Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-144

Efficient and accurate implementation of the k-ω turbulence model in the NLR multi-block Navier-Stokes system

J.C. Kok and S.P. Spekreijse

Nationaal Lucht- en Ruimtevaartlaboratorium
National Aerospace Laboratory NLR

NLR-TP-2000-144

Efficient and accurate implementation of the k-ω turbulence model in the NLR multi-block Navier-Stokes system

J.C. Kok and S.P. Spekreijse

This investigation has been carried out under a contract awarded by the Netherlands Agency for Aerospace Programmes (NIVR), contract number 01105N. NIVR has granted NLR permission to publish this report.

This report is based on a presentation held at the European Congress on Computational Methods in Applied Sciences and Engineering, ECCOMAS 2000, Barcelona, Spain, 11-14 September 2000.

Division:	Fluid Dynamics
Issued:	May 2000
Classification of title:	Unclassified

Summary

This paper deals with the implementation of the k–ω turbulence model in a multi-block Navier–Stokes flow solver, intended for the simulation of compressible flows around complex, aerodynamic configurations. The standard k–ω model has two main drawbacks: the results depend in an unphysical manner on the free-stream value of the turbulence variables and ω exhibits a singular behaviour at solid walls. A recently derived version of the k–ω model is employed that resolves the so-called free-stream dependency without using the wall distance. The numerical accuracy is strongly improved by using $\tau = 1/\omega$ as dependent variable, thus avoiding the singular behaviour of ω. Furthermore, the weak solution of the model equations at boundary-layer edges is properly accounted for in the numerical diffusion. The basic flow equations and the k–ω model equations are solved as one system of equations by a multi-grid scheme. Efficiency of this scheme is enhanced by using separate time steps (and residual-averaging coefficients) for the basic and the k–ω equations and by accounting for grid cells with high aspect ratios.

Contents

14 Figures

(30 pages in total)

List of symbols

$\vec{A}$	cell-face area vector
c	speed of sound
C_D	cross-diffusion term
C_p,C_f	pressure and skin-friction coefficients
C_L,C_D	lift and drag coefficients
E	total energy per unit volume
F^c,F^d,F^a	convective, diffusive, and artificial diffusive fluxes
g	$g = \sqrt{1/\omega}$
h	mesh size
k	turbulent kinetic energy
M	Mach number
P_k,P_ω,P_τ,P_g	production terms of k, ω, τ, and g
Re	Reynolds number
Re_t	turbulent Reynolds number: $Re_t = \mu_t/\mu$
S	'source' vector with non-conservative terms
t	time
U	flow-state vector: $U = (\rho, \rho\vec{u}, \rho E, \rho k, \rho\tau)^T$
V	cell volume
y	normal wall distance
$\vec{x}$	position vector with Cartesian components
$\vec{u}$,u	velocity vector and magnitude
α	angle of attack
α_ω	coefficient of production term of ω
β_k,β_ω	coefficients of dissipation terms of k and ω
δv,$\delta_3 v$	first- and third-order differences ($\delta v = v_{i,j,k} - v_{i-1,j,k}$, $\delta_3 v = v_{i+1,j,k} - 3v_{i,j,k} + 3v_{i-1,j,k} - v_{i-2,j,k}$)
Δt	time step
ϵ	turbulent dissipation rate
$\epsilon^{(2)}$,$\epsilon^{(4)}$	coefficients of artificial diffusion
λ	scaling factor of artificial diffusion
μ_k,μ_ω	viscosity coefficients of k and ω equations
μ,μ_t	dynamic molecular and eddy viscosity coefficients
ν	kinematic molecular viscosity coefficient

List of symbols (continued)

ρ	density
σ_k,σ_ω,σ_d	coefficients of diffusion terms of k and ω
τ	turbulent time scale: $\tau = 1/\omega$
$\boldsymbol{\tau}^R$	Reynolds-stress tensor
(ξ, η, ζ)	computational coordinates
ω	specific turbulent dissipation rate

Subscripts and superscripts:

f	face number
i, j, k	grid-cell indices
$+$	law–of–the–wall scaling
∞	free-stream value

List of figures

This page is intentionally left blank.

1 Introduction

The solution of the steady, compressible, Reynolds-averaged Navier–Stokes equations together with two-equation turbulence models is currently feasible for complex aerodynamic configurations, such as transport aircraft. Nevertheless, further improvements, in terms of accuracy, efficiency, and robustness, are still required if such computations are to become an engineering tool for aircraft design (Ref. 1). For such simulations, the k–ω turbulence model has become one of the more popular models for two main reasons: wall-damping functions are not strictly necessary, so that the computation of wall distances is not needed, and it is less stiff than k–ϵ models in the near-wall region. In particular the first property is desirable for complex configurations. However, the reliability of simulations based on the k–ω model may suffer from two problems: the results depend in an unphysical manner on the free-stream value of the turbulence variables and ω exhibits a singular behaviour at solid walls.

The unphysical free-stream dependence of the original Wilcox k–ω model (Ref. 2) was first recognized by Menter, who resolved this problem by using a blending between the k–ω and the k–ϵ model (Ref. 3), but at the cost of introducing the wall distance. The model switches from k–ω to k–ϵ (in a k–ω formulation) when approaching the boundary-layer edge, with as main effect the introduction of the so-called cross-diffusion term. Recently (Ref. 4), we have derived a new set of diffusion coefficients for the k–ω model, including the cross-diffusion term, that resolves the free-stream dependency, and that is valid throughout the complete boundary layer. Thus, a blending function and therefore also the wall distance is not required, making the model suitable for complex configurations. In this version of the model, the cross diffusion is only included when it is positive (as proposed by Wilcox (Ref. 5)), so that it is not effective close to the solid wall, as in the original k–ω model.

The singular behaviour of ω at solid walls (ω is inverse proportional to the square of the wall distance) may result for most numerical schemes in considerable numerical errors, implying a strong grid dependency, in particular for the skin-friction coefficient. This problem can be resolved by using as second dependent variable $\tau = 1/\omega$ instead of ω. Alternatively, one could also use $g = \sqrt{\tau}$ as in the k–g model of Kalitzin et al. (Ref. 6). However, it will be shown that the dominant terms in the equation for τ all remain bounded when approaching the solid wall, while the dominant terms in the equation for g (as in the equation for ω) are unbounded.

The k–ω model has been implemented in the Navier–Stokes flow solver ENSOLV (Refs. 7, 8), which previously employed algebraic turbulence models. The governing equations consist of the

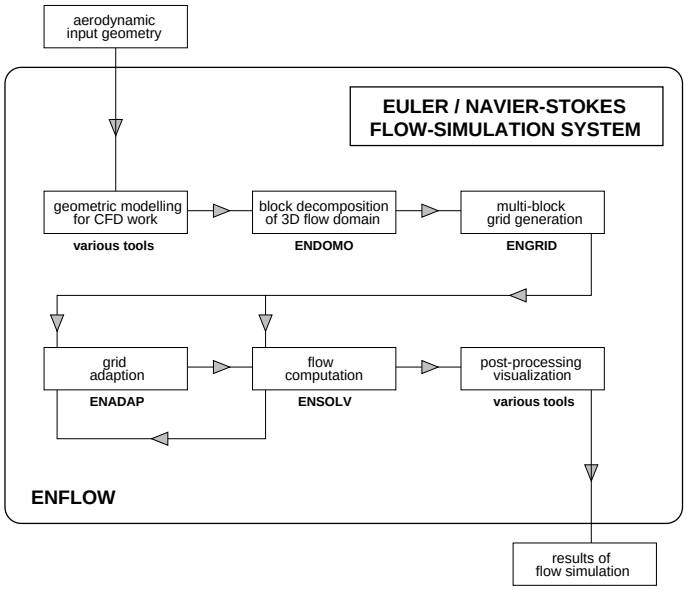


Fig. 1 The NLR multi-block flow-simulation system ENFLOW.

Reynolds-averaged Navier–Stokes equations, assuming a calorically perfect gas, using the Sutherland law to model the molecular-viscosity coefficient, and using the Boussinesq hypothesis to model the Reynolds-stress tensor. Here, we focus on the numerical aspects of including the k–ω model in this flow solver. In particular, we consider the numerical accuracy, for which the singular behaviour of ω is the main concern, and the efficiency of the method, aiming for a similar convergence rate as was previously obtained with the algebraic turbulence models.

The flow solver ENSOLV is part of the NLR flow-simulation system ENFLOW for multi-block structured grids (Ref. 9). This system further includes tools for geometry modelling, grid generation, and flow visualization (figure 1). The system has been employed to simulate the flows (both on the basis of the Euler equations and of the Reynolds-averaged Navier–Stokes equations) around a large variety of complex aerodynamic configurations, extending from civil and military aircraft to spacecraft. Main bottleneck to allow efficient employment of the system for engineering purposes is the man-hour consuming grid-generation task. In the ENFLOW system, this task is done in two steps (Ref. 10): first, the flow domain is divided into subdomains called blocks using the domain modeller ENDOMO; second, structured grids are generated subsequently in the edges, the faces, and the blocks using the grid generator ENGRID. Currently, work is ongoing to reduce the turn-around times by automatic generation of blocks through solving the potential equation (Ref. 11).

2 Turbulence-Model Equations

2.1 The k–ω model with TNT coefficients

The k–ω model equations, including the so-called cross-diffusion term, are given by

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho k \vec{u}) = P_k - \beta_k \rho \omega k + \nabla \cdot (\mu_k \nabla k), \tag{1}$$

$$\frac{\partial \rho \omega}{\partial t} + \nabla \cdot (\rho \omega \vec{u}) = P_\omega - \beta_\omega \rho \omega^2 + \nabla \cdot (\mu_\omega \nabla \omega) + C_D, \tag{2}$$

with ρ the density, $\vec{u}$ the velocity vector, k the turbulent kinetic energy, and ω the specific turbulent dissipation. The respective viscosity coefficients are given by

$$\mu_k = \mu + \sigma_k \mu_t, \tag{3}$$

$$\mu_\omega = \mu + \sigma_\omega \mu_t, \tag{4}$$

with μ the molecular-viscosity coefficient and $\mu_t = \rho k/\omega$ the eddy-viscosity coefficient. The production terms and the cross-diffusion term are given by

$$P_k = \boldsymbol{\tau}^R \cdot \nabla \vec{u}, \tag{5}$$

$$P_\omega = \frac{\alpha_\omega \omega}{k} P_k, \tag{6}$$

$$C_D = \sigma_d \frac{\rho}{\omega} \max \{\nabla k \cdot \nabla \omega, 0\}. \tag{7}$$

with $\boldsymbol{\tau}^R$ the Reynolds-stress tensor. Note that the cross-diffusion term is only included when it is positive. This means that it is not effective close to the solid wall (where k and ω have opposite gradients), which is crucial for a correct behaviour of the k–ω model. The k–ω model has six closure coefficients: α_ω, β_k, β_ω, σ_k, σ_ω, and σ_d. In the original k–ω model, these coefficients are given by

$$\alpha_\omega = 0.5, \quad \beta_k = 0.09, \quad \beta_\omega = 0.075, \quad \sigma_k = 0.5, \quad \sigma_\omega = 0.5, \quad \sigma_d = 0.0. \tag{8}$$

However, here we employ the TNT set of values for the diffusion coefficients, given by

$$\sigma_k = 2/3, \quad \sigma_\omega = 0.5, \quad \sigma_d = 0.5. \tag{9}$$

These coefficients were recently derived by Kok (Ref. 4) using a theoretical analysis for turbulent / non-turbulent (TNT) interfaces that was based on an analysis by Cazalbou et al. (Ref. 12) for k–ϵ models. It was demonstrated that this set of coefficients effectively resolves the free-stream dependency.

2.2 The k–τ formulation

Solution of the k–ω model results in a singular behaviour for ω at solid walls with perfectly smooth surfaces. This solution, which is valid in the viscous sublayer, is given by (Ref. 13)

$$\omega \rightarrow \frac{6\nu}{\beta_\omega y^2}, \qquad y \rightarrow 0, \tag{10}$$

with ν the kinematic viscosity and y the normal distance to the solid wall. Near the wall, the dominant terms in the ω equation are the dissipation and diffusion terms, which behave as

$$\beta_\omega \rho \omega^2 \rightarrow \frac{36\rho\nu^2}{\beta_\omega y^4}, \qquad \nabla \cdot (\mu_\omega \nabla \omega) \rightarrow \frac{36\rho\nu^2}{\beta_\omega y^4}. \tag{11}$$

This singularity must be treated carefully in the numerical discretization of the equations in order to avoid large numerical errors. In practice, often a finite value for ω is used at solid walls, e.g., using a rough wall boundary condition with the surface roughness small enough to simulate a hydraulically smooth wall (Ref. 13). However, the distribution of ω still exhibits a 'near' singular behaviour, with large first and second-order derivatives, which may easily lead to large discretization errors.

The singular behaviour can be avoided by switching to a different dependent variable. In particular, $\tau = 1/\omega$ ($\sim y^2$ close to the wall) and $g = \sqrt{\tau}$ ($\sim y$) may be considered, as was also done by Kalitzin et al. (Ref. 6). The equation for τ is given by

$$\frac{\partial \rho\tau}{\partial t} + \nabla \cdot (\rho\tau\vec{u}) = -P_\tau + \beta_\omega \rho + \nabla \cdot (\mu_\omega \nabla \tau) - 8\mu_\omega \left\| \nabla\sqrt{\tau} \right\|^2 + C_D, \tag{12}$$

$$P_\tau = \frac{\alpha_\omega \tau}{k} P_k, \qquad C_D = \sigma_d \rho\tau \min\left\{ \nabla k \cdot \nabla \tau, 0 \right\}, \tag{13}$$

while the equation for g is given by

$$\frac{\partial \rho g}{\partial t} + \nabla \cdot (\rho g\vec{u}) = -P_g + \frac{\beta_\omega \rho}{2g} + \nabla \cdot (\mu_\omega \nabla g) - 3\frac{\mu_\omega}{g} \left\| \nabla g \right\|^2 + C_D, \tag{14}$$

$$P_g = \frac{\alpha_\omega g}{2k} P_k, \qquad C_D = \sigma_d \rho g^2 \min\left\{ \nabla k \cdot \nabla g, 0 \right\}, \tag{15}$$

Near the wall, the dissipation and diffusion terms in the τ equation behave as

$$\beta_\omega \rho \rightarrow \beta_\omega \rho, \qquad \nabla \cdot (\mu \nabla \tau) \rightarrow \tfrac{1}{3}\beta_\omega \rho, \qquad 8\mu_\omega \left\| \nabla\sqrt{\tau} \right\|^2 \rightarrow \tfrac{4}{3}\beta_\omega \rho, \tag{16}$$

while in the g equation these terms behave as

$$\frac{\beta_\omega \rho}{2g} \rightarrow \sqrt{\tfrac{3}{2}\nu\beta_\omega}\frac{\rho}{y}, \qquad \nabla \cdot (\mu_\omega \nabla g) \rightarrow 0, \qquad 3\frac{\mu_\omega}{g} \left\| \nabla g \right\|^2 \rightarrow \sqrt{\tfrac{3}{2}\nu\beta_\omega}\frac{\rho}{y}, \tag{17}$$

Thus, we see that in the τ equation all terms remain bounded, but that in the g equation, just like in the ω equation, the dominant terms are singular (although their behaviour is more moderate than in the ω equation). Therefore, we prefer to use the τ formulation. In order to allow that ω approaches zero in the free stream, τ is redefined as $\tau = 1/(\omega + \omega_0)$ with $\omega_0 = 20u_\infty/L$ by default (which is generally the order of ω near the boundary-layer edge).

In the k–g model of Kalitzin et al., the g formulation was chosen because of the linear behaviour of g near the solid wall. They found that the quadratic behaviour of τ close to the wall caused large numerical errors when computing its gradient using central differences. This was particularly the case for the additional diffusion term in the τ equation containing a product of gradients of τ as given by

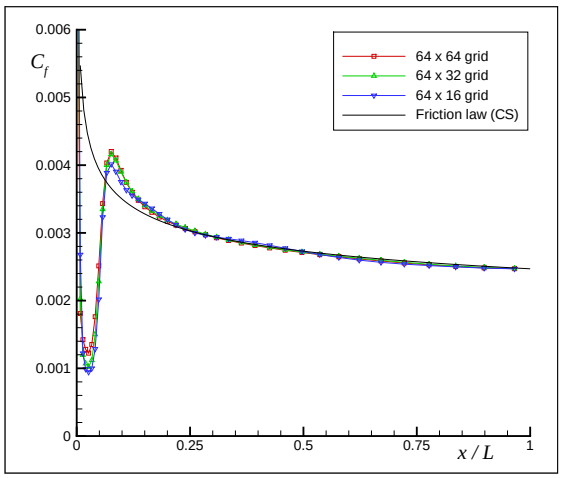
$$2\frac{\mu_\omega}{\tau}\left\|\nabla\tau\right\|^2. \tag{18}$$

However, to circumvent this problem it is not necessary to completely switch to the g equation, but it is sufficient to rewrite this additional diffusion term using gradients of $\sqrt{\tau}$ (i.e. g), as is done in equation (12).

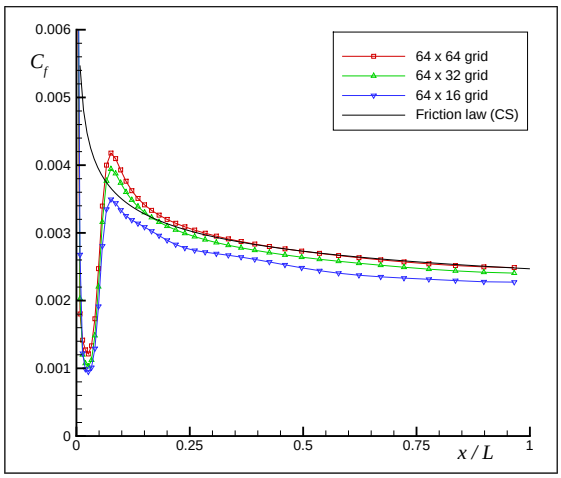
The improved numerical accuracy when using the k–τ formulation is illustrated by computations for the constant-pressure flat-plate boundary layer at $Re_\infty = 10^7$ and $M_\infty = 0.5$. Transition is prescribed at 5% from the leading edge. The free-stream turbulent Reynolds number is taken as $Re_{t,\infty} = (\mu_t/\mu)_\infty = 10^{-2}$, while the free-stream value of k is taken as $k_\infty/u_\infty^2 = 10^{-6}$. A grid is used with 64×64 grid cells of which 40 in stream direction on the flat plate and approximately 30 to 40 in normal direction inside the turbulent boundary layer. For the first grid point above the flat plate $y^+ \approx 1$.

The grid dependency for the k–ω and the k–τ formulations are illustrated by figure 2, showing the skin-friction coefficient as well as the velocity and ω distributions at the location $Re_x = 5 \cdot 10^6$. To focus on the numerical accuracy related to the normal direction, the grid is coarsened in the normal direction only. The strong difference in grid dependency is apparent. With the k–τ formulation, the correct level of the skin-friction coefficient is already obtained on the coarsest grid, which has only about 10 grid points inside the boundary layer. For the velocity and ω distributions, only minor differences are observed on the coarser grids. The distribution of ω is consistent with the theoretical law–of–the–wall solution, particular in the viscous sublayer (equation (10)). For the k–ω formulation, the 'slightly' rough-wall condition for ω results in a different solution in the viscous sublayer. The roughness is small enough to obtain the correct skin friction on the finest

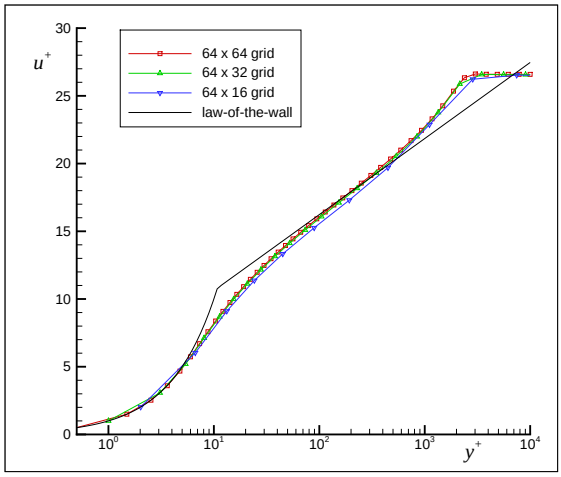
grid level. However, on the coarser grids, numerical errors cause too high values of ω, extending into the log layer, which reduce the eddy-viscosity level and subsequently also the skin-friction coefficient.



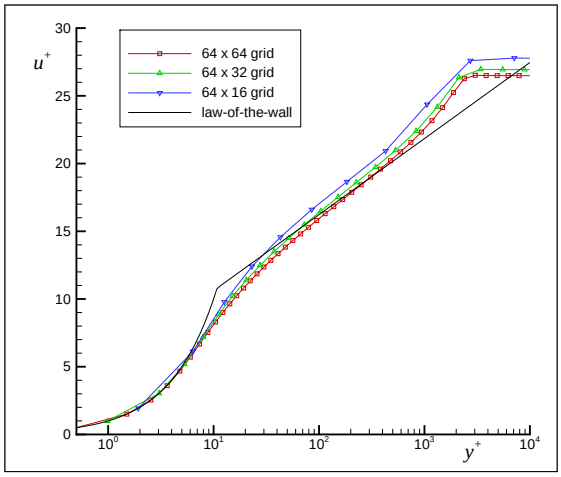
a) Skin-friction coefficient; k–τ formulation



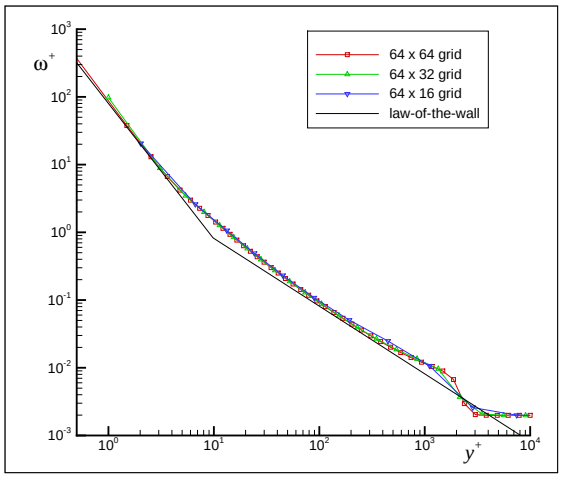
b) Skin-friction coefficient; k–ω formulation



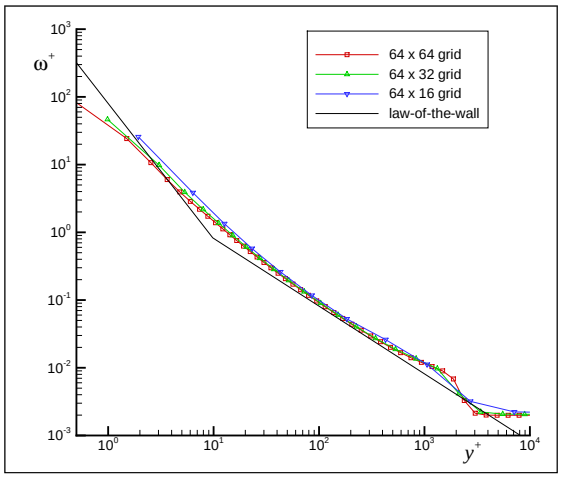
c) Velocity distribution; k–τ formulation



d) Velocity distribution; k–ω formulation



e) Distribution of ω; k–τ formulation



f) Distribution of ω; k–ω formulation

Fig. 2 Flow solution of k–ω model both in k–τ and in k–ω formulation for flat plate at $Re_\infty = 10^7$ and $M_\infty = 0.5$.

3 Numerical Method

3.1 Space discretization

The k–ω model is implemented in the multi-block Reynolds-averaged Navier–Stokes flow solver ENSOLV by considering the basic flow equations and the k–τ equations as one system of equations. The equations are discretized in space using a cell-centred finite-volume scheme with scalar and matrix Jameson-type artificial diffusion. For each block (containing a structured grid), let the grid cells be indicated with the indices (i, j, k). For each cell, let the cell faces be numbered from 1 to 6 (with positive i direction from face 1 to 2, positive j from 3 to 4, and positive k from 5 to 6). Let the flow state vector at each cell centre be given by $U_{i,j,k} = (\rho, \rho\vec{u}, \rho E, \rho k, \rho\tau)^T_{i,j,k}$, with E the total energy per unit mass. Integrating the flow equations (in conservative form) over the cell volume, and applying Gauss' divergence theorem, the set of semi-discrete equations is obtained:

$$\frac{dU_{i,j,k}}{dt} = \frac{1}{V_{i,j,k}} \sum_{f=1}^{6} \left(-F_f^c + F_f^d + F_f^a \right)_{i,j,k} + S_{i,j,k}, \tag{19}$$

with $V_{i,j,k}$ the cell volume, F_f^c, F_f^d, and F_f^a the convective, diffusive, and artificial diffusive fluxes across the six cell faces, and $S_{i,j,k}$ the 'source' term containing the non-conservative terms of the k–τ equations. For the convective and diffusive fluxes, values of the flow variables and their gradients are needed at the cell faces. The value of the flow variables are obtained by simple averaging. For an arbitrary flow variable v, the gradient is computed by

$$\nabla v = \frac{\partial v}{\partial \xi}\nabla\xi + \frac{\partial v}{\partial \eta}\nabla\eta + \frac{\partial v}{\partial \zeta}\nabla\zeta, \tag{20}$$

with (ξ, η, ζ) the computational coordinates corresponding to the (i, j, k) indices. To compute the gradients of the computational coordinates, the Jacobian $\mathcal{J} = (\partial\vec{x}/\partial\xi, \partial\vec{x}/\partial\eta, \partial\vec{x}/\partial\zeta)$ is inverted. For the differences of v and $\vec{x}$ with respect to the computational coordinates, the same stencil is employed, so that the computation of the gradient of v is exact in case of a linear distribution of v.

For the basic flow equations, the artificial-diffusive flux is defined by the method of Jameson (Ref. 14): a blending of first and third-order differences with a pressure-based shock sensor. In order to improve the numerical accuracy in the boundary layers, without additional costs, the scheme is extended with a reduced form of the matrix scaling of Swanson and Turkel (Ref. 15), as described by Kok (Refs. 7, 8).

For the turbulence-model equations, the artificial-diffusive flux is also defined by a blending of first and third-order differences,

$$F_f^a = \lambda_f(\tfrac{1}{2}\epsilon^{(2)}\delta v - \epsilon^{(4)}\delta_3 v), \tag{21}$$

with v either ρk or $\rho\tau$, and with the scaling factor λ_f given by the eigenvalue of the convective flux of the k–τ equations, which is the velocity component normal to the cell face. In order to avoid that the artificial-diffusive flux becomes equal to zero when the flow is aligned with the grid, the spectral radius of the convective flux Jacobian of the basic flow equations is used to define a lower bound:

$$\lambda_f = \max\left\{|\vec{u}\cdot\vec{A}_f|\,,\, \varepsilon(|\vec{u}\cdot\vec{A}_f| + c\|\vec{A}_f\|)\right\} \tag{22}$$

with c the speed of sound, $\vec{A}_f$ the cell-face area vector, and $\varepsilon = 0.01$.

The turbulence equations generally allow a weak solution at the edge of the boundary layer. Also, very high gradients that are represented on the grid practically as discontinuities may appear near transition and near trailing edges. Improper treatment of these situations may result in oscillations in the solution and as a consequence possibly negative values of the turbulence variables. Apart from being unphysical, these negative values may also induce a breakdown of the computations. To properly capture the weak solution, a separate TVD sensor (distinct form the pressure-based sensor of the basic flow equations) is used to switch on the second-order artificial diffusion in the turbulence equations. This sensor was derived by Swanson and Turkel (Ref. 15) for central schemes from the 'upwind' Van Leer sensor, and is given for the i direction by

$$\nu_{i,j,k} = \frac{|v_{i+1,j,k} - 2v_{i,j,k} + v_{i-1,j,k}|}{|v_{i+1,j,k} - v_{i,j,k}| + |v_{i,j,k} - v_{i-1,j,k}| + 10^{-20}}. \tag{23}$$

The coefficients of the artificial-diffusive flux at the cell face $f = 1$ are now defined by

$$\epsilon^{(2)} = \max\{\nu_{i-1,j,k}, \nu_{i,j,k}\}, \tag{24}$$

$$\epsilon^{(4)} = \tfrac{1}{64}k^{(4)}(1 - \epsilon^{(2)}), \tag{25}$$

with $k^{(4)}$ a numerical parameter that is also used for the fourth-order artificial diffusion of the basic flow equations (and taken as $k^{(4)} = 2$). Near discontinuities (and extrema) in the solution, the TVD sensor goes to one, so that the second-order diffusion is fully switched on (and the fourth-order diffusion is fully switched off) and thus locally the scheme reduces to essentially a first-order upwind scheme.

Remains the discretization of the non-conservative terms in the k–τ equations, needed at cell centres. The velocity gradient, required in the production terms, as well as the gradients of k and τ, required in the cross-diffusion term, are computed by integrating over the cell volume and applying Gauss' divergence theorem

$$(\nabla v)_{i,j,k} = \frac{1}{V_{i,j,k}} \sum_{f=1}^{6} v_f \vec{A}_f, \tag{26}$$

with the values at the cell faces computed by simple averaging. The cross-diffusion term is a convection-like term that is important at turbulent/non-turbulent interfaces, such as the boundary-layer edge (where the k–τ equations have a weak solution) as was shown in (Ref. 4). Central discretization may then introduce oscillations. In practice, however, the artificial and physical diffusion already included in the τ equation appears to be sufficient to avoid this. Finally, the additional diffusive term in the τ equation (the fourth term on the right-hand side of equation (12)), requires some extra attention. This term is always negative, and, if not treated carefully, may generate negative values of τ. Therefore, the gradient of $\sqrt{\tau}$ is discretized such that it is exactly equal to zero in case of a local minimum of $\sqrt{\tau}$. Using an analogy with an upwind discretization, the gradient is obtained by using equation (26) but with the value of $v = \sqrt{\tau}$ at the cell face $f = 1$ computed by

$$v_f = \tfrac{1}{2}(v_{i-1,j,k} + v_{i,j,k}) - \tfrac{1}{2}\nu_{i,j,k}|v_{i,j,k} - v_{i-1,j,k}|, \tag{27}$$

where ν is the same sensor as equation (23). Thus, if the cell-centre value is a local minimum, then the sensors for the three computational directions all equal one, implying $v_f = v_{i,j,k}$ for all the six cell faces, and thus $\nabla v = 0$.

3.2 Solution procedure

The solution procedure for the basic flow equations (excluding the turbulence model equations), consists of a multigrid scheme with Runge-Kutta time integration as relaxation procedure, accelerated by local time stepping and implicit residual averaging with varying coefficients (Ref. 8). A W-cycle is used, with 5 pre and 5 post relaxations (which was found to be more robust than using just one pre/post relaxation). On each grid level of a multi-grid cycle, a loop over all blocks is performed, while a complete Runge–Kutta time step is performed for each block successively. Restriction is done by volume-weighted averaging, while prolongation is done by trilinear interpolation.

Including the k–ω model, the same solution method is applied considering the basic flow equations and the turbulence-model equations as one system of equations. The only essentially new

terms to take into account are the source terms of the k–τ equations. These are treated explicitly, including an appropriate restriction of the time step based on the eigenvalues of the source terms. Furthermore, the source terms are also computed on the coarse levels of the multi-grid scheme; only the eddy-viscosity coefficient is restricted form the finest level.

The convergence rate of this solution procedure is dominated by the stiffness of the equations close to the solid wall. There, the three time-step restrictions that are required, related to convection, to diffusion, and to the source terms of the k–τ equations, are given by, respectively,

$$\Delta t_c = \frac{h}{|u_n| + c}, \qquad \Delta t_d = \frac{h^2}{4\nu}, \qquad \Delta t_s = \frac{1}{\beta_k \omega}, \tag{28}$$

with h the mesh size in normal direction and u_n the normal velocity component. Applying law–of–the–wall scaling, substituting the viscous-sublayer solutions of velocity ($u^+ = y^+$) and of ω (equation (10)), and neglecting the normal velocity in the convective time step, we find

$$\Delta t_c^+ = M \frac{h^+}{y^+}, \qquad \Delta t_d^+ = \tfrac{1}{4}(h^+)^2, \qquad \Delta t_s^+ = \frac{\beta_\omega}{6\beta_k}(y^+)^2, \tag{29}$$

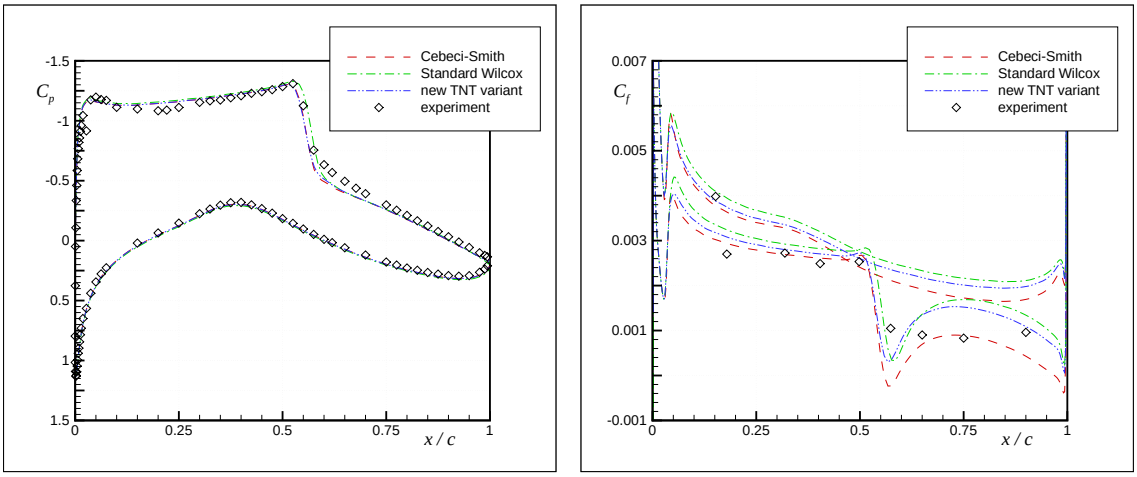
with M the local Mach number. In aerodynamic computations, it is common practice to use for the first layer of grid cells along the solid wall a mesh size $h^+ \approx 1$ (which was also shown to result in sufficient accuracy for the flat plate, at least for the k–τ formulation). It is then clear that for these first grid cells (where $y^+ \approx 1$ and $M \ll 1$), the convective time step dominates, while the source-term time step is of the same order as the diffusive time step. Thus, there is no need to treat the source terms implicitly. The convective time step, however, is related to the maximum convective eigenvalue of the basic flow equations and can significantly slow down the convergence of the k–τ equations, for which the convective eigenvalue is proportional to the normal velocity component. Therefore, to enhance the efficiency of the scheme, a separate time step is used for the k–τ equations based on the convective eigenvalue given by equation (22) with $\varepsilon = 0.1$. Finally, the source-term time step is only included in the time step for the turbulence equations.

Grid cells with high aspect ratios, which are usually present close to the wall, can seriously reduce the convergence rate. For the basic flow equations, this is resolved by an appropriate scaling of the artificial diffusion and by using varying coefficients for the residual averaging following Martinelli (Ref. 16). For the turbulence-model equations, the high-aspect-ratio scaling is applied to the fourth-order diffusion only.

4 Results

As a first representative 2D test case, we consider the RAE2822 airfoil at $M_\infty = 0.73$, $Re_\infty = 6.5 \cdot 10^6$, and $\alpha = 2.8^\circ$ (case 9 of reference 17). Transition is fixed at 3% from the leading edge. The free-stream turbulent variables are taken as $Re_{t,\infty} = (\mu_t/\mu)_\infty = 10^{-2}$ and $k_\infty/u_\infty^2 = 10^{-6}$ (which values are also used for the other testcases). An 8-block C-type grid is used with 528×96 grid cells, of which 384 around the airfoil and approximately 30 to 40 in normal direction inside the bounary layer. For the first grid point above the arfoil surface $y^+ < 1$. The far field is located at 50 chords from the airfoil.

The solution of the k–ω model with the new TNT coefficients is compared to the original Wilcox model and the Cebeci–Smith algebraic model in figure 3. The Wilcox model gives a shock position

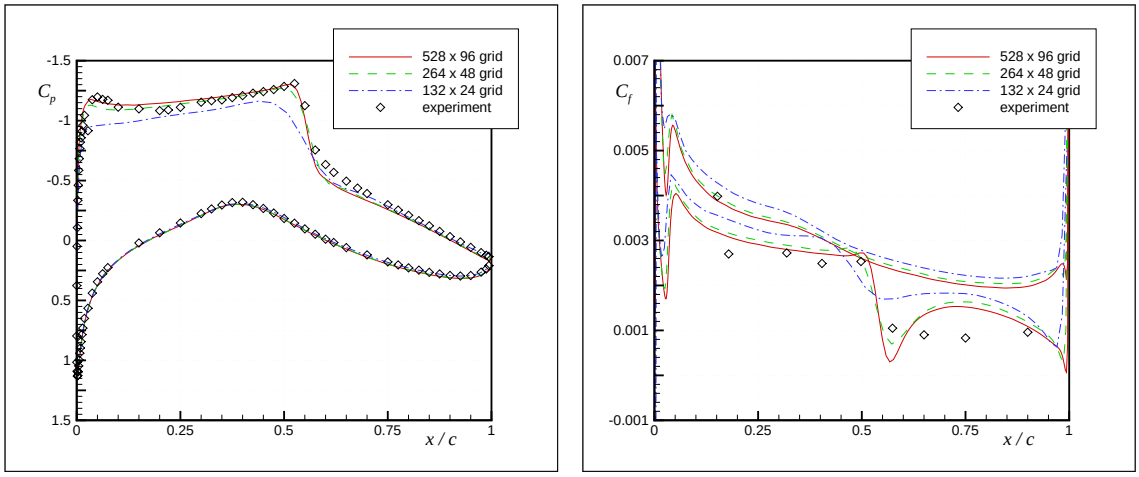


a) Pressure coefficient *b) Skin-friction coefficient*

Fig. 3 Flow solution with TNT and Wilcox k–ω models, and with Cebeci–Smith model for RAE2822 airfoil, case 9.

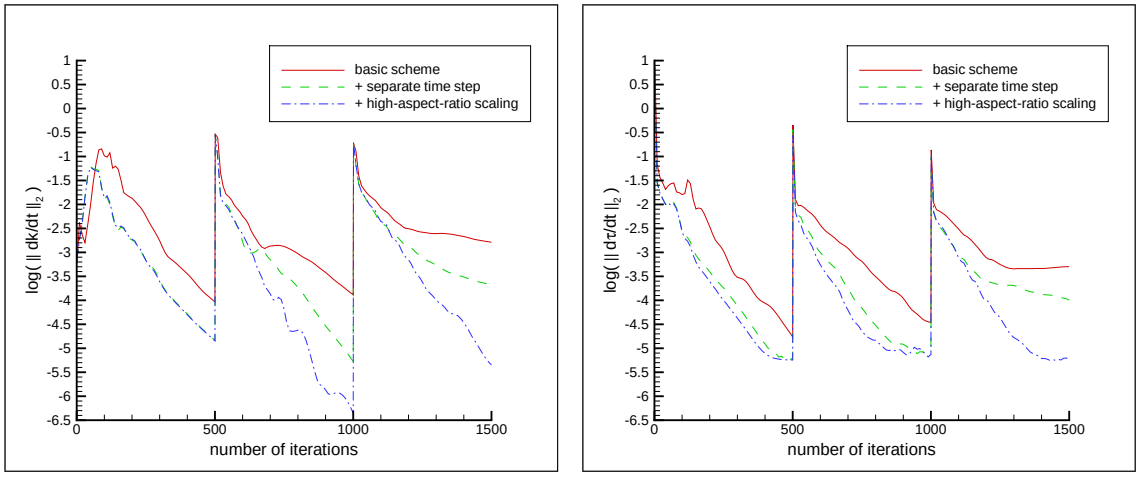
slightly aft of the other two models as well as a somewhat higher skin-friction level. As pointed out in reference 4, this may very well be a consequence of the free-stream dependency. The grid dependency with the TNT k–ω variant (in k–τ formulation), shown in figure 4, is found to be satisfactory, although for the skin-friction coefficient it appears to be somewhat stronger compared to the flat plate. However, in this case the grid was coarsened in both computational directions, so that the dependency on the grid resolution in circumferential direction dominates (in particular near shock and transition).

The convergence histories of the k and τ equations for this case are shown in figure 5. The basic scheme, with the same time step for the basic and the turbulence equations and without high-



a) Pressure coefficient *b) Skin-friction coefficient*

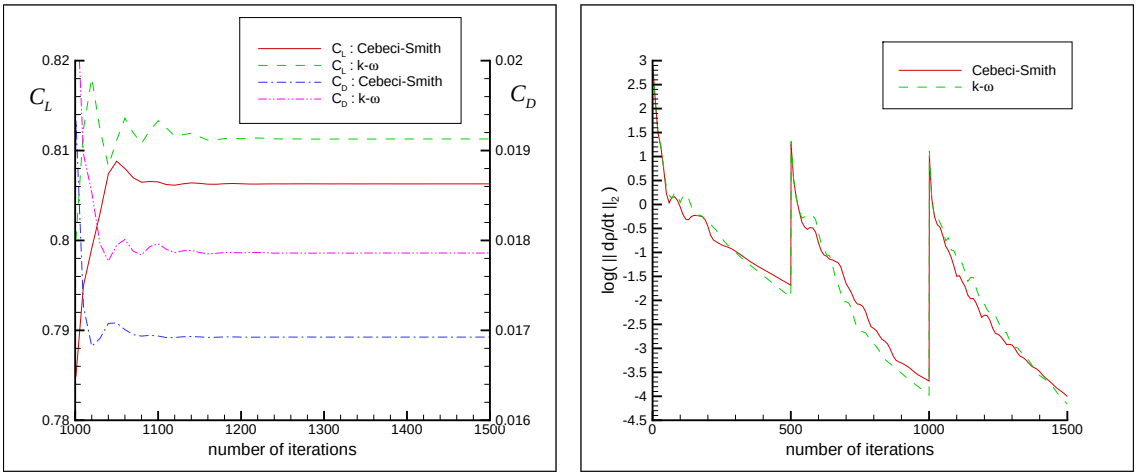
Fig. 4 Grid dependency of flow solution with TNT k–ω model in k–τ formulation for RAE2822 airfoil, case 9.



a) k equation *b) τ equation*

Fig. 5 Convergence history of k and τ equations for TNT k–ω model, RAE2822 airfoil, case 9.

aspect-ratio scaling of the artificial diffusion in the turbulence equations, gives a slow convergence rate, which even seems to stall for the τ equation on the finest grid. Using a separate time step for the turbulence equation clearly improves the convergence rate, in particular on the coarser grid levels and during the initial iterations on the finest level. A satisfactory convergence rate is only obtained if also the high-aspect-ratio scaling is included for the k–τ equations. With this scheme, a convergence rate is obtained that is comparable to the computations with the Cebeci–Smith model, as shown in figure 6 for the aerodynamic forces and for the mass equation. The force coefficients



a) Aerodynamic forces *b) Mass equation*

Fig. 6 Convergence history of aerodynamic forces and mass equation with Cebeci–Smith and with TNT k–ω model for RAE2822 airfoil, case 9.

are already sufficiently converged (C_D within 1 count) within 200 iterations. Note that with one iteration is meant one fine-grid relaxation (or Runge–Kutta time step), so that those 200 iterations are in fact equivalent to 20 multi-grid cycles (with 5 pre and 5 post relaxations per cycle).

The efficiency of the method is further evaluated for a 3D test case: the ONERA M6 wing at $M_\infty = 0.84$, $Re_\infty = 11.8 \cdot 10^6$, and $\alpha = 3.06°$. A 12-block, CO-type grid is employed with $256 \times 64 \times 48$ grid cells. An impression of the flow solution is given in figure 7. The convergence is slower than in 2D, due to higher cell aspect ratios, but with the k–ω model still a convergence rate is obtained comparable to computations with algebraic models (in this case Baldwin–Lomax), as shown in figure 8. As before, the force coefficients are sufficiently converged within 200 iterations. Only on the finest grid level does the convergence with the k–ω model tend to slow down, as shown in figure 9 for the k and τ equations. The convergence was found to be the slowest in the wake (not near the airfoil surface), as a consequence of the fact that the boundary-layer grid resolution was largely maintained in the wake, while the damping due to the source terms and physical diffusion

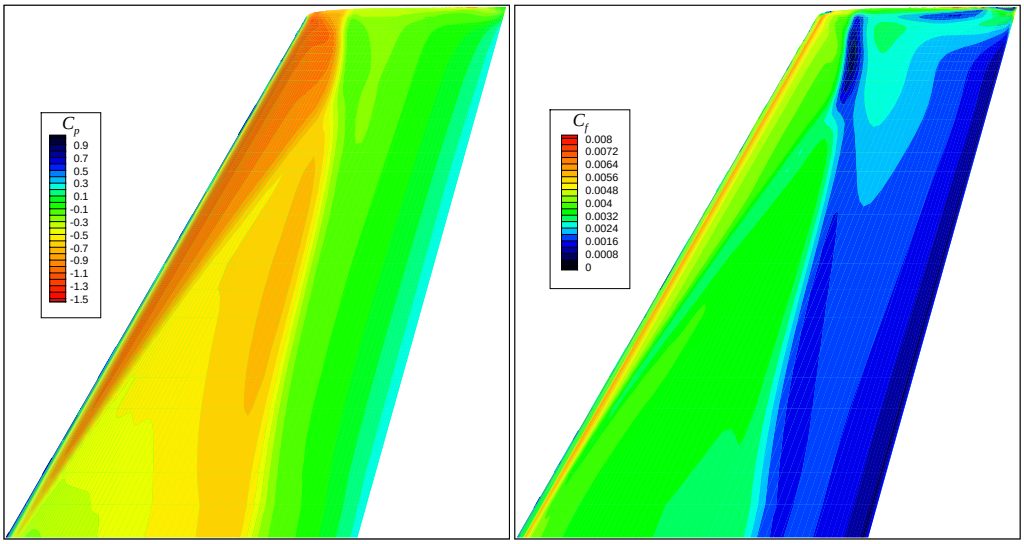
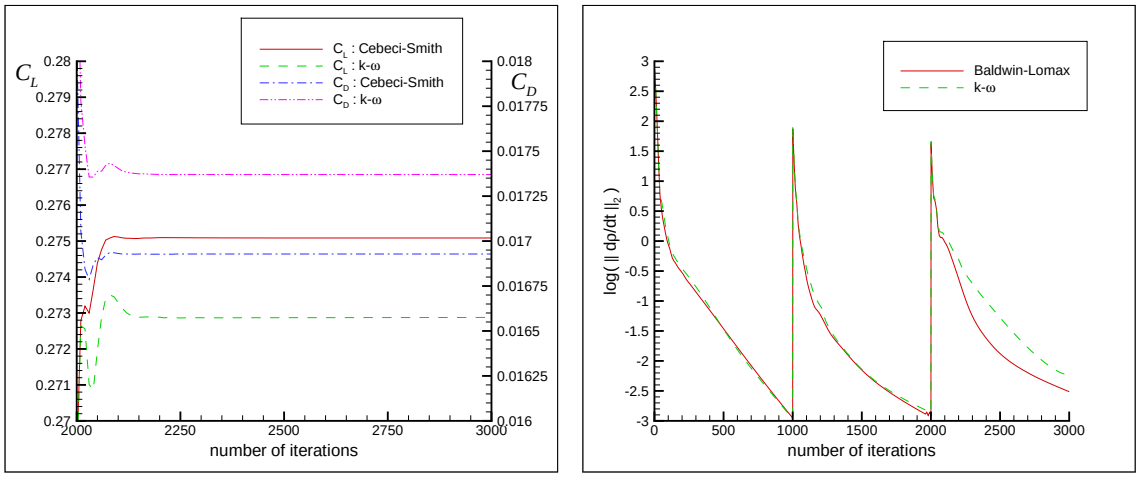
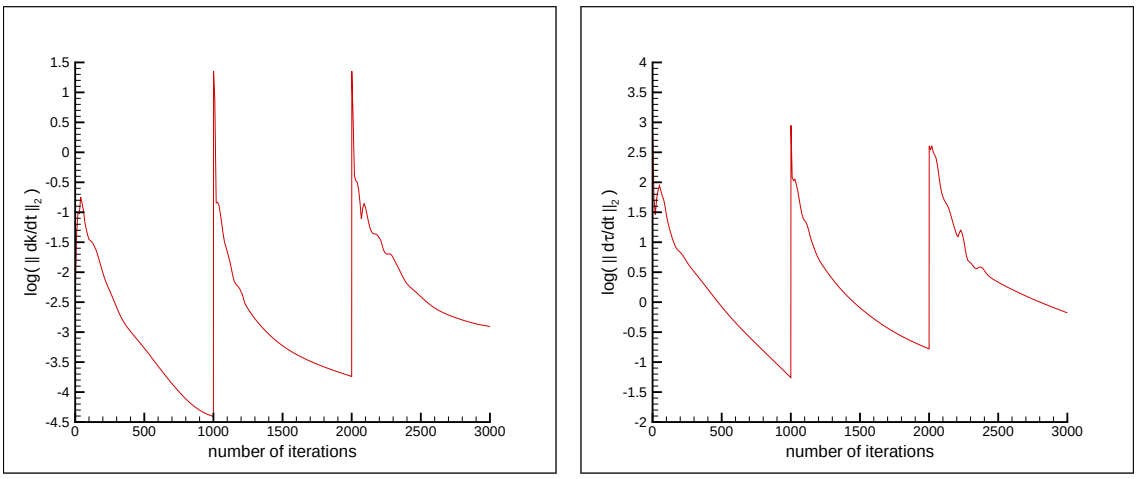


Fig. 7 Pressure and skin-friction distribution on upper side of ONERA M6 wing for TNT k–ω model ($M_\infty = 0.84$, $Re_\infty = 11.8 \cdot 10^6$, and $\alpha = 3.06°$).



a) Aerodynamic forces *b) Mass equation*

Fig. 8 Convergence history of aerodynamic forces and mass equation with Baldwin–Lomax and with TNT k–ω model for ONERA M6 wing.



a) k equation *b) τ equation*

Fig. 9 Convergence history of k and τ equations for TNT k–ω model, ONERA M6 wing.

are much weaker there. This can be alleviated easily by spreading out the grid lines in the wake as was done in fact for the RAE2822 airfoil.

Finally, the applicability of the ENFLOW system to complex configurations is illustrated by two typical transport-type aircraft configurations: a wing/body with pylon and flow-through nacelle, and a wing/body with pylon and turbofan engine. For both cases, grids have been generated with 240 and 120 blocks, respectively, with 4 to 5 million grid cells, and with for the first grid point above the wing surface $y^+ \approx 1$ to 3. In order to obtain a well-converged solution, 500 to 1000 iterations (i.e. 50 to 100 multi-grid cycles) were needed.

For the first configuration, computations have been performed at $M_\infty = 0.80$, $Re_\infty = 10.16 \cdot 10^6$, and $\alpha = 2.2^\circ$. An impression of the grid is given in figure 10. The comparison of the pressure coefficient with experimental results at several span-wise stations (figure 11) is found to be in line with what is commonly reported (e.g. reference 1). Comparison with results on a coarse grid (half the number of grid cells in each direction) for the pressure and the skin-friction (figure 12) shows that the grid dependence is fairly weak, in particular for the pressure coefficient.

For the second wing/body/pylon/engine configuration, computations have been performed at a lift coefficient of $C_L = 0.50$, and further $M_\infty = 0.75$ and $Re_\infty = 4.3 \cdot 10^6$. An impression of the surface grid is given in figure 13. At the fan exit, boundary conditions have been applied to simulate both fan 'through-flow' and fan 'power-on' conditions. An impression of the fan-power effect on the pressure distribution at the aircraft surface near the engine afterbody is given in figure 14.

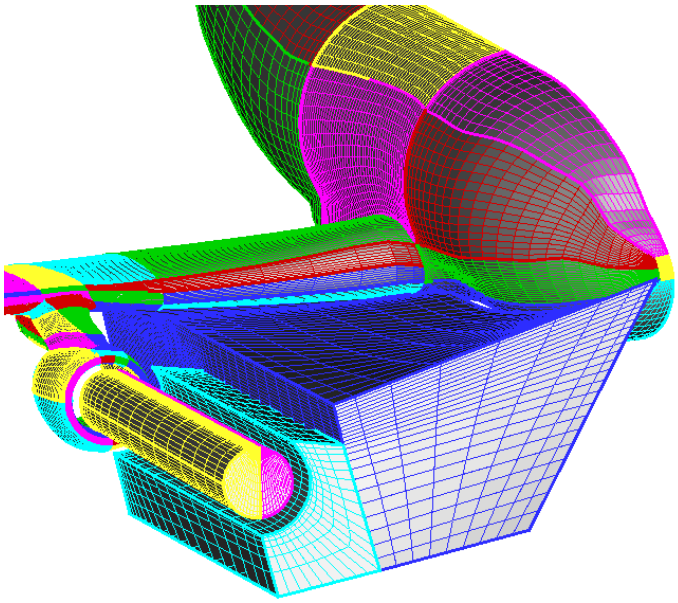


Fig. 10 Impression of grid around typical wing/body/pylon/nacelle.

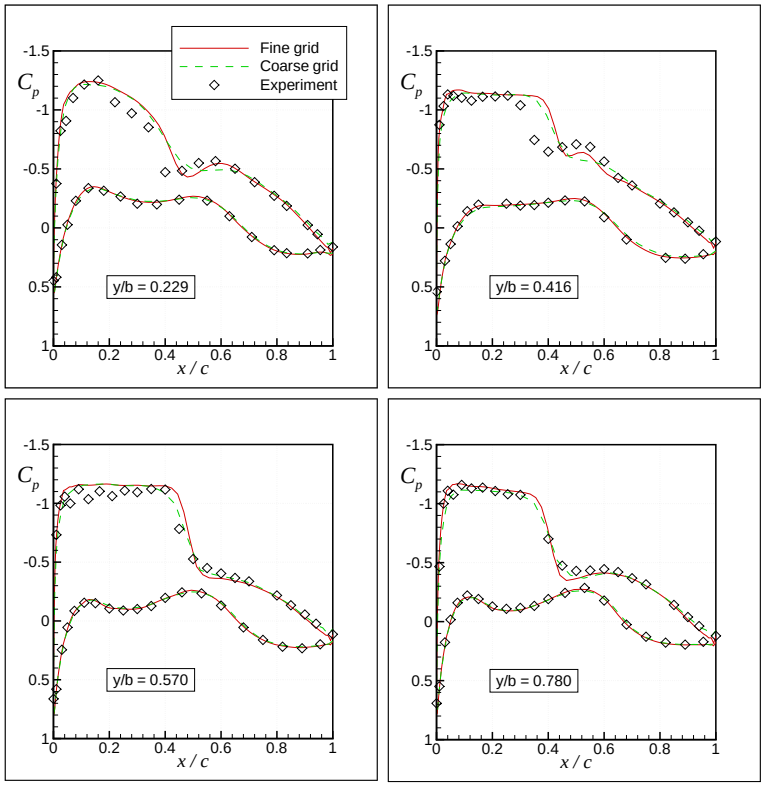


Fig. 11 Pressure distribution with k–ω model for typical wing/body/pylon/nacelle ($M_\infty = 0.80$, $Re_\infty = 10.16 \cdot 10^6$, and $\alpha = 2.2°$).

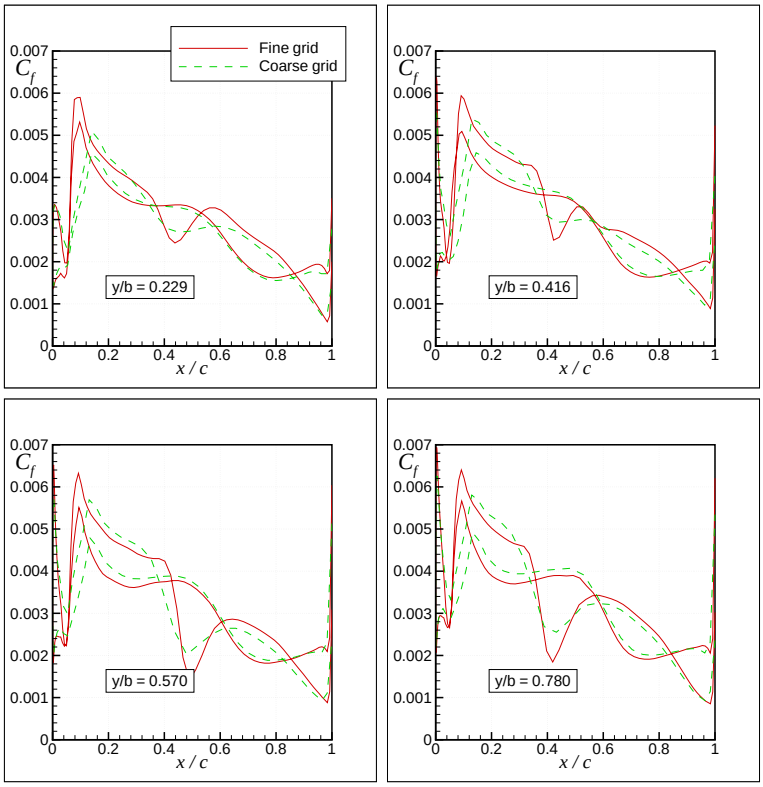


Fig. 12 Skin-friction coefficient with k–ω model for typical wing/body/pylon/nacelle ($M_\infty = 0.80$, $Re_\infty = 10.16 \cdot 10^6$, and $\alpha = 2.2°$).

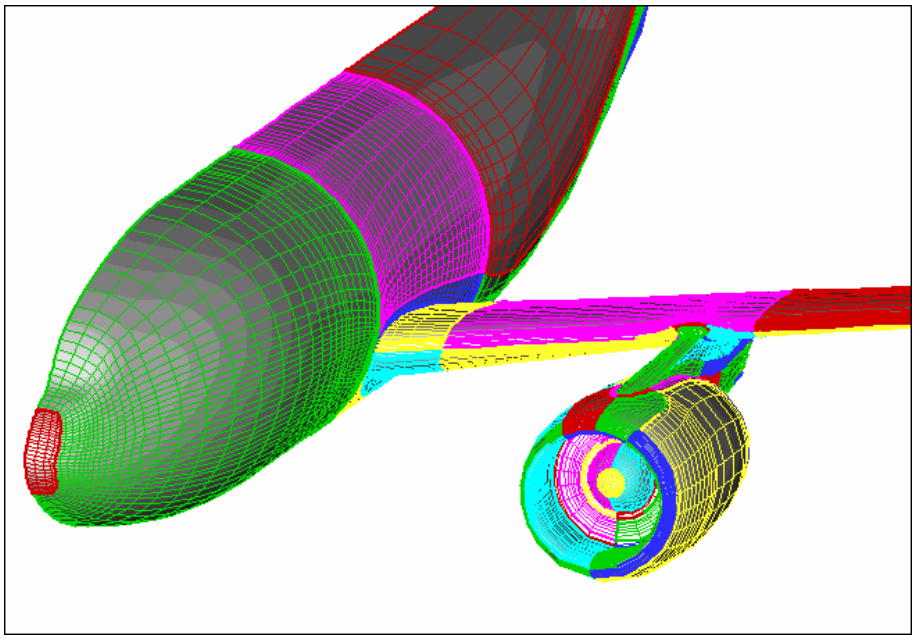


Fig. 13 Impression of coarse surface grid for typical wing/body with turbofan engine.

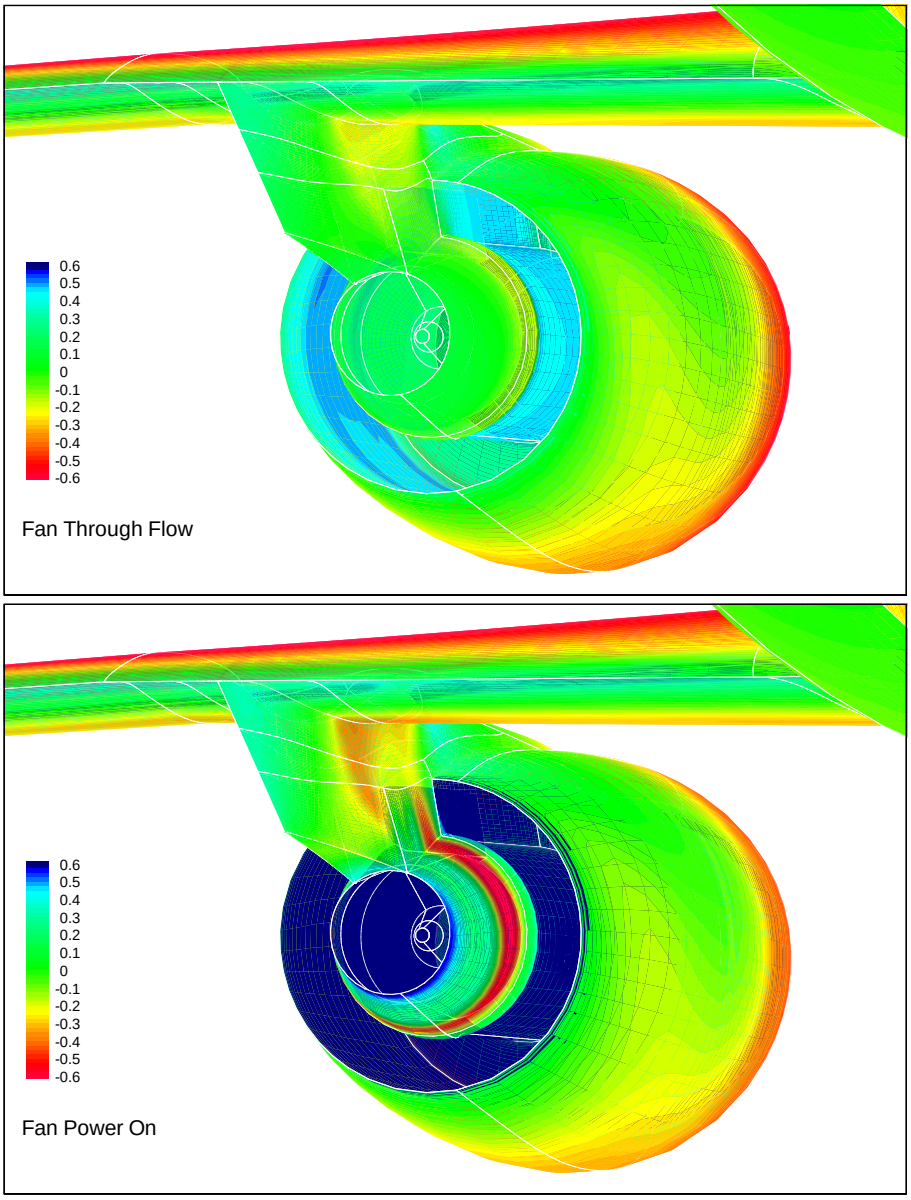


Fig. 14 Pressure distribution for typical wing/body with turbofan engine at fan 'through flow' and fan 'power on' conditions ($C_L = 0.50$, $M_\infty = 0.75$, $Re_\infty = 4.3 \cdot 10^6$).

5 Conclusions

The k–ω turbulence model has been implemented in the compressible, Reynolds-averaged Navier–Stokes flow solver ENSOLV, which is part of the NLR multi-block flow-simulation system ENFLOW.

- In order to resolve the unphysical free-stream dependency of the k–ω model, the TNT set of diffusion coefficients is used, as recently derived (Ref. 4), which is valid throughout the complete boundary layer. Thus, a blending function and therefore also the wall distance is not required, making the model suitable for complex configurations.

- The numerical accuracy is strongly improved by rewriting the k–ω model in a k–τ formulation (with $\tau = 1/\omega$). The dependency on the grid resolution in the direction normal to the surface has been shown to be weak (requiring only 10 grid cells to obtain the correct skin-friction level, within plotting accuracy, for a flat plate).

- The ENSOLV convergence rates of computations with the k–ω turbulence model are comparable to those of computations with algebraic turbulence models. It is not necessary to treat the source terms implicitly. Instead, a separate time step has been used for the turbulence equations, based on their own convective eigenvalue, and scaling for high-aspect-ratio grid cells has been applied to the fourth-order artificial diffusion.

- ENSOLV computations for transport-type aircraft (wing/body/pylon/nacelle configurations), with 4 to 5 million grid cells, require in the order of 500 to 1000 fine-grid iterations (i.e. 50 to 100 multi-grid cycles).

6 References

1. Gould, A., Courty, J.-C., Sillen, M., Elsholz, E., and Abbas, A., "The AVTAC Project – A Review of European Aerospace CFD", ECCOMAS 2000, Barcelona, 11–14 September 2000.
2. Wilcox, D.C., "Reassessment of the Scale-Determining Equation for Advanced Turbulence Models", *AIAA Journal*, Vol. 26, No. 11, 1988, pp. 1299–1310.
3. Menter, F.R., "Zonal Two Equation k–ω Turbulence Models for Aerodynamic Flows", AIAA Paper 93-2906, 1993.
4. Kok, J.C., "Resolving the Dependence on Free-Stream Values for the k–ω Turbulence Model", NLR-TP-99295, 1999 (to be published in the AIAA Journal 2000).
5. Wilcox, D.C., "A Two-Equation Turbulence Model for Wall-Bounded and Free-Shear Flows", AIAA Paper 93-2905, 1993.
6. Kalitzin, G., Gould, A.R.B., and Benton, J.J., "Application of Two-Equation Turbulence Models in Aircraft Design", AIAA Paper 96-0327, 1996.
7. Kok, J.C., Boerstoel, J.W., Kassies, A., and Spekreijse. S.P., "A Robust Multi-Block Navier–Stokes Flow Solver for Industrial Applications", Third ECCOMAS CFD Conference, Paris, September 1996 (NLR TP 96323).
8. Kok, J.C., *An Industrially Applicable Solver for Compressible, Turbulent Flows*, Ph.D. dissertation, Delft University of Technology, 1998.
9. Boerstoel, J.W., Kassies, A., Kok, J.C., and Spekreijse. S.P., "ENFLOW, a Full-Functionality System of CFD Codes for Industrial Euler/Navier–Stokes Flow Computations", NLR-TP-96286, 1996 (presented at the 2nd Int. Symp. on Aeron. Science and Techn., Jakarta, 1996).
10. Spekreijse, S.P. and Boerstoel, J.W., "Multi-Block Grid Generation", Von Karman Institute for Fluid Dynamics (VKI), 27^{th} CFD Course, Lecture Series 1996-02, 1996 (NLR TP 96338).
11. Spekreijse, S.P. and Kok, J.C., "Semi-Automatic Domain Decomposition Based on Potential Theory", submitted to 7^{th} International Conference on Numerical Grid Generation in Computational Field Simulations, Whistler, Canada, September 2000.
12. Cazalbou, J.B., Spalart, P.R., and Bradshaw, P., "On the Behaviour of Two-Equation Models at the Edge of a Turbulent Region", *Physics of Fluids*, Vol. 6, No. 5, 1994, pp. 1797–1804.
13. Wilcox, D.C., *Turbulence Modeling for CFD*, 2nd ed., DCW Industries, Inc., La Cañada, California, 1998.
14. Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes" AIAA Paper 81-1259, 1981.
15. Swanson R.C. and Turkel E., "On Central-Difference and Upwind Schemes", *Journal of Computational Physics*, Vol. 101, 1992, pp. 292–306.

16. Martinelli, L., *Calculations of Viscous Flows with a Multigrid Method*, Ph.D. dissertation, Princeton University, 1987.
17. Cook, P., McDonald, M., and Firmin, M., “Airfoil RAE 2822 – Pressure Distributions and Boundary Layer Wake Measurements”, AGARD AR 138, 1979.