Simulation of Maximum Lift Using URANS for a High-Lift Transport Configuration

J.W. van der Burg and M. Luehmann

Airbus Operations GmbH, Airbusallee 1, 28199 Bremen
http://www.airbus.com/

1 Introduction

Maximum lift prediction for a modern realistic transport high-lift configurations using a Computational Fluid Dynamics (CFD) based method still poses a significant challenge. The flow around a high-lift aircraft configuration can be characterized by a number of physical flow features such as flow separation, laminar/turbulent boundary layers, wakes of wing elements, vortical flows structures, vortex break down, but also deformation of the wing and its components. The accurate modelling of these individual physical features require the development, verification and validation of specific CFD algorithms. This paper focuses only on one of these algorithms, namely a time-integration algorithm.

Flow around a realistic high-lift configuration can be expected to be unsteady especially near maximum lift. Therefore, it is necessary to verify and apply unsteady time-integration algorithms for these flow conditions.

The intention of this work is to explore the possibilities that current unsteady Reynolds-Averaged Navier-Stokes (RANS) capabilities offer in modelling unsteady high-lift flows. Unsteady RANS (URANS) methodology is not (yet) widely applied to three-dimensional high-lift configurations basically due to the inherent high computational costs. Current practice is to compute high-lift flows with a steady flow solution method, where typically a Runge-Kutta or a lower-upper symmetric Gauss-Seidel (LUSGS) time integration algorithm using convergence acceleration techniques (local time stepping, multigrid) are employed.

The aim of this CFD study is threefold, firstly to get some more experience with unsteady flow computations for a realistic high-lift configuration. Secondly, alternative computational strategies are investigated in order to contribute to a best practice for unsteady flow computations. And finally, the aim is to attempt to improve maximum lift prediction, if possible, and otherwise to understand why maximum lift is not reached. In the latter case it is important to try to assess the dominant effect that prohibits maximum lift prediction.

In this paper firstly, the computational set-up is described. Next, the results of a number of alternative computational strategies including unsteady computations are presented. Subsequently, an extended post-processing analysis is made to try to understand and explain the occuring lift break down near maximum lift.

N. Kroll et al. (eds.), *Computational Flight Testing*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 123,
DOI: 10.1007/978-3-642-38877-4_6, © Springer-Verlag Berlin Heidelberg 2013

2 Unsteady Flow Computation

In an unsteady flow computation based on the dual time step approach the size of the time step needs to be chosen. It is observed in various research studies that for a three dimensional high-lift configuration the time step can not be taken too small since the computational effort becomes too large, see Refs. [1], [2], [3]. Therefore, a time integration approach with a relatively large time step is investigated here. The idea of taking a relatively large time step, is that high frequency components will be damped and low frequency components are computed/modelled. In an unsteady flow computation the time step in each dual grid cell is the same. In a steady flow computation the time step in each dual cell differs due to the local time stepping algorithm approach adopted. For a limited number of three-dimensional high-lift configurations it is demonstrated that an unsteady flow computation can stabilize the aerodynamic coefficients. Nevertheless, more experience with unsteady flow computations is necessary. For instance it could be beneficial to employ an unsteady time integration method in the initial startup phase of the computation where the flow is basically unsteady.

Some notation is introduced to explain the choice of the time step size. A characteristic time step can be derived from the reference velocity and reference as length

$$\Delta T_{ref} = \frac{L_{ref}}{u_{ref}} \tag{1}$$

which is basically the time that the flow needs to travel one chord length. In the computational study the following time steps are considered (approximately): $\Delta T = \Delta T_{ref}$, $\Delta T = \Delta T_{ref}/10$ and $\Delta T = \Delta T_{ref}/30$. In Refs. [1], [2], [3] much smaller time steps are considered.

3 Computational Set-Up

3.1 The Geometry of the High-Lift Configuration

In the CFD study described here the wind tunnel geometry of a high-lift transport configuration is studied. The high-lift configuration includes a through-flow nacelle. At the main wing leading edge inboard a droop-nose device is installed and outboard a slat is deployed. At the main wing trailing edge an inner flap, outer flap and an aileron are deployed, see Figure 1. At the wing tip a winglet is installed.

3.2 Computational Grid, Flow Conditions, Flow Solver Settings

Starting point for the CFD study is a polar computation on an existing unstructured grid which for the purpose of this study is referred to as the baseline computation. The computational grid has 29.8 million points and is created with the current practice for Centaur meshing [5]. For the flow computations the Tau

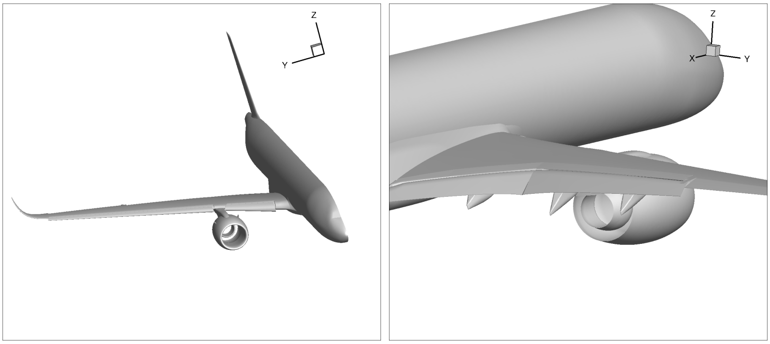

Fig. 1. Front and rear view of the transport aircraft configuration

flow solver, see [4], is utilised. Fully turbulent flow computations are performed using the Spalart-Allmaras turbulence model on 72 processors.

In order to describe the computational strategies some notation needs to be introduced for the angles of attack. Let α_0 be an angle in the mid linear range of the lift polar and define $\alpha_{i+1} = \alpha_i + c$ degree with c a fixed constant. The largest angle considered in this study is α_{10}.

The practice to compute maximum lift for a high-lift configuration is to perform an alpha sweep with the steady flow solver starting in the linear range α_0. The flow solution for angle α_i is then determined by restarting from the flow solution at α_{i-1}. A start from an initial uniform flow solution may lead to a different solution for larger angles and hence this approach is not pursued.

In the unsteady flow computations considered in this paper 200 inner iterations per time step are taken. For displaying the lift polars in the unsteady computations the lift coefficient is averaged over the last 20 unsteady time steps (hence no averaging over the computed lift in the inner iteration). The amplitude in lift oscillation over these 20 unsteady time steps is displayed by means of an error bar. A large amplitude in lift-oscillation is then expressed by a larger error bar in the lift convergence plot. It should be mentioned beforehand that the 20 unsteady time steps taken is too low to measure low frequency lift oscillations. Hence, the error bar is basically a measure for high-frequency components. In a steady flow computations the lift is averaged over the last 200 multigrid iterations.

4 Numerical Results

4.1 Baseline Computation

In the baseline computation the steady flow solver is used to compute for the angles $\alpha_0, ..., \alpha_{10}$. For each angle a restart is performed from the preceding angle of attack. Per angle of attack 3000 multigrid cycles are taken (initial angle from scratch). In the baseline computation a 3w multigrid cycling strategy using lower-upper symmetric Gauss-Seidel implicit time integration scheme with a CFL number of 5.0 is adopted.

In Figure 2 it can be observed that maximum lift measured in an ETW wind tunnel campaign is underpredicted significantly in the baseline computation. The convergence history for the baseline computation is shown in Figure 2 as well. It can be observed that in the linear range, i.e. for angles $\alpha_0, ..., \alpha_2$, the lift coefficient stabilizes. For larger angles $\alpha_3, ..., \alpha_6$, there is an oscillatory behaviour and for $\alpha_7, ..., \alpha_{10}$ the computed lift has not converged yet and is still decreasing. This oscillatory and non-converging behaviour is experienced as unsatisfactory. In addition for the largest angles the computed lift is even decreasing leading to an even larger difference with experimental values.

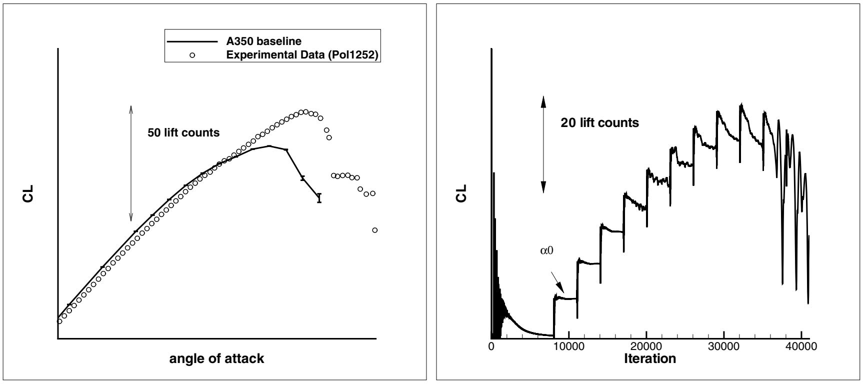

Fig. 2. Cross comparison of the lift computed in the baseline configuration and ETW experimental lift (top); convergence history of the lift-coefficient in the baseline computation (bottom)

4.2 Alternative Computational Strategies

Attempts are made with alternative computational strategies to stabilize lift and possibly enhance lift in the maximum lift range. To influence the convergence behaviour unsteady computations combined with steady computations are conducted. These are described here.

a) Alpha-Sweep with Unsteady Restart. An alpha sweep is performed starting in the linear range, similar as in the baseline computation. The idea here is to damp the oscillations in the convergence by restarting the steady flow computation with an unsteady computation at the same angle of attack. Firstly, by restarting from the computed solution at the preceding angle of attack, a limited number of 1000 3w multigrid cycles with the steady solver are taken. This is followed by 50 times steps using the unsteady solver. The time step size

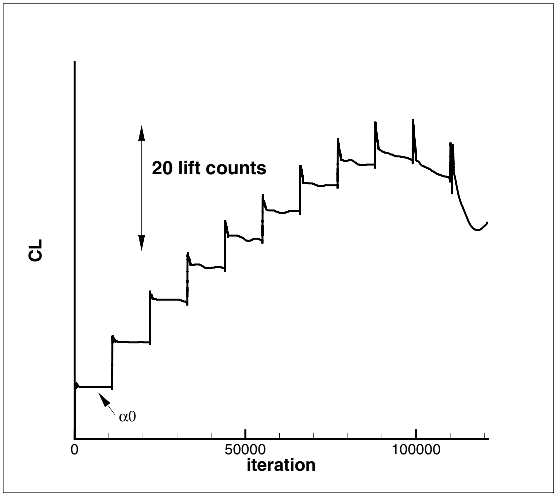

Fig. 3. Convergence history of the lift-coefficient for the computation where the steady flow computation is restarted with an unsteady flow computation for each angle of attack

is chosen as $\Delta t = 0.004$ which is approximately amounts to $\Delta t_{ref}/30$. This time step size has been found to be useful in other 3D high-lift computations, not reported here.

It can be observed in Figure 3 that the lift coefficient stabilizes and that the large oscillations in the flow computation are actually damped. As a matter a fact the oscillatory behaviour is damped for the angles $\alpha_3, ..., \alpha_9$. It can be observed in Figure 4 that computed lift is larger for the angles $\alpha_6, ..., \alpha_{10}$ compared to the baseline computation. Nevertheless, experimental lift is underpredicted still. It should be noted that the flow computation has not fully progressed to a converged lift-coefficient in the unsteady phase and more iterations/time steps are likely needed. In the linear range approximately the same lift is computed. For the largest angles of attack the lift oscillation in in the unsteady computation (as indicated by the error bars) is much smaller than the computed lift oscillation in the baseline computation.

b) Steady Flow Computation with 2v Multigrid Cycles. Near maximum lift flow separations occur which make the multigrid solution process inefficient. This can be expected basically while the coarse grid operator forms an inaccurate representation of the fine grid operator in case of flow separations. One way to circumvent this, is to employ a reduced number of multigrid levels. A common strategy is to switch to a 2v multigrid cycling strategy with the steady solver for the largest angles of attack.

In this computation a restart is made from the preceding angle of attack followed by 6000 multigrid 2-v cycles. The convergence history is shown in

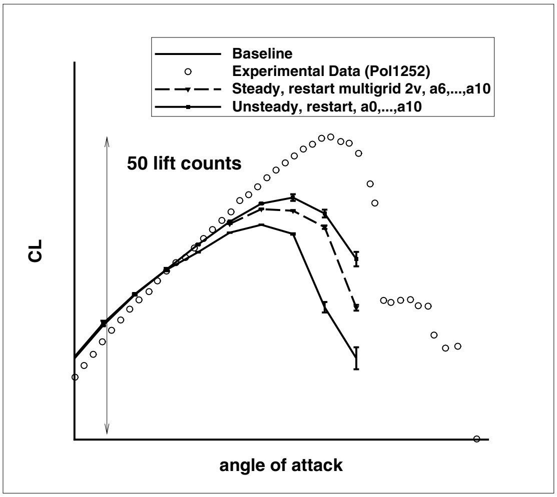

Fig. 4. Cross-comparison of lift coefficients computed in the baseline computation, the lift coefficients in the computation with unsteady-restart (after 1000 steady iterations) for angles$\alpha_0, ..., \alpha_{10}$, the lift coefficient computed in a steady computation using the 2v-multigrid strategy where restarts are made from the preceding angle of attack and ETW experimental lift.

Figure 5. It can observed that the computation stabilizes for $\alpha_6, ..., \alpha_9$ although the lift coefficients are not fully converged yet. For angle α_{10} large oscillations in computed lift are present.

Figure 4 shows a cross-comparison of the lift coefficients computed with the steady multigrid 2v-cycles compared to the computed lift in the baseline computation. It can be observed for the largest angles that a somewhat larger lift is computed compared to the computed lift in the the baseline configuration. Experimental lift is underpredicted. There is some concern that by taking a larger number of multigrid iterations the lift coefficient destabilizes. This will be addressed later.

c) Unsteady Restart from Computed Steady Solution. The computed baseline solutions at $\alpha_7, ..., \alpha_{10}$ are restarted with an unsteady computation with 400 time steps and a time step size of $\Delta t = 0.01$ which is approximately $\Delta t_{ref}/10$. The convergence histories for the unsteady computations at angles α_7 and α_8 are shown in Figure 6. Approximately, the same level of lift is computed. It can be observed for angle α_7 the lift coefficient is stabilizing and for α_8 the coefficient slowly varying with a small amplitude. For angles α_9 and α_{10} an oscillatory behaviour with a larger amplitude in lift is observed.

d) Start Unsteady Flow Computations from an Initial Flow Solution and Continue with 2v Multigrid. The reason for attempting an unsteady

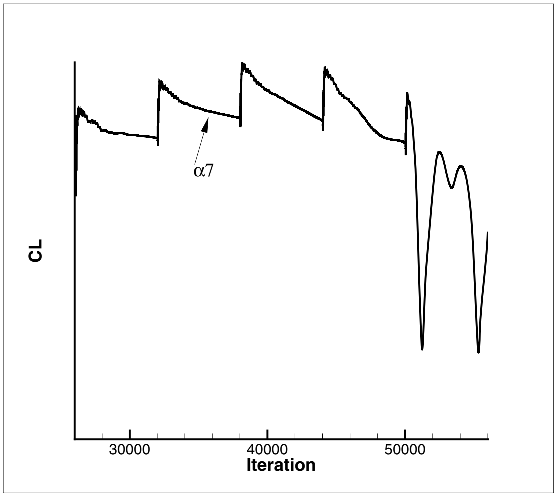

Fig. 5. Convergence history of the lift-coefficient for a steady flow computation using the multigrid 2v-cycling strategy. The steady flow computation is restarted from the baseline solution computed at angle α_5.

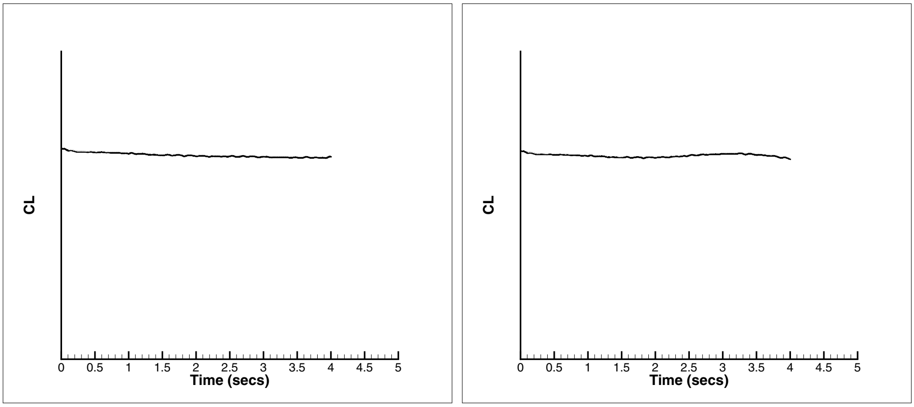

Fig. 6. Convergence history of the computed lift after restarting the baseline flow solution with an unsteady computation at angle α_7 and α_8

computation from an initial flow solution is that experience has learned that initialisation can have a large effect on the final computed solution. Common practice is the computational strategy pursued in the baseline computation (steady restart).

In this computation for each angle of attack an unsteady computation is performed starting from an initial solution by taking 50 time steps with $\Delta t =$ 0.01 (corresponds to 50 times 200=10000 multigrid iterations). Subsequently, 18000 2-v multigrid cycles are taken. Hence, in total 10000 unsteady plus 18000

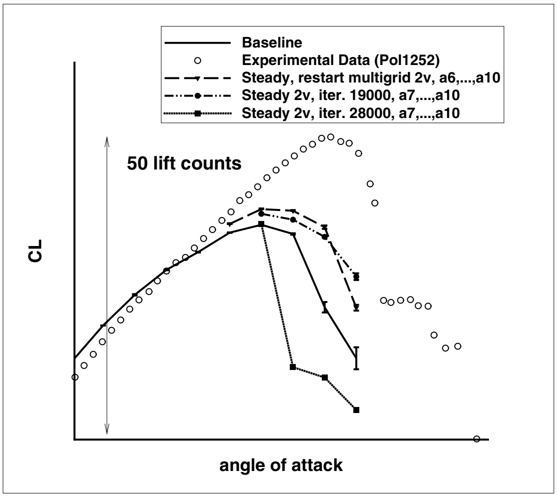

Fig. 7. Cross-comparison of the lift coefficients computed with steady 2v-multigrid after 19000 and 28000 iterations (with an unsteady computation from scratch, 10000 inner iterations, $\Delta t = 0.01$), the lift computed in the baseline computation, the lift computed in a steady 2v-multigrid computation where a restart is made from the preceding angle of attack and ETW experimental lift.

steady iterations are performed. A large number of multigrid 2v-cycles are taken in order to examine whether the flow solution is converging/stabilizing. It turns out that the lift coefficient is converging for angle α_7. But for larger angles the lift coefficients are no longer converging.

The computed lift coefficients in the steady phase after 19000 and 28000 iterations are shown in Figure 7. It can be observed that the computed lift after 28000 iterations is significantly reduced even to a level smaller than the computed lift in the baseline computation. In section 4.3 it is attempted to find an explanation for this behaviour.

e) Unsteady Flow Computations with a Start from an Initial Flow Solution. Finally, an unsteady computation is performed for the angles of attack $\alpha_7, ..., \alpha_{10}$ starting from an initial flow solution. In total 625 time steps are taken with unsteady time step sizes $\Delta t = 0.004$ which amounts to a simulated time of 2.5 secs.

The lift coefficient does not converge to a stationary value for the computed angles. For angles α_7 and α_8 lift is slowly steadily decreasing (non-converging) and for the larger angles α_9 and α_{10} an oscillatory behaviour in lift is observed.

The computed lift in the unsteady computation is compared to the baseline result in Figure 8. It can be observed that the lift reduces significantly from 1.2 secs to 2.5 secs. This will be examined more closely in Section 4.3 "Extended post-processing". It should be mentioned that the flow solution for α_9 actually

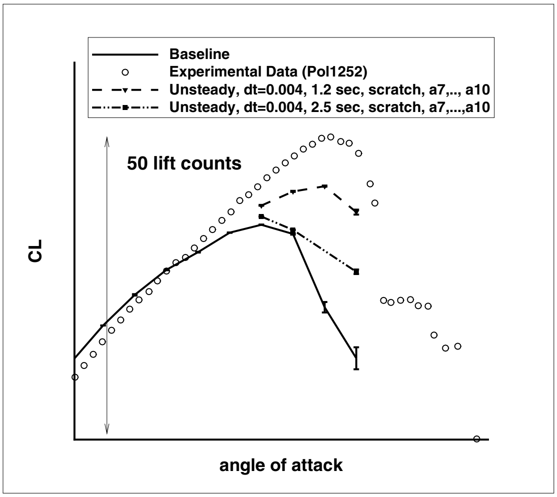

Fig. 8. Cross-comparison of lift coefficients computed in an unsteady computation with $\Delta t = 0.004$ at time 1.2 secs and 2.5 secs, the lift coefficients computation in the baseline computation and ETW experimental lift

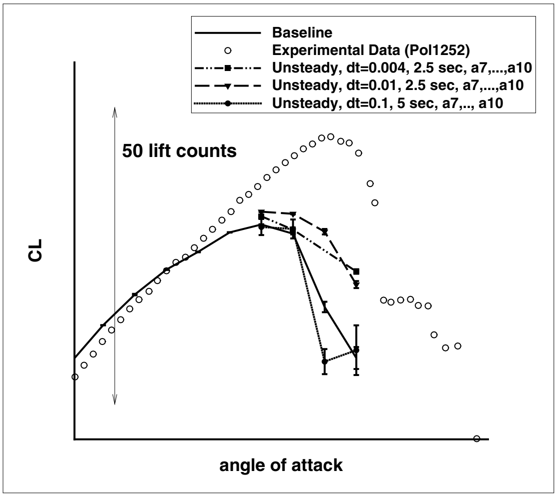

Fig. 9. Cross-comparison of the lift coefficients computed in three unsteady computations with time step sizes $\Delta t = 0.004$, $\Delta t = 0.01$ and $\Delta t = 0.1$ (unsteady computation from an initial solution), the lift coefficients computed in the baseline computation and ETW experimental lift

crashes at t=2.45 secs. A reason for the crash may be that the number of inner iterations is chosen too small. A check of the convergence history learns that in 200 iterations the residual drops about 0.5 orders. Another possible explanation is that the unsteady extrapolation order is too large (second-order in this study). The extrapolation might lead to an unphysical start solution.

As a second step the time step size in the unsteady computation is varied. It is necessary to check whether the computed flow solution depends on the time step size. Therefore, three different time step sizes are compared. The time step size is increased to $\Delta t = 0.01$ (250 time steps) and $\Delta t = 0.1$ (50 time steps). The results are shown in Figure 9. It can be observed in the unsteady computations for the angles α_7, α_8 and α_9 that the computed lift with $\Delta t = 0.01$ is somewhat larger than the lift with $\Delta t = 0.004$. Obviously, a change in time step has only a small effect on computed lift. It can be observed as well that for a larger simulated time, 0.5 secs, using $\Delta t = 0.1$ the computed lift is at approximately the same level as the lift computed in the baseline computation.

4.3 Extended Post-Processing

In the steady and unsteady flow computations it is found that the lift breaks down for the largest angles $\alpha_8, ..., \alpha_{10}$. In the computational strategies considered before it was not possible to achieve a better agreement with the experimental lift in the maximum lift range. Although, in some (unsteady) computations it appeared that lift is improved in the maximum lift range similar as in the steady computation. Additional unsteady time steps, nevertheless, lead to a break down in lift. In this section an attempt is made to find an explanation for the break down in lift.

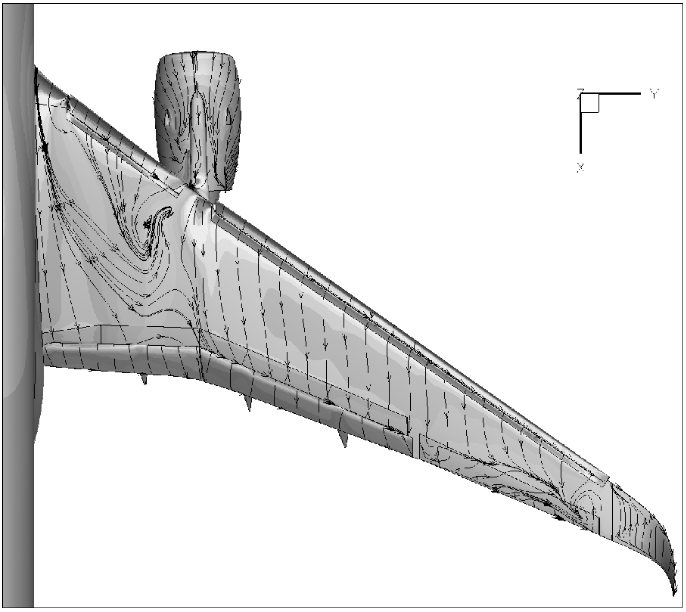

Fig. 10. Computed pressure distribution and skin friction lines for angle of attack α_8

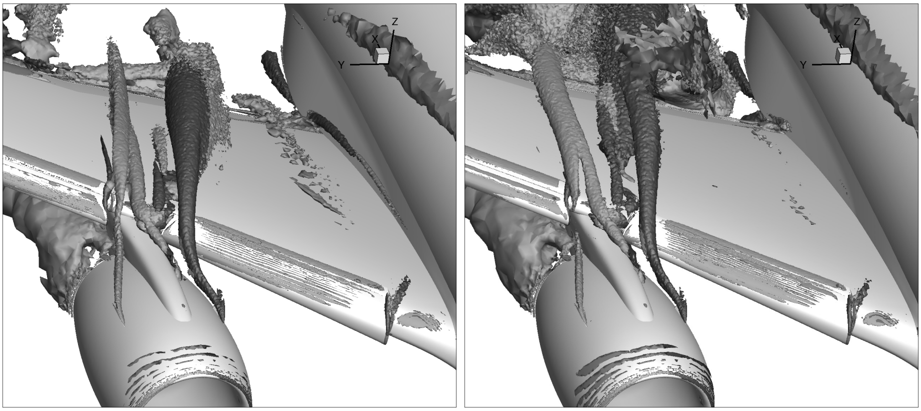

Fig. 11. The computed kinematic vorticity number shaded by x-vorticity 19000 iterations (left) and 28000 iterations (right). Isoplots of the kinematic vorticity number are shown at value 1.3. Dark shaded vortices indicates anti-clockwise rotation.

The computed pressure coefficient distribution and skin friction lines for the baseline computation is shown in Figure 10 for angle of attack α_8. For this angle of attack the flow is separated on the upper inboard wing, but also outboard on the upper wing near the aileron. It is verified that in the other computations, basically the same flow characteristcs occur, but with local differences in flow topology.

The computed flow solution with the steady solver using 2v multigrid cycling (described in section 6) is shown in Figure 11. In these figures isoplots of the kinematic vorticity number (i.e. a measure for the rotational stress divided by the shear stress) is shown which is shaded by x-vorticity to distinguish between rotation directions. These figures mainly show the same vortical structures. When focussing in more detail on the steady solver flow solutions firstly, see Figure 11, it can observed that a large difference occurs in the wake of the inner strake vortex. It appears that the vortex starting from the area between the deployed droop nose device and the pylon breaks down. This can for instance be seen in the right picture of Figure 11. The wake of this vortex has broken up and this wake interacts with the inner strake vortex. One possible explanation might be that the vortex has an unfavourable interaction with the prismatic/tetrahedral grid interface, so that it diffuses and breaks down. Another possible explanation might be that the inner strake vortex breaks down (above the flap). One way to avoid this would be to make the computational grid in this area coarser, hence introduce vortex break down artificially.

A second observation is that the vortex emanating from the pylon leading edge seems to hit the wing leading edge indicated by a small local area of vortex break down. The vortex reattaches and continues its way over the main wing element. Furthermore, it can be observed in Figure 11 that the vortices are not distinctly captured at the at the main wing leading edge between pylon and droop nose device. This may be improved by introducing a local grid refinement.

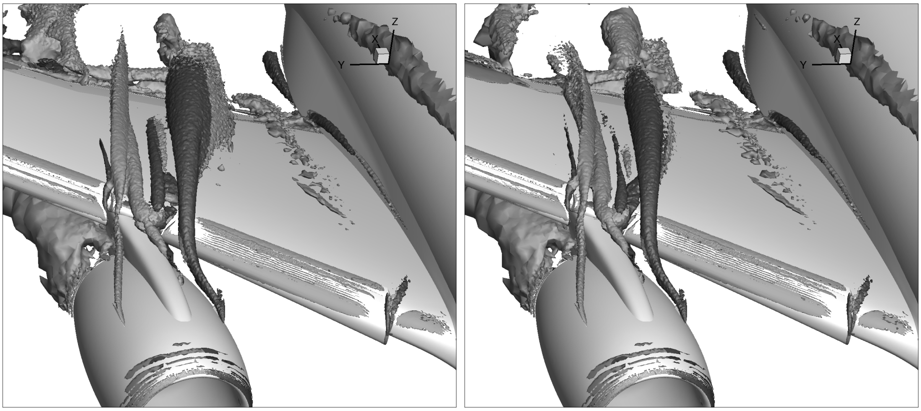

Fig. 12. The computed kinematic vorticity number shaded by x-vorticity at t=1.2 sec (left) and t=2.5 sec (right). Isoplots of the kinematic vorticity number are shown at value 1.3. Dark shaded vortices indicates anti-clockwise rotation.

The computed flow solutions of the unsteady flow solution (described in section 5 with $\Delta t = 0.004$) at t=1.2 sec and 2.5 sec are compared in Figure 12. The computed flow solution at t=2.5 sec resembles the flow solution computed with the steady solver using the 2v multigrid strategy after 19000 iterations. In general one can say that the vortical structures are represented somewhat more clearly.

The influence of the prismatic/tetrahedral grid interface on the main wing can be observed in Figure 12. The vortex that emanates from the main wing leading edge between the pylon and droop nose device changes location and transports an area of clock-wise vorticity (near the main wing under the inner strake vortex) downstream. The occurrence of this flow feature may be exaggerated by the grid interface.

5 Conclusions

A number of alternative computational strategies, both steady and unsteady flow computations, are investigated. It is concluded, that an unsteady computation with a relatively large time can stabilise the aerodynamic coefficients during convergence. This confirms the presumption that by taking a relatively large time step the small scale frequency components are damped.

It is also shown that the same effect can be realised with a steady 2v multigrid cycling strategy. Nevertheless, by taking a large number of 2v multigrid cycles the convergence history becomes oscillatory as well and lift breaks down similar as in the baseline computation. These steady results indicate that the flow solution is inherently unsteady.

In the computational strategies considered in this paper it was not possible to achieve a better agreement with the experimental lift in the maximum lift range. Although, in some (unsteady) computations it appeared that lift was improved in the maximum lift range, additional unsteady time steps would lead to a lift break down, similar as in the steady computation. Several attempts are made to compute with an unsteady solver with different time step sizes. In these unsteady computations basically the same flow physics is found. This leads to the belief that in order to improve the prediction of maximum lift either the computational grid needs to be improved or the flow modelling (based on Spalart-Allmaras turbulence model here) is insufficient.

A necessity for investigating/improving the high-lift flow computation is the possibility to locally adapt/improve a computational mesh. This would allow to locally manipulate/enhance vortical flow structures and to examine their influence on the flow without completely changing the computational grid. This is not possible at the moment with the current grid generation capability. Some suggestions for improving in computational grid size can be proposed, such as refine the grid in the area of the wing leading edge/pylon intersection to capture the vortices emanating at pylon, nacelle and droop nose device edge; make the computational grid trailing edge coarser to fix vortex break down of strake vortices.

One idea might be then to locally introduce a structured Chimera grid on the main wing, ranging from fuselage to approximately the spanwise position outbaord of the engine pylon. This would allow to more accurately capture vorticity and would remove the unfavourable influence of the prismatic/tetrahedral grid interface.

An even better solution will be achieved vie the so-called Hyperflex grid approach, in which it becomes possible to run the solver on a mixed structured-unstructured mesh. This approach could provide ideal mesh configurations to resolve the physical phenomena and to avoid unwanted mesh metric discontinuity influence on the computed flow.

Concerning enhanced flow modelling it would be a possibility to switch to a more complex turbulence like Menter-SST and/or a Reynolds stress model (in an industrialisation phase). The advantage of using RSM over Menter-SST is that vortices are better represented, since off-diagonal stress components are modelled. Nevertheless, even an RSM turbulence model is insufficient to model vortex-break down. As an engineering approach this still needs to be handled by choosing appropriate (coarse) grid resolution.

References

1. Eliasson, P., Marongiu, C., Mikhaylov, S., Vlasenko, V., Bosnyakov, S., Kazhan, E.: Acceleration of URANS for application to separated high-lift flows. In: Eberhardsteiner, J., et al. (eds.) European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS 2012), Vienna, Austria, September 10-14 (2012)
2. Acisu, I.: Maximum lift computations using unsteady RANS with a Reynolds Stress Model. Presented at the ComFlite final Workshop (June 2012)

3. Knobloch, O.: Untersuchung der Fahigkeiten von URANS, verbesserte ablösenahe und abgelöste Strömungen im Bereich des Maximalauftriebs vorherzusagen, milestone report 2.1.3 (November 2010)
4. Schwamborn, D., Gerhold, T., Heinrich, R.: The DLR Tau-Code: Recent Applications in Research and Industry. In: Wesseling, P., Oñate, E., Périaux, J. (eds.) Proceedings of the European Conference on Computational Fluid Dynamics ECCOMAS CDF 2006, The Netherlands (2006)
5. Internet-site, `www.centaursoft.com`