TURBULENT FLOW MODELLING USING EARSM ON PARALLEL COMPUTERS

MATTIAS SILLÉN

Saab Aerospace,
SE-581 88 Linköping, Sweden
Mattias.Sillen@saab.se

(Received 18 January 2001)

Abstract: The compressible Navier-Stokes equations are solved numerically for turbulent transonic aerospace applications on parallel computers. The turbulence is modelled by an Explicit Algebraic Reynolds Stress Model (EARSM). Expressing the EARSM as an extension of an eddy-viscosity model makes the implementation straightforward in a flow solver with existing two-equation eddy-viscosity models. The k-ω transport equations are used as a platform for the model. The EARSM approach significantly improves the shock position for transonic flow over wings without substantial increase in computational cost. Industrial use of advanced flow modelling requires a short turn-around time of computations. This is enabled through the use of parallel computers. To achieve good parallel performance the computational load has to be evenly distributed between the processors of the parallel machine. A heuristic algorithm is described for distributing and splitting the blocks of a structured multiblock grid for a good static load balance. Speed-up results are presented for turbulent flow around a wing on a number of parallel platforms.

Keywords: aviation, Computational Fluid Dynamics, aerodynamics of airplanes

1. Introduction

In the aerospace industry today Computational Fluid Dynamics (CFD) plays an increasingly important role as a tool for design and analysis. Typical characteristics for the industrial CFD environment are the often complicated geometry, the frequency of large problems, the requirement of high fidelity results, the short time schedule for producing results, *etc*. A variety of methods is used from preliminary design studies, where short problem turn-around time is crucial, to detailed component analysis where accurate results are required. To further improve the confidence of the predictions in the aerodynamic design process, advanced CFD methods, *i.e.* Reynolds Average Navier-Stokes (RANS) based 3D methods, must be deployed at an even earlier stage than today. Early in the design process when the geometry is rapidly changing, the allowable time frame to produce a complete flow analysis, is no more than a day. To be able to meet this goal the CFD program execution time has to be less than 15 hours, *i.e.* overnight. This allows for evaluation of the results and preparation of an improved design during the working day. A cost-efficient way to reduce the problem turn-around time is to use parallel computers and efficient numerical algorithms to solve the flow equations. To achieve good parallel performance of a CFD code the computational load has to be evenly distributed between the processors of a parallel machine. An algorithm has been described here to show how to partition structured multiblock grids for efficient use of parallel computers.

Until recently the prevailing *industrial* state-of-the-art CFD methods [1] included viscous coupled Euler solvers or RANS-methods with algebraic turbulence closures. To be able to deal with complex flows and obtain results with a higher degree of confidence more advanced methods are required. Methods based on RANS equations with two-equation turbulence models or higher have been around in the research community for a long time but just recently entered the industrial aerodynamic design process. Explicit Algebraic Reynolds Stress Model (EARSM) has emerged as an industrially feasible class of turbulence models offering improved physical modelling compared to linear eddy-viscosity models without being as computationally expansive as the more complete Reynolds stress models.

The present article describes recent development in turbulence modelling and parallel implementation in one of the RANS methods used for aerodynamic design and analysis at Saab Aerospace. In Section 2 the turbulence models are described with an emphasis on the EARSM approach. The numerical implementation is discussed in the following section. The parallel load balancing strategy is described in Section 4. Numerical results showing the benefit of using EARSM compared to a standard two-equation eddy-viscosity model for a number of transonic wings conclude the paper, where also speed-up results are presented for different parallel platforms.

2. Turbulence modelling

The compressible Navier-Stokes equations model the flow of air around aeroplanes very well. For such applications the influence of turbulence must be modelled, because of insufficient memory capacity and computational speed of present computers to solve the turbulent flow equations directly. From an industrial point of view, a suitable compromise between robustness, efficiency and validity for turbulence closure has often been found in the class of two-equation eddy-viscosity models represented by *e.g.* the k-ε or the k-ω models. A frequently used model in the aerospace industry is the Wilcox k-ω [2] model. Primarily due to its numerical robustness and that it does not need any wall-distance information, which can be cumbersome to compute for complex three-dimensional configurations. In the Wilcox standard k-ω model the transport equations read:

$$\frac{D}{Dt}(\rho k) = P - \rho\varepsilon + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j}\right] \tag{1}$$

$$\frac{D}{Dt}(\rho\omega) = \alpha\frac{\omega}{k}P - \beta\rho\omega^2 + \frac{\partial}{\partial x_j}\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\frac{\partial \omega}{\partial x_j}\right] \tag{2}$$

where k is the turbulent kinetic energy, ω is the specific dissipation rate and $\varepsilon \equiv \beta^*\omega k$ is the dissipation rate. The coefficients $\alpha = 5/9$, $\beta = 3/40$, $\beta^* = 9/100$, $\sigma_k = 2.0$ and $\sigma_\omega = 2.0$ are the model constants.

The production of turbulent kinetic energy, P, is defined as:

$$P \equiv -\overline{\rho u_i u_j}\frac{\partial U_i}{\partial x_j} \tag{3}$$

In a two-equation eddy-viscosity model the Reynolds stresses are modelled as:

$$\overline{\rho u_i u_j} = \frac{2}{3}\rho k \delta_{ij} - 2\mu_t S^*_{ij} \tag{4}$$

where

$$S^*_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3}\frac{\partial U_k}{\partial x_k}\delta_{ij}\right) \tag{5}$$

and the eddy-viscosity is defined as:

$$\mu_t = \frac{\rho k}{\omega} \tag{6}$$

One often noticed drawback of this model is its free-stream dependency. A recent proposal by Kok, *cf.* [3], resolves this effectively by introducing a diffusion correction that has negligible additional cost and is free of inconvenient parameters such as local wall-distances. A cross-diffusion term is added to the ω transport equation, Equation (2):

$$C_D = \sigma_d \frac{\rho}{\omega}\max\left\{\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i}, 0\right\} \tag{7}$$

$\sigma_d = 0.5$ is an additional model constant. The value of the model parameter σ_k is here modified to 1.5.

For more demanding flow cases where flow separation or streamline curvature is present the standard two-equation models often fail to produce accurate results. This is often due to the linear stress/strain relationship. A promising way to introduce enhanced physics with a computational cost comparable to the existing two-equation models is to use the non-linear stress/strain relationship offered by an EARSM. The EARSM proposed by Wallin and Johansson [4] is derived from a differential Reynolds Stress Model by Launder, Reece and Rodi [5]. A self-consistent and fully explicit algebraic relation for the Reynolds stresses in terms of the mean flow field is obtained by applying the so-called 'equilibrium assumption', where the advection and diffusion of the Reynolds stress anisotropy are neglected. The model is based on the k-ω transport equations as given in Equations (1) and (2).

In the EARSM the Reynolds stresses may be written in terms of an effective turbulent viscosity μ_t and an extra anisotropy $a^{(ex)}_{ij}$:

$$\overline{\rho u_i u_j} = \frac{2}{3}\rho k \delta_{ij} - 2\mu_t S^*_{ij} + \rho k a^{(ex)}_{ij} \tag{8}$$

where the effective turbulent viscosity is:

$$\mu_t = -\frac{1}{2}(\beta_1 + II_\Omega \beta_6)\rho k \tau \tag{9}$$

and the extra anisotropy becomes:

$$\begin{aligned} a^{(ex)} = {} & \beta_3\left(\Omega^2 - \frac{1}{3}II_\Omega \mathrm{I}\right) + \beta_4(S\Omega - \Omega S) + \beta_6\left(S\Omega^2 + \Omega^2 S - II_\Omega S - \frac{2}{3}IV\mathrm{I}\right) + \\ & \beta_9\left(\Omega S \Omega^2 - \Omega^2 S \Omega\right) \end{aligned} \tag{10}$$

The invariants are defined by:

$$II_S = tr\{S^2\}, \quad II_\Omega = tr\{\Omega^2\}, \quad IV = tr\{S\Omega^2\} \tag{11}$$

Here a, S and Ω denote second rank tensors, $tr\{\}$ denotes the trace and I is the identity matrix. The inner product of two matrices is defined as $(SS)_{ij} \equiv (S^2)_{ij} \equiv S_{ik}S_{kj}$. The normalised mean strain- and rotation rate tensors are defined as:

$$S_{ij} = \tau S^*_{ij}, \quad \Omega_{ij} = \tau \Omega^*_{ij} \tag{12}$$

where the turbulent time-scale is defined by:

$$\tau = \max\left(\frac{k}{\varepsilon}, C_\tau \sqrt{\frac{\mu}{\rho\varepsilon}}\right) \tag{13}$$

and $C_\tau = 6.0$ is a model constant.

The dimensional strain- and rotation rate tensors are:

$$S^*_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} - \frac{2}{3}\frac{\partial U_k}{\partial x_k}\delta_{ij}\right) \quad \text{and} \quad \Omega^*_{ij} = \frac{1}{2}\left(\frac{\partial U_i}{\partial x_j} - \frac{\partial U_j}{\partial x_i}\right) \tag{14}$$

The β-coefficients are given by:

$$\beta_1 = -\frac{N(2N^2 - 7II_\Omega)}{Q}, \quad \beta_3 = -\frac{12N^{-1}IV}{Q}, \quad \beta_4 = -\frac{2\left(N^2 - 2II_\Omega\right)}{Q}, \quad \beta_6 = -\frac{6N}{Q}, \quad \beta_9 = \frac{6}{Q} \tag{15}$$

with the non-singular denominator

$$Q = \frac{5}{6}\left(N^2 - 2II_\Omega\right)\left(2N^2 - II_\Omega\right) \tag{16}$$

N is given by:

$$N = \begin{cases} \dfrac{c_1'}{3} + \left(P_1 + \sqrt{P_2}\right)^{1/3} + sign\left(P_1 - \sqrt{P_2}\right)\left|P_1 - \sqrt{P_2}\right|^{1/3}, & P_2 \geq 0 \\[2ex] \dfrac{c_1'}{3} + 2\left(P_1^2 - P_2\right)^{1/6}\cos\left(\dfrac{1}{3}\arccos\left(\dfrac{P_1}{\sqrt{P_1^2 - P_2}}\right)\right), & P_2 < 0 \end{cases} \tag{17}$$

with

$$P_1 = \left(\frac{c_1'^2}{27} + \frac{9}{20}II_S - \frac{2}{3}II_\Omega\right)c_1', \quad P_2 = P_1^2 - \left(\frac{c_1'^2}{9} + \frac{9}{10}II_S + \frac{2}{3}II_\Omega\right)^3 \tag{18}$$

and

$$c_1' = \frac{9}{4}(c_1 - 1) \tag{19}$$

$c_1 = 1.8$ is a model constant.

3. Numerical implementation

The Navier-Stokes equations are a non-linear system of time-dependent partial differential equations to solve for the density ρ, the momentum components in the three Cartesian co-ordinate directions ρu_i, $i = 1,2,3$, and the total internal energy ρE in the field. The flow equations are approximated on a structured grid with a finite volume approximation using central differencing. A blend of second and fourth order artificial viscosity is added to avoid oscillatory behaviour in the smooth parts and in the vicinity of shocks [6, 7]. The flow equations are integrated forward in time by a Runge-Kutta method until the time derivatives are sufficiently small and we are close to a steady state solution. The convergence to steady state is accelerated by a multi-grid method, where the solutions on a sequence of coarser grids are combined to improve the convergence rate.

The numerical implementation of the k-ω transport equations is implicit, solving the steady transport equations by ADI technique. One relaxation sweep is performed in each direction after every complete Runge-Kutta cycle for the flow equations. The spatial derivatives are evaluated using the same finite volume technique as the flow equations but with a hybrid central/upwind differencing.

The Reynolds stresses expressed by the EARSM are written in terms of an effective turbulent viscosity μ_t and an extra anisotropy $a_{ij}^{(ex)}$. The reason for that is that most flow solvers with two-equation eddy-viscosity models can reuse subroutines when the model is formulated using an effective turbulent viscosity. It has shown that it is sufficient to only consider the effective turbulent viscosity when determining the stability limit for the flow equations. The extra terms associated with the extra anisotropy can be treated as a fully decoupled source term, see also [8].

The production of the turbulent kinetic energy in a two-equation eddy-viscosity model, $P^{(EVM)}$, is expressed by using the assumption in Equation (4):

$$P^{(EVM)} = \left(2\mu_t S_{ij}^* - \frac{2}{3}\rho k \delta_{ij}\right)\frac{\partial U_i}{\partial x_j} \tag{20}$$

Now, using the effective turbulent viscosity defined in (9) in relation (4) and by adding the contribution from the extra anisotropy the production of the turbulent kinetic energy in the EARSM is written as:

$$P = P^{(EVM)} - \rho k a_{ij}^{(ex)}\frac{\partial U_i}{\partial x_j} \tag{21}$$

This term is the most important to modify to include effects of streamline curvature and local and global rotation. The second most important term to modify is the turbulent transport of momentum in the momentum equation. In a two-equation eddy-viscosity model the term is modelled as:

$$T_i^{(EVM)} = \frac{\partial}{\partial x_j}\left(2\mu_t S_{ij}^* - \frac{2}{3}\rho k \delta_{ij}\right) \tag{22}$$

In the EARSM a more physical sound modelling of this term is:

$$T_i = T_i^{(EVM)} - \frac{\partial}{\partial x_j}\left(\rho k a_{ij}^{(ex)}\right) \tag{23}$$

using the effective turbulent viscosity in (9).

An alternative to EARSM based on standard k-ω, that is even simpler to implement, is the so-called linear EARSM. It is actually an eddy-viscosity model where only the eddy-viscosity part of relation (8) is kept. This means that $a_{ij}^{(ex)} = 0$. Some of the physics covered of the full EARSM is naturally lost in this approach. It is however very easily introduced into an existing two-equation eddy-viscosity model where only the routine that computes the eddy-viscosity needs to be modified. Compared to a standard eddy-viscosity model with a constant C_μ it is based on a sounder basis.

The same physical boundary conditions are used for k and ω in the EARSM as in the two-equation implementation. The physical conditions are:

$$k=0 \quad \text{and} \quad \omega \to \frac{6\mu}{\beta\rho y^2} \tag{24}$$

For the applications so far the EARSM has shown the same numerical behaviour as the standard Wilcox k-ω and the computational cost is approximately 8% higher than for the standard Wilcox.

4. Parallel implementation and load balancing

The code described here was originally developed for the Cray vector-computers. The effort to parallelize it was modest because of the multiblock structure already introduced in the serial version. A master-slave model is used where a master process initialises and supervises the iterations. Each slave process is responsible for a subset of the blocks. The communication is implemented using PVM for portability between different platforms.

The difference stencil at a block boundary needs solution data from the neighbouring block on the other side of the boundary. These data are sent over the network if the neighbouring block is located on another processor or moved internally in the memory if the processor also takes care of the neighbour. Updated boundary data are transferred between neighbours at every stage or at the end of a full Runge-Kutta step. This is also the case when multiple grids are used.

Efficiency is of particular importance in parallel computing. Multi-grid acceleration can improve the convergence rate and lower the number of iterations substantially in both serial and parallel mode. In our parallel implementation the same processor handles the same blocks on all grid levels. In this way, all processors are kept busy also on the coarsest grid, even if the quotient between useful computing time and communication time decreases. The load balance is critical to good performance. A usual assumption is that the grid can be partitioned arbitrarily. This is not the case here, as we have a block structure from the beginning created by a grid generator. The original partitioning of the grid into blocks is assumed to be determined only by the topology of the configuration. The blocks are distributed based on an analysis of the computational and communication time before the iterations start for optimal use of the computer. If an even load is not obtained with the original blocks, then they are split until a convergence criterion is satisfied.

The CPU time for a time-step on a processor is proportional to the number of grid cells in the blocks on the processor. The time of transferring block boundary data from one processor to another depends on the number of boundary cells the blocks on the processors have in common. An estimate of the total time t for a time-step with a given partition of N_{block} blocks on N_{proc} processors, Ψ, is:

$$t(\Psi)=\max_{k=1;N_{proc}}(t_k^{CPU})+\sum_{n=1}^{N_{proc}} t_n^{comm} \tag{25}$$

where t_k^{CPU} is an estimate of the CPU-time for a time-step for the blocks on the processor k. The CPU-time is based on the number of floating point operations per cell and the processor's speed. Accurate estimates of the number of floating point operations per cell are available for different equations. The communication time t_k^{comm} for sending data from one processor k to all other processors is based on the number of cells the processors have in common, the number of variables to communicate and the bandwidth of the shared network. A more detailed analysis of the time estimate is found in [9].

The algorithm tries to find the distribution Ψ with the minimum total execution time $t(\Psi)$. The N_{block} blocks in the grid are distributed into N_{proc} subsets of blocks in a pre-processing step. The subsets are determined statically.

1. Sort the blocks in decreasing order according to the execution time t_i, $i=1,\dots, N_{block}$ assuming that the problem is solved on a parallel computer with N_{block} identical processors processing one block on each processor.
2. Start with empty subsets.
3. Then each block is taken from the sorted set and inserted into the subset that gives the smallest value of t for the partitioning built up so far.

Sometimes it is not possible to obtain a good balance with the original sizes of the blocks. Then it is necessary to split some of the blocks and re-compute the load distribution to see if a more satisfactory solution has been found. To keep the quotient between computation time and communication time large, we should split as few blocks as possible. When a block has been selected for splitting it is divided in the middle of the longest edge, so the two new blocks introduce as little extra communication as possible.

There is no guarantee that this heuristic algorithm above finds the optimal solution. It will however produce an acceptable load balance even though based on a simple communication model with the advantage of being computationally inexpensive.

5. Numerical examples

In this section there are a number of industrially relevant examples of viscous compressible flow problems presented where the benefit of using advanced turbulence models is clearly demonstrated. The examples cover a 2D airfoil, a 3D wing and a complete Airbus configuration in transonic flow. The advantage of using parallel computers to achieve a short problem turn-around time is exemplified with viscous flow around the ONERA M6 wing.

5.1. RAE 2822 airfoil

The two-dimensional airfoil RAE 2822 in transonic flow conditions is a typical example of a demanding aerospace application. The flow condition is referred to as Case 10, *i.e.* $M_\infty = 0.754$, $\alpha = 2.57^\circ$ and $Re_c = 6.2\cdot 10^6$. The airfoil is highly loaded and a boundary layer separation occurs at the shock impingement point. The experimental data are from Cock *et al.* [10]. The computational mesh consists of 272 cells in the stream-wise direction and 80 in the normal direction, see Figure 1. Transition is prescribed at 3% of the chord.

In Figure 2 there are the pressure coefficient distributions presented for computations using different turbulence models. The applied models include Wilcox k-ω, k-ω with cross-diffusion according to Kok, EARSM based on Wilcox, EARSM based on Kok and linear EARSM based on Wilcox. The different EARSM perform significantly better than the linear eddy-viscosity k-ω models, even though there is still a small mismatch in shock location compared to the measurements. The results produced by the linear EARSM fall in between the full EARSM and the k-ω results. There are other computations published on this case that have better agreements. The reason for this is that there exists a number of different geometries for this case, the measured or design geometry with or without an additional camber correction. In this case the measured geometry is used with the camber correction. Figures 3 and 4 give the velocity profiles computed with the different turbulence models compared to measurements. At location $x/c = 0.404$, *i.e.* upstream of the shock, the different models predict almost the same velocity profile. Downstream of the shock, at $x/c = 0.900$, the scatter is greater but the more advanced models compare better with measurements.

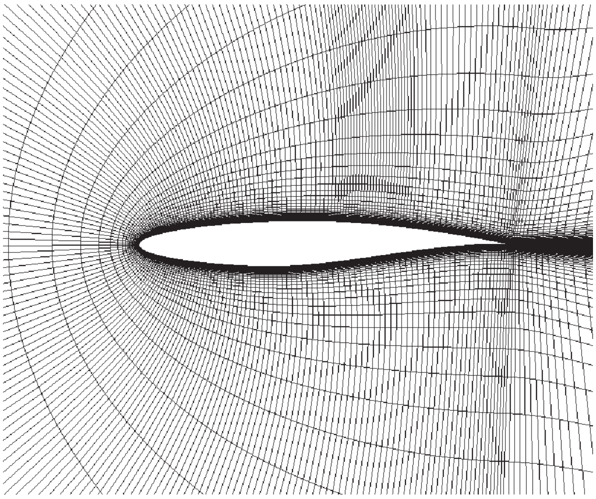

Figure 1. Computational grid around RAE 2822 airfoil

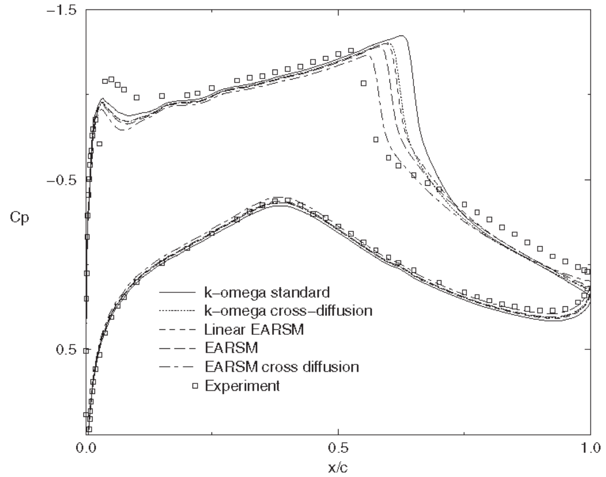

Figure 2. C_p distribution computed with different turbulence models on RAE 2822 compared with measurements. Flow case $M_\infty = 0.754$, $\alpha = 2.57°$ and $Re_c = 6.2 \cdot 10^6$

5.2. LANN wing

The LANN wing is a supercritical transport wing, which was measured at transonic wind tunnel conditions by Horsten *et al.* [11]. The computational mesh consists of 192 cells in the stream-wise direction, 48 cells in the span-wise direction and 64 in the normal

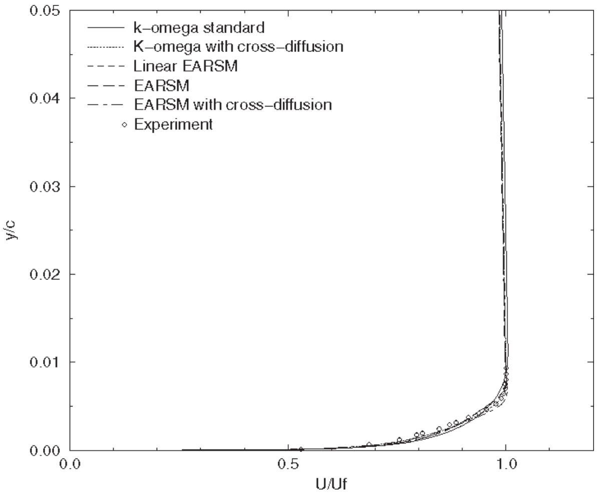

Figure 3. Velocity profile at $x/c = 0.404$ computed with different turbulence models compared with measurements. Flow case $M_\infty = 0.754$, $\alpha = 2.57°$ and $Re_c = 6.2 \cdot 10^6$

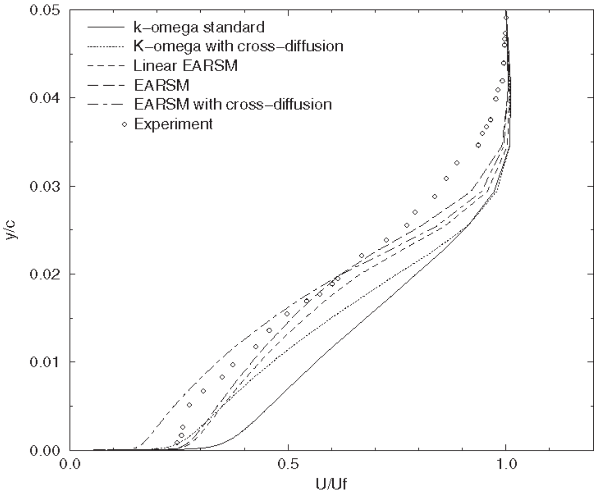

Figure 4. Velocity profile at $x/c = 0.900$ computed with different turbulence models compared with measurements. Flow case $M_\infty = 0.754$, $\alpha = 2.57°$ and $Re_c = 6.2 \cdot 10^6$

direction. The wing surface is discretised by 128×32 cells. The on-flow conditions are $M_\infty = 0.82$, $\alpha = 2.60°$ and $Re_{mac} = 5.32 \cdot 10^6$.

The flow is characterised by a strong shock followed by a boundary layer separation on the upper surface, see computational results in Figure 5.

The EARSM predicts a larger separated flow region compared to the k-ω model. The computed pressure distribution is dramatically improved by using the EARSM instead of the k-ω model as seen in Figures 6 and 7. The computed pressure distributions are compared with measurements at 47.5% and 82.5% of the span. The EARSM based on the k-ω model with the cross diffusion term performs very well, predicting the same shock location as in the measurements.

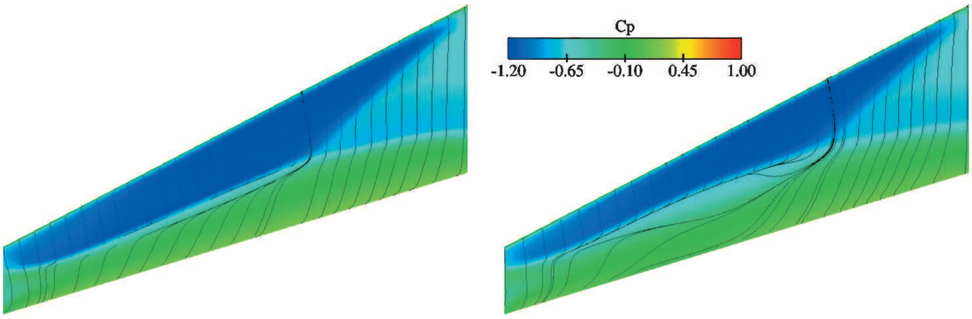

Figure 5. Surface pressure distributions and skin friction lines on upper side of the LANN wing as computed by the Wilcox *k*-*ω* model (left) and by the EARSM based on *k*-*ω* with cross-diffusion by Kok (right). The on-flow conditions are $M_\infty = 0.82$, $\alpha = 2.60°$ and $Re_{mac} = 5.32 \cdot 10^6$

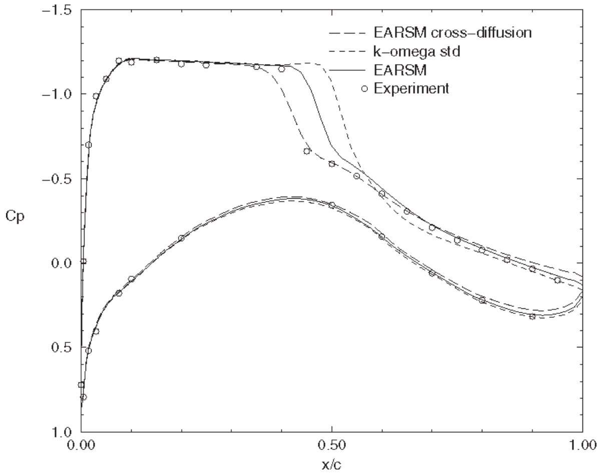

Figure 6. Pressure distribution at $y/b = 47.5\%$ as computed with different turbulence models compared with measurements

5.3. Airbus-like configuration

The Airbus-like AS28 wing-body-pylon-nacelle configuration is a demanding example with respect to the geometrical complexity and computational mesh size. It is a typical example of an industrial aerospace application.

The AS28 configuration is a wind tunnel tested by Aerospatiale in different configurations. A structured multiblock grid is generated with a total of 270 blocks and almost 5 million cells. The considered flow case is defined by $M_\infty = 0.80$, $\alpha = 2.2°$ and $Re_{mac} = 9.97 \cdot 10^6$. The nacelle has through-flow conditions. The surface pressure distribution as given by the EARSM is presented in Figure 8. On the starboard wing the surface streamlines are plotted.

Computational results obtained with standard Wilcox *k*-*ω* are compared with results produced by EARSM based on *k*-*ω* with cross-diffusion according to Kok and wind tunnel measurements at 17% and 57% of the span, see Figures 9 and 10. The pressure distributions predicted by the different turbulence models compare very well on the lower part of the wing. On the upper side the shock position is more accurately predicted by the EARSM while the standard *k*-*ω* model predicts a too far down-stream position. The pressure at the

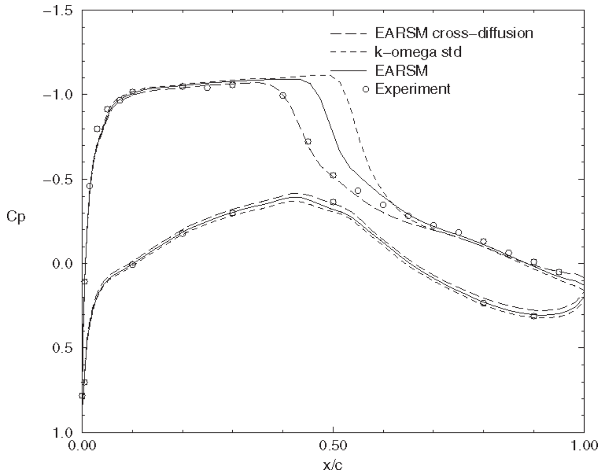

Figure 7. Pressure distribution at $y/b = 82.5\%$ as computed with different turbulence models compared with measurements

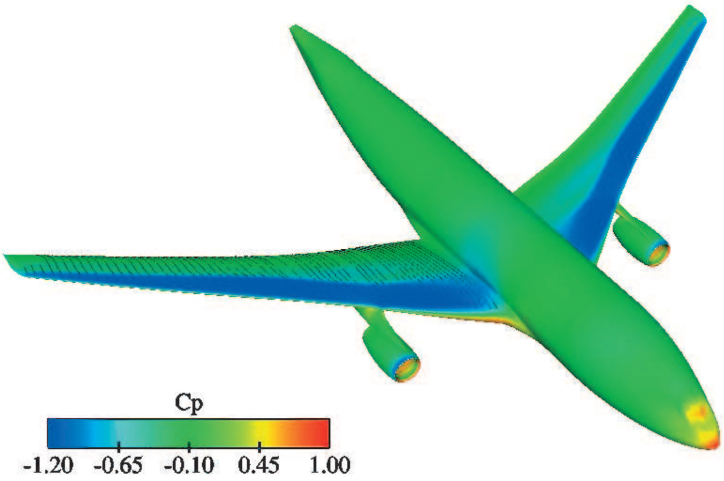

Figure 8. Surface pressure predicted by the EARSM at $M_\infty = 0.80$, $\alpha = 2.2°$ and $Re_{mac} = 9.97 \cdot 10^6$

trailing edge is also better predicted by the EARSM. Both models have minor problems predicting the right pressure level just down-stream of the shock. The flow over the wing is close to separation at the shock and at the trailing edge.

The typical run time for an application of this size is around 40 h on a Cray T3E using 30 processors.

The problem turn around time is of importance when using advanced flow modelling at an early stage in the design cycle. Here we present the parallel efficiency of the flow solver evaluated on different parallel platforms using computation of the turbulent flow field around the ONERA M6 wing at free-stream conditions $M_\infty = 0.84$, $\alpha = 3.06°$ and $Re_{mac} = 11.7 \cdot 10^6$. The surface pressure on the wing is presented in Figure 11. The computational mesh consists of 1.2 million cells divided in 32 blocks. The flow is modelled by the Wilcox k-ω model. Load balancing of the computation is performed with the algorithm described

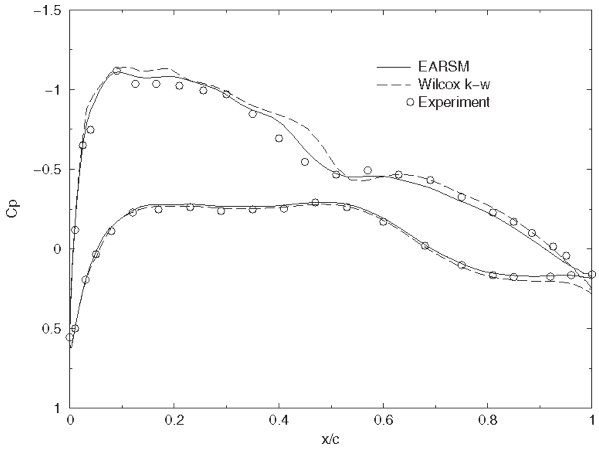

Figure 9. Pressure distribution at $y/b = 17\%$. Computations with EARSM and k-ω models compared with measurements

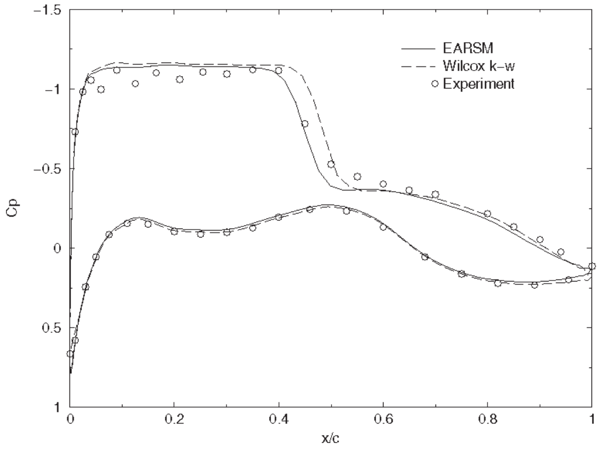

Figure 10. Pressure distribution at $y/b = 57\%$. Computations with EARSM and k-ω models compared with measurements

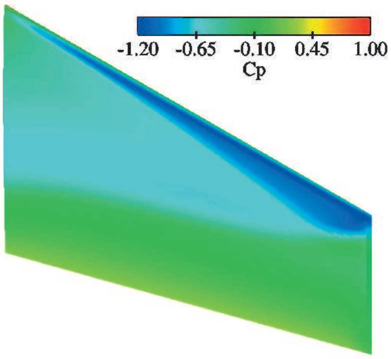

Figure 11. Surface pressure distribution on upper side of ONERA M6 at free-stream conditions $M_\infty = 0.84$, $\alpha = 3.06°$ and $Re_{mac} = 11.7 \cdot 10^6$

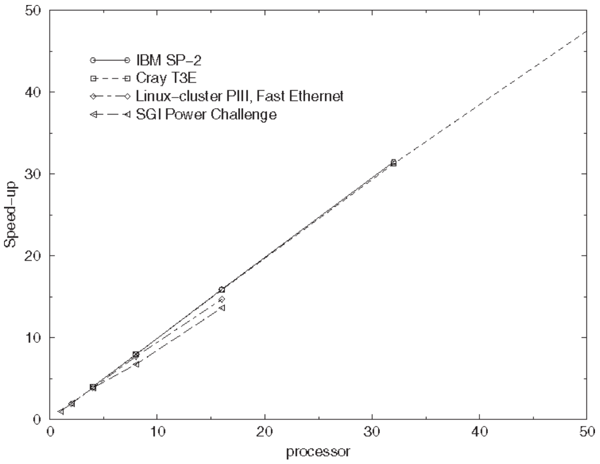

Figure 12. Speed-up numbers versus number of processors using different parallel computer platforms

in Section 4. The tested computers include IBM SP-2, SGI Power Challenge, Cray T3E and a Linux cluster. As can be seen in Figure 12 the speed-up is nearly linear for the evaluated computer configurations. The Cray T3E scales very well up to 128 processors (not fully included in the figure), where a speed-up of 116 is achieved. The latest tested computer, a Linux-cluster consisting of Pentium III processors connected via a Fast Ethernet network, performs well up to 16 processors. A slight degradation of the speed-up is seen above 8 processors, where the network is becoming saturated. The SGI Power Challenge shows a degradation of the speed-up at 8 processors due to contention problems on the shared memory bus, but recovers slightly at 16 processors. This is probably an effect of an improved usage of the cache memories.

This example shows that this type of explicit CFD code performs very well on parallel computers. To achieve a balanced load when using many processors the original 32 blocks are subdivided as described in the previous section. The results show that the load-balancing algorithm works well, introducing only a minor loss in efficiency compared to the theoretical speed-up.

6. Conclusions

The physically more advanced modelling of turbulent flow, as given by an EARSM, has been shown to give clearly improved results for a number of transonic aerospace applications compared to the traditional two-equation eddy-viscosity models represented by the Wilcox k-ω model.

For a flow simulation system to be applicable in the early stages of the design process, the problem turn-around time must be kept short. The use of parallel computers is a way to significantly reduce the CFD-program execution time. An algorithm has been described to show how to partition a structured multi-block grid for efficient use on parallel computer platforms. Good speed-up numbers have been reported for various parallel computers on a wing case.

References

[1] Gould A, Courty J-C, Sillén M, Elsholz E and Abbas A 2000 *The AVTAC Project – a Review of European Aerospace CFD* ECCOMAS Barcelona

[2] Wilcox D C 1993 *Turbulence modeling for CFD* DCW Industries Inc.

[3] Kok J 2000 *AIAA Journal* **38** (7)

[4] Wallin S and Johansson A V 2000 *J. Fluid Mech.* **403** 89

[5] Launder B E, Reece G J and Rodi W 1975 *J. Fluid Mech.* **68** 537

[6] Jameson A, Schmitt W and Turkel E 1981 *Numerical solution of the Euler equations by finite volume methods using Runge-Kutta time-stepping schems* AIAA paper 81-1259

[7] Jameson A 1988 *Pure and Applied Mathematics* **XLI** 507

[8] Wallin S 1999 *An efficient explicit algebraic Reynolds stress k-ω model (EARSM) for aeronautical applications* FFA TN 1999-71

[9] Ålund A, Lötstedt P and Sillén M 1997 *Computers & Fluids* **26** (7) 775

[10] Cock P H, MacDonald M A and Firmin M C P 1979 *Aerofoil 2822 – Pressure distribution, boundary layer and wake measurements* AGARD AR 138

[11] Horsten J J, Den Boer R G and Zwaan R J 1983 *Unsteady transonic pressure measurements on a semi-span wind tunnel model of a transport-type supercritical wing (LANN model)* AFWAL-TR-83-3039