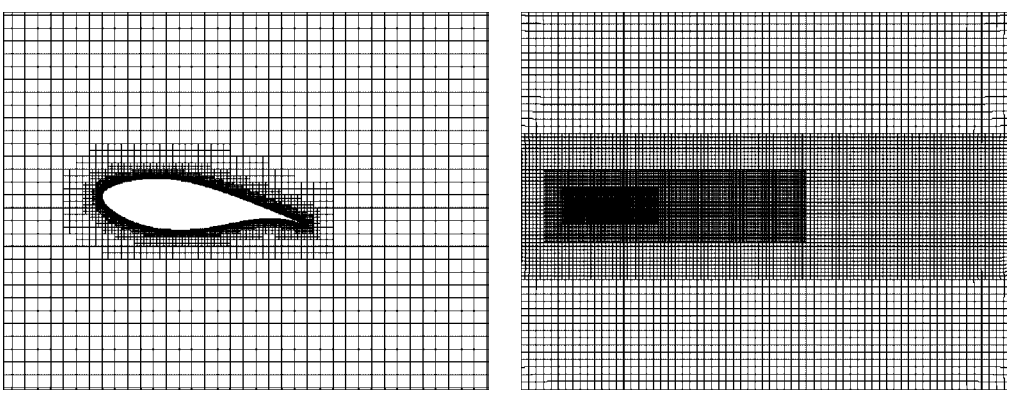

Figure 2. Initial mesh with approximately 41k cells.

Table II. Lift and Drag coefficients for 0 to 3 adaptation steps compared to experimental values; pressure sensor, $\beta_r = \beta_c = 0.3$.

Source	# cells	C_l	% ΔC_l	C_d	% ΔC_d
No adaptation	41k	0.566	6.9	0.01355	74.3
1 adaptation	73k	0.574	8.9	0.00923	18.8
2 adaptations	153k	0.555	5.3	0.00867	11.6
3 adaptations	362k	0.545	3.4	0.00855	10.0
Experiment		0.527		0.00777	

5.1. *Influence of flow sensor*

First, the influence of the flow sensor is investigated. Hereto meshes and resulting lift and drag coefficients are compared for three sensors: the velocity sensor, the pressure sensor and a combination of velocity and pressure sensor. These sensors were chosen because for the present low-speed application, pressure and velocity are the principle unknowns. For this study the refinement and coarsening parameters are kept constant at 0.3 and the number of adaptation steps is equal to three.

In Figure 3 the pressure coefficient c_p for adaptation based on the three sensor choices is compared to the experiment. The largest difference is found at 0.8 of the chord at the lower surface of the airfoil, where the pressure sensor clearly results in the most accurate c_p. This is not immediately clear, studying the lift and drag coefficient only. For the lift coefficient the underprediction of all three computations at the bottom between 0.05 and 0.4 is compensated by the overprediction at the bottom around 0.8. Therefore, the lift and drag coefficients presented in Table III are slightly misleading. However, as the relative difference between lift predictions (3.4% vs 2.7%) is much smaller than that for the drag prediction (10% vs 26%), and as the drag prediction is the best for the pressure sensor, the pressure sensor is chosen for the rest of the adaptation study. Although, we have been unable to identify the exact reason for the success of the pressure sensor, a closer look at the resulting meshes sheds some light at the underlying reason.

In Figures 4–6 the adapted meshes are shown for the three sensors. As can also be seen from Table III the velocity sensor leads to the smallest mesh. Apparently the finite difference sensor based on the velocity has a more compactly distribution, so that the same threshold value of 0.3 results in much less refinement. For the velocity sensor the cells in the wake are not coarsened as much as for the other two sensors. It can also be seen that the adapted meshes for the pressure sensor and the velocity and pressure sensor combined resemble each other closely. For the combined sensor in the far wake the coarsening of the wake due to the pressure sensor dominates, although close to the airfoil a few more cells in the wake are refined. The very fine mesh close to the airfoil resulting from the pressure sensor is probably responsible for the more accurate c_p and drag prediction for these cases. The reason for the increased accuracy of the drag for the pressure sensor alone remains unclear.

Knowing that the initial mesh consisted of 41k cells, from Table III it is found that the number of mesh cells increases significantly for all three sensors. The increase is between 370% for the velocity and 976% for the combined sensor. There can be three reasons for this

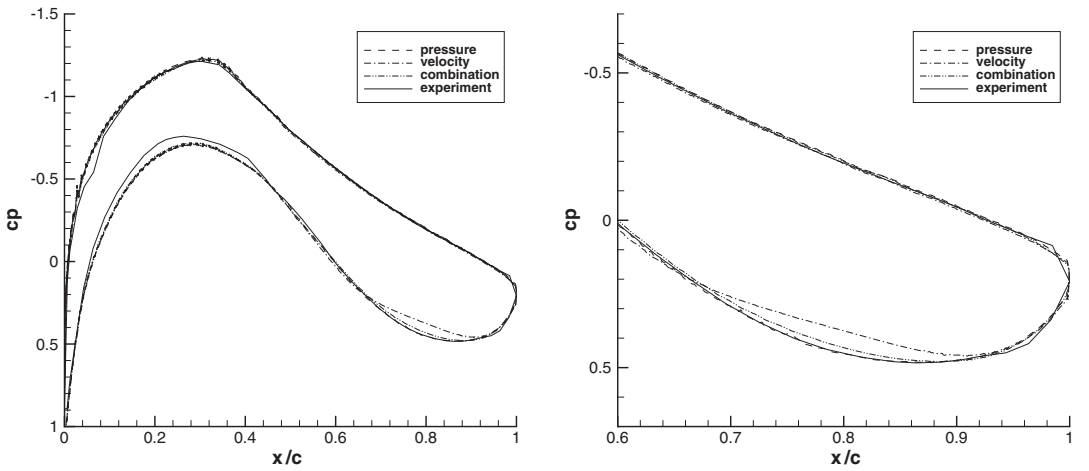

Figure 3. Pressure coefficient for the three different adaptation sensors compared to the experiment; $\beta_r = \beta_c = 0.3$.

Table III. Lift and Drag coefficients compared to experimental values for pressure sensor, velocity sensor and pressure and velocity combined; three adaptation steps, $\beta_r = \beta_c = 0.3$.

Sensor	# cells	% ΔC_l	% ΔC_d
p	362k	3.4	10.0
u	192k	2.7	26.0
p and u	441k	2.8	17.5

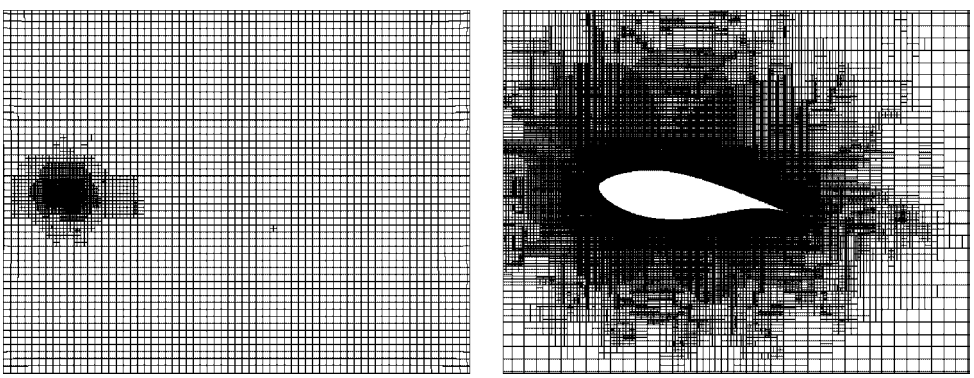

Figure 4. Mesh after three adaptations with pressure sensor; $\beta_r = \beta_c = 0.3$.

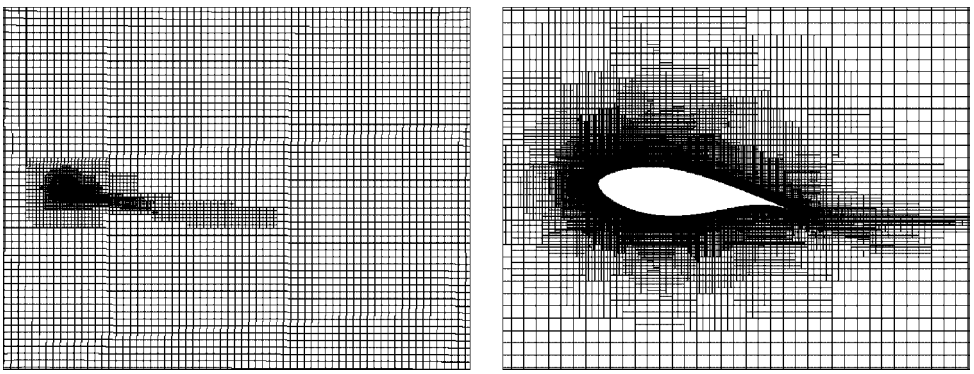

Figure 5. Mesh after three adaptations with velocity sensor; $\beta_r = \beta_c = 0.3$.

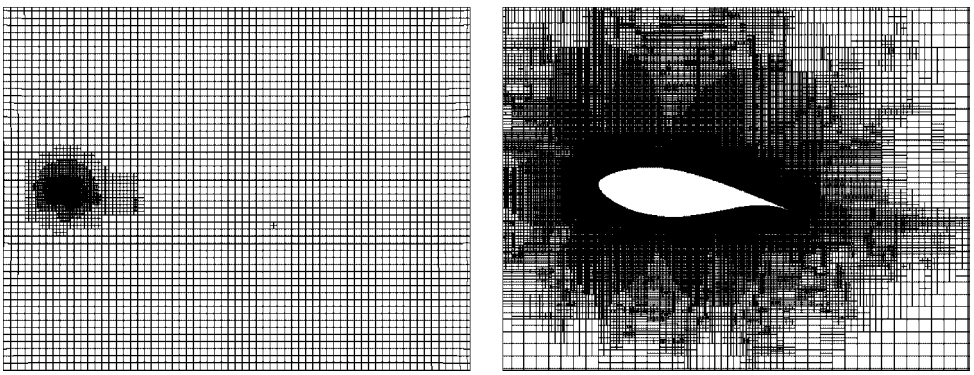

Figure 6. Mesh after three adaptations with pressure and velocity sensor; $\beta_r = \beta_c = 0.3$.

growth:

1. Refinement is more likely to occur due to parameter choices in the adaptation process (see Section 3).
2. There is a limit to the amount of coarsening, when the parent level is reached.
3. The distribution of the sensors is not symmetrical with a bias towards refinement.

It is not straightforward to determine which of these is major and which minor. This is currently under investigation.

5.2. *Influence of threshold values*

As discussed in the previous section our threshold strategy involves the choice of two parameters: β_r the refinement parameter and β_c the coarsening parameter. When they are chosen equal the total number of cells will grow, due to the fact that in our strategy (also described in the previous section) refinement is more like to occur than coarsening. As the cell growth is highly problem and sensor dependent, we decided for this study to chose β_r and β_c equal.

Table IV. Lift and Drag coefficients compared to experimental values for different threshold values; pressure sensor, three adaptation steps.

$\alpha=\beta$	# cells	% ΔC_l	% ΔC_d
0.3	362k	3.4	10.0
0.5	311k	4.4	10.3
0.8	239k	5.7	13.0

Values of α and β close to 0 or 1 were far from optimal. Values of β_r slightly close to 0 result in a situation where almost all cells are refined. Preliminary tests showed that β_r should at least be chosen larger than 0.1. For values of β_r and β_c above 1 very few cells were marked for refinement or coarsening resulting in a largely unchanged mesh.

Therefore, β_r and β_c were chosen to be 0.3, 0.5 and 0.8, together with the pressure sensor and three adaptation steps. Note that for small β_r and β_c, the threshold-values are closer to the mean, so that more mesh cells are adapted. From Table IV it is found that more adaptation leads to more mesh cells, 362k for $\beta_r = \beta_c = 0.3$ vs 239k for $\beta_r = \beta_c = 0.8$. The most accurate result is obtained on the most strongly adapted mesh. Choices smaller than 0.3 are, of course, also possible. However, these cases resulted in meshes larger than 400k, too large for our current workstation, and also outside the range we would recommend for two-dimensional airfoil RANS computations. Another option to test smaller threshold values is to start from a smaller mesh. However, it turned out that for a same accuracy more adaptation steps were necessary.

5.3. Influence of number of adaptation steps

In Table II the results for none up to three adaptation steps are shown for the pressure sensor and $\beta_r = \beta_c = 0.3$. The resulting meshes are shown in Figures 7–9. From this it is found that the first adaptation, is mainly used for refinement very close to the airfoil and along the pressure contours in the neighbourhood of the airfoil, which leads to the largest improvement of the drag coefficient from a 74.3 to 18.8% difference with the experiment. At the same time, unfortunately, the lift coefficient deteriorates from a 6.9 to 8.9% difference with the experiment. Even after detailed study of the results we were unable to identify the reason for this. However, the relative improvement of the drag prediction, known to be the hardest of the two, is much larger than the lift deterioration. During the second and third adaptation steps the refinement in the same regions is even stronger. In these steps also the coarsening of the mesh further away from the airfoil becomes apparent. In the second and third adaptation the lift coefficient improves again, to a difference of 3.4%, half of the value on the initial grid. The improvement of the drag coefficient clearly levels off: 18.8–11.6% and 10.0%. More adaptations led only to even smaller improvements, of which it might be concluded that after three adaptations the modelling error has become dominant.

6. VALIDATION OF STEADY COMPUTATIONS

The adaptative mesh strategy determined in the previous section for 1° is now validated for the other steady cases, i.e. angles of attack of 6.24° and 12.24°, described in Section 4.

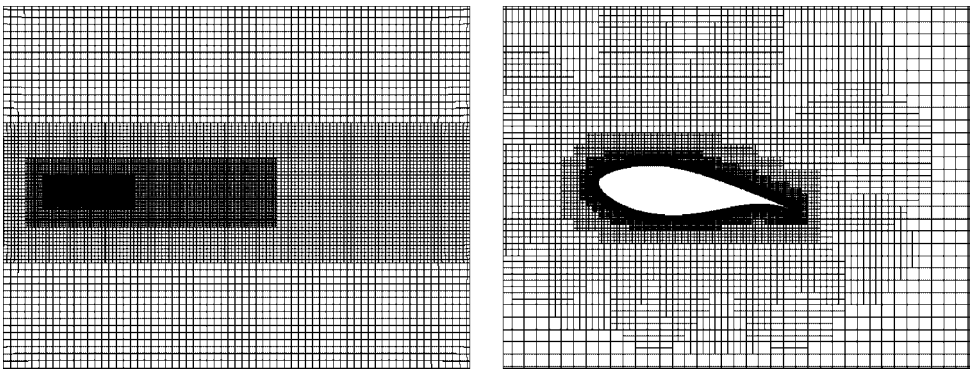

Figure 7. Mesh after one adaptation with pressure sensor; $\beta_r = \beta_c = 0.3$.

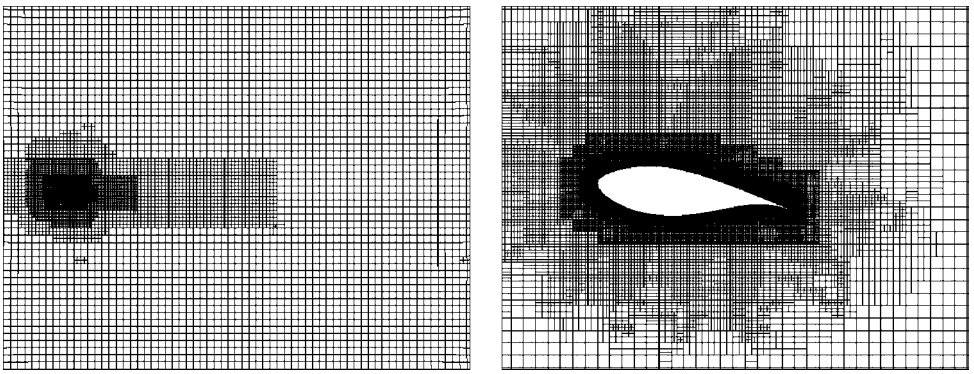

Figure 8. Mesh after two adaptations with pressure sensor; $\beta_r = \beta_c = 0.3$.

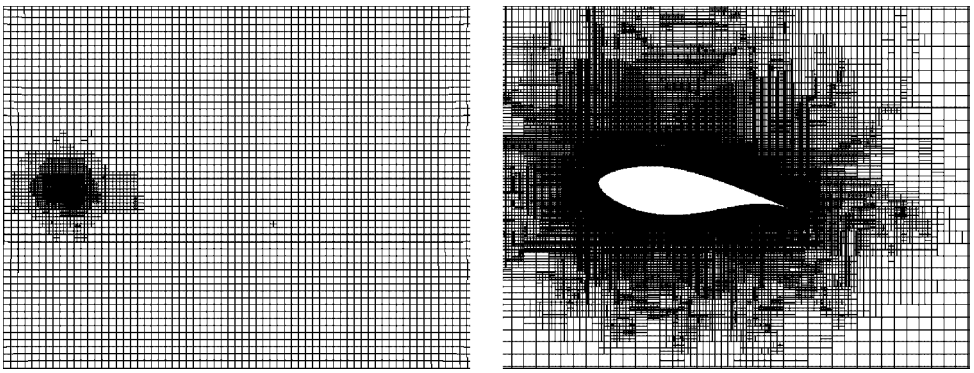

Figure 9. Mesh after three adaptations with pressure sensor; $\beta_r = \beta_c = 0.3$.

Hereto the lift and drag coefficients are compared to those computed on a non-adapted mesh, results computed with another non-adaptive flow solver and with the experiments. This other flow solver is also a RANS solver with the same S–A model. It has the same discretization technique as ours. Main differences are the character of the mesh—structured vs unstructured—the computational mesh, which consists of almost twice the number of mesh points, and the iterative solution technique—pressure correction vs pseudo-time. The adaptation strategy consists of a combination of the pressure sensor, $\beta_r = \beta_c = 0.3$ and three adaptation steps.

6.1. *The* 6.24° *test case*

The results of the validation for the 6.24° test case are presented in Table V. From this it can be concluded that adaptation significantly improves the relative accuracy of the drag coefficient (from 34 to 21%), with a small deterioration of the lift coefficient (1 to 3%). The different flow solver without adaptive capabilities with 50% more mesh cells is significantly more accurate in the drag prediction (5 vs 21%), although the lift is less accurate (6 vs 3%). Also pressure coefficients are compared between adapted computations and experiment, see Figure 10. Then it is found that especially the small plateau around the small separation

Table V. Validation of the 6.24° test case.

	# cells	C_l		C_d	
Experiment		1.132		0.01209	
Adaptation	334k	1.161	3%	0.01459	21%
No adaptation	334k	1.147	1%	0.01615	34%
Other flow solver	501k	1.197	6%	0.0115221	5%

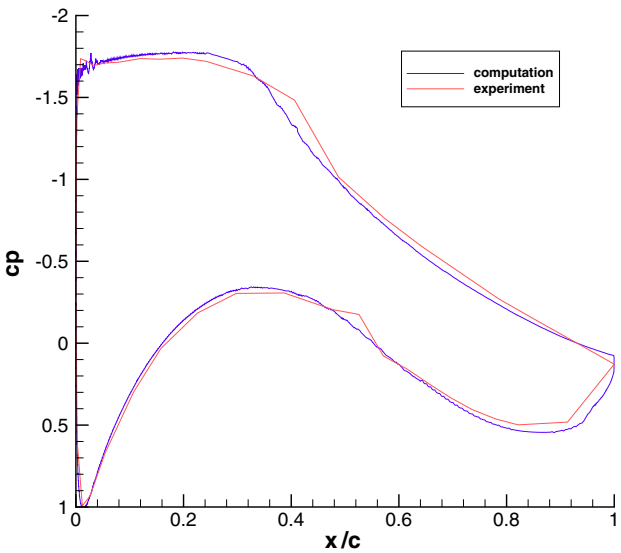

Figure 10. Pressure distribution at 6.24° compared to experimental results; pressure sensor, $\beta_r = \beta_c = 0.3$, three adaptation steps.

bubble on the lower side of the airfoil at 0.5 is not captured. As the pressure only changes weakly in a separation region, this might be an artefact of our pressure sensor. The velocity sensor or a combination of velocity and pressure, however, did not change this behaviour. Also interesting to note are the wiggles in c_p at the nose of the airfoil. These wiggles are caused by the cell size of the computational mesh that has become smaller than the cell size of the geometric representation of the airfoil. As a consequence the 'abrupt' changes in the geometry are seen by the mesh. This might also decrease the overall accuracy.

6.2. The 12.24° test case

The results of the validation for the 12.24° test case are presented in Table VI. This case is much harder than the lower angle of attack cases, since there is a significant area with separated flow, as can be seen from the plateau in the pressure coefficient shown in Figure 11. Due to the three-dimensional nature of the flow, see Figure 1, it is hard to draw conclusions from the present two-dimensional computations. Furthermore, for this case the measurements have a larger uncertainty caused by corrections performed to the wind tunnel data in order to

Table VI. Validation of the 12.24° test case.

	# cells	C_l		C_d	
Experiment		1.216		0.05067	
Adaptation	304k	1.569	29%	0.03985	21%
No adaptation	304k	1.400	15%	0.03975	22%
Other flow solver	501k	1.529	26%	0.03815	25%

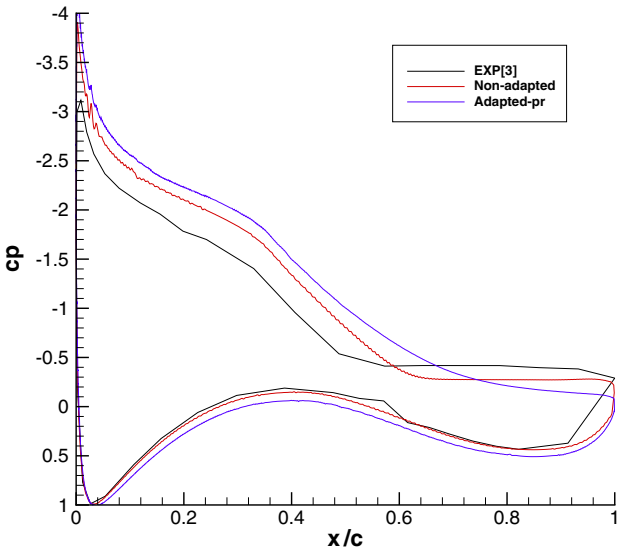

Figure 11. Pressure distribution at 12.24° compared to experimental results; pressure sensor, $\beta_r = \beta_c = 0.3$, three adaptation steps.

eliminate the influence of the wind tunnel walls which are inaccurate at larger angles of attack. Both the other CFD code with 50% more cells and the adapted version of the present code lead to less accurate results. It is, however, questionable at which location along the blade span the experimental results should be taken for comparison. Therefore, three-dimensional computations and measurements at various spanwise locations should be performed.

7. VALIDATION OF UNSTEADY COMPUTATIONS

As we are ultimately interested in unsteady flows, validation for an unsteady case is performed too. At this stage, however, the adaptation algorithm is not suitable for unsteady flows, so only non-adapted unsteady computations were performed. The computations are performed for an angle of attack of 40° at which the flow is massively separated. A computational mesh of 41 k cells was used, as more cells would lead to very large computational times. A time step refinement is performed: 4 time steps, of which three resulted in an unsteady flow, see Table VII. The computed frequency seems to converge to a value slightly larger, in the order of 5%, than the measured frequency. This slight overprediction is likely caused by the three-dimensional effects that are not captured in our two-dimensional computations. For two-dimensional computations of vortex shedding behind a circular cylinder similar overpredictions of the frequency compared to the experimental value have been found [26–29]. The lift and drag coefficient converge to values much larger than the measured values. For two-dimensional computations of flow around a circular cylinder similar differences have been reported [27, 29]. Furthermore, Mittal and Balachandar [28] showed by comparing two-dimensional to three-dimensional computations that this is due to the three-dimensional nature of the flow, which resulted in a lower suction peak. Another possible explanation is the turbulence model, which is known to be less accurate for separated flows. For the largest time step of five points/wavelength a steady solution is obtained. It is interesting to note that for this case the lift and drag coefficient prediction are considerably closer to the experimental values than for the unsteady cases. The time averaged pressure coefficient for the experiments and unsteady computations using the smallest time step is compared to the pressure coefficient for the steady computations in Figure 12. There it can be seen that the unsteady computations have a much stronger suction peak at the upper surface. Furthermore, the effect of the vortex shedding can be seen from the increased pressure coefficient at the end of the upper surface. As lift and drag were overpredicted in the unsteady computations addition of significant numerical dissipation from the large time step results in lower time averaged values, which, in this case, are closer to the experimental values.

Table VII. Validation of the 40° test case.

	Pnts/wavelength	av. C_l	av. C_d	Freq.
Experiment	—	0.98	0.84	39.3
No adaptation	25.4	1.67	1.24	40.3
No adaptation	12.7	1.65	1.23	39.7
No adaptation	8.5	1.54	1.18	36.5
No adaptation	5	1.09	0.84	—

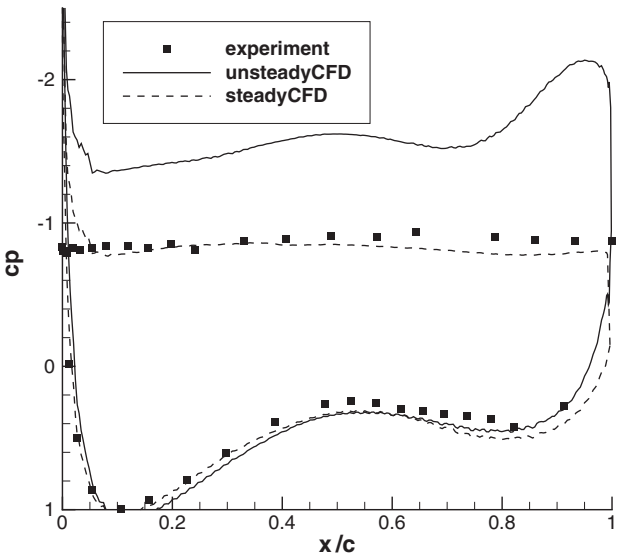

Figure 12. Pressure distribution at 12.24° compared to experimental results; pressure sensor, $\beta_r = \beta_c = 0.3$, three adaptation steps.

8. CONCLUSIONS

The present validation of adaptive mesh computations of flow around a wind turbine airfoil shows that:

- a good local adaptation strategy is obtained using:
  - pressure sensor
  - coarsening and refinement parameters $\beta_r = \beta_c = 0.3$.
  - three adaptation steps
- for steady attached flows the accuracy using adaptation improves
- for steady separated flow the accuracy using adaptation seems to deteriorate. This might be due to the three dimensional nature of the flow and/or the accuracy of the turbulence model
- for unsteady flows the non-adapted flow solver gives promising results

Adaptive meshing clearly increases the potential of CFD for prediction of flows around wind turbines. Currently, the adaptation strategy is made suitable for unsteady flow computations. In the near future also adapted computations of unsteady cases with varying angles of attack will be presented.

REFERENCES

1. Snel H. Review of the present status of rotor aerodynamics. *Wind Energy* 1998; **1**:46–69.
2. Snel H. Review of aerodynamics for wind turbines. *Wind Energy* 2003; **6**:203–211.

3. Simms D, Schreck S, Hand M, Fingersh LJ. NREL unsteady aerodynamics experiment in the NASA-Ames wind tunnel: a comparison of predictions to measurements. *NREL/TP-500-29494*, National Renewable Energy Laboratory, USA, June 2001.
4. Schreck S *et al*. The NREL full-scale wind tunnel experiment (special issue). *Wind Energy* 2002; **5**:77–257.
5. Chavriaropoulos PK *et al*. Viscous and aeroelastic effects on wind turbine blades. The VISCEL project. Part 1: 3D Navier–Stokes rotor simulations. *Wind Energy* 2003; **6**:365–385.
6. Chavriaropoulos PK *et al*. Viscous and aeroelastic effects on wind turbine blades. The VISCEL project. Part 2: aeroelastic stability investigations. *Wind Energy* 2003; **6**:387–403.
7. Dervieux A, Leservoisier D, George P-L, Coudiere Y. About theoretical and practical impact of mesh adaptation on approximation of functions and PDE solutions. *International Journal for Numerical Methods in Fluids* 2003; **43**:507–516.
8. Alauzet F, George PL, Mohammadi B, Frey P, Borouchaki H. Transient fixed point-based unstructured mesh adaptation. *International Journal for Numerical Methods in Fluids* 2003; **43**:729–745.
9. Pain CC, Umpleby AP, Oliveira CRE, Goddard AJH. Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:3771–3796.
10. Castro-Diaz MJ, Hecht F, Mohammadi B, Prionneau O. Anisotropic unstructured mesh adaptation for flow simulations. *International Journal for Numerical Methods in Fluids* 1997; **25**(4):475–491.
11. Mavriplis DJ. Adaptive meshing techniques for viscous flow calculations on mixed element unstructured meshes. *International Journal for Numerical Methods in Fluids* 2000; **34**:93–111.
12. Coirier WJ, Powell KG. Solution-adaptive Cartesian cell approach for viscous and inviscid flows. *AIAA Journal* 1996; **34**(5):938–945.
13. Ham FE, Lien FS, Strong AB. A Cartesian grid method with transient anisotropic adaptation. *Journal of Computational Physics* 2002; **179**:469–494.
14. Tchon KF, Hirsch C, Schneiders R. Octree-based hexahedral mesh generation for viscous flow simulations. *AIAA Paper 97-1980*, *13th AIAA CFD Conference*, Snowmass, CA, June 1997.
15. Wang ZJ, Chen RF, Hariharan N, Przekwas AJ. A 2^N tree bases automated viscous Cartesian grid methodology for feature capturing. *AIAA Paper 99-3300-CP*, *14th CFD Conference*, Norfolk, VA, June 1999.
16. Patel A. Development of an unstructured RANS solver for unstructured hexahedral meshes. *Ph.D. Thesis*, Vrije Universiteit Bruseel, 2003.
17. Patel A, Hirsch C. All hexahedra unstructured flow solver for external aerodynamics with application to aeroelasticity. *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, vol. 81. Springer: Berlin, 2002; 105–116.
18. Spalart PR, Allmaras SR. A one-equation turbulence model for aerodynamic flows. *AIAA Paper 92-0439*, 1992.
19. Jameson A, Schmidt W, Turkel E. Numerical solutions of the Euler equations by finite volume methods with Runge–Kutta time stepping schemes. *AIAA Paper 81-1259*, 1981.
20. NUMECA International. Numeca's flow integrated environment for turbomachinery and internal flows. *User Manual*, Belgium, April 2000.
21. Holmes DG, Connell SD. Solution of the 2D Navier–Stokes equations on unstructured adaptive grids. *AIAA Paper 89-1932*, 1989.
22. Brandt A. Multilevel adaptive solutions to boundary value problems. *Mathematics of Computation* 1977; **31**: 333–390.
23. Kallinderis Y, Baron JR. Adaptation methods for a new Navier–Stokes algorithm. *AIAA Journal* 1989; **27**(1): 37–43.
24. Timmer WA. Some aspects of high angle-of-attack flow on airfoils for wind turbine application. *EWEC 2001 European Wind Energy Conference*, 2001.
25. Allen HJ, Vincenti WG. Wall interference in a two-dimensional wind tunnel with consideration of the effect of compressibility. *NACA Report No. 782*, 1947.
26. Bijl H, Carpenter MH. Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow. *Journal of Computational Physics* 2002; **179**:313–329.
27. Cox JR, Brentner KS, Rumsey CL. Computation of vortex shedding and radiated sound for a circular cylinder: sub-critical to transcritical Reynolds number. *Theoretical and Computational Fluid Dynamics* 1998; **12**(4): 233–253.
28. Mittal R, Balachander S. Effect of three-dimensionality on the lift and drag of nominal two-dimensional cylinders. *Physics of Fluids* 1995; **7**(8):1841–1865.
29. Tang SJ, Aubry N. On the symmetry breaking instability leading to vortex shedding. *Physics of Fluids* 1997; **9**(9):2550–2571.

