



# Implementation, Realization and an Effective Solver of Two-Equation Turbulence Models

S. Langer<sup>1</sup> · R. C. Swanson<sup>2</sup>

Received: 20 May 2021 / Revised: 6 October 2023 / Accepted: 19 October 2023 /  
Published online: 22 November 2023  
© The Author(s) 2023

## Abstract

Currently, when the Reynolds-Averaged Navier–Stokes (RANS) equations are solved using turbulence modelling, most often the one-equation model of Spalart and Allmaras is used. Then, it is only necessary to solve the RANS equations in conjunction with a single transport equation for modeling turbulence. For this model, considerable assessment and analysis has been performed, allowing the possibility of a reliable solution method for an eddy viscosity required to compute the Reynolds stresses in the RANS equations. Such evaluation along with analysis has not been performed for similar performance with two-equation models of the  $k$ - $\omega$  type. The primary objective of this paper is to present and discuss the components of an effective numerical algorithm for solving the RANS equations and the two transport equations of  $k$ - $\omega$  type turbulence models. All the important details of the turbulence model as actually implemented are given, which is sometimes not done in various papers considering such modeling. The viability and effectiveness of this solution algorithm are demonstrated by solving both two-dimensional and three-dimensional aerodynamic flows. In all applications, a linear rate of convergence without oscillations or other evidence of unstable behavior is observed. This behavior is also particularly true when the proposed algorithm is applied to systematically refined mesh sequences, which is generally not observed with algorithms solving more than one transport equation. Thus numerical integration errors are systematically reduced, allowing for a significantly more reliable assessment of the effectiveness of the model itself. Additionally, in this paper analysis of the solution algorithm, including linear stability, is also performed for a particular flow problem.

**Keywords** Navier–Stokes equations · Two-equation turbulence models · Solution algorithms

---

✉ S. Langer  
stefan.langer@dlr.de

<sup>1</sup> German Aerospace Center, Member of the Helmholtz Association, Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7, 38108 Brunswick, Germany

<sup>2</sup> DRA, Computational Aerosciences Branch, NASA Langley Research Center, Hampton, VA 23681, USA

## 1 Introduction

In the field of Computational Fluid Dynamics (CFD) there has been a significant effort to develop numerical algorithms for solving the RANS equations in conjunction with turbulence models. This effort has continued with the objective to extend the viability of CFD to perform a complete design of complex aircraft configurations. Thus, the convergence and reliability of a solver must be verified. Extensive data assessments of solution accuracy are clearly required. Prominent examples of these data assessments are the numerous Drag Prediction (see for example [1–4]) and High Lift Prediction Workshops (see for example [5–7]), which are carried out regularly.

Assertions about uncertainties and errors from computed numerical data are clearly a requirement to use data obtained by numerical simulations in the aerodynamic certification processes. When designing new aircraft using numerical data, this data must be proven to be reliable. Using past AIAA workshops as a guide, different CFD computer codes typically yield a wide spread in results, even when ostensibly solving the same equations. Reasons for the wide spread are difficult to determine. Considering the fact that all participants assume that they solve the same equations with a consistent and convergent method, differences in the results must disappear as the degrees of freedom increase. For some codes and some test cases this behavior can actually be observed. However, for simulations at the border of the flight envelope or if two-equation turbulence models are used, there are often significant deviations in the results shown at the workshops. Even statistical analyses are carried out to quantify the uncertainties [2–4]. Since convergence histories are rarely shown for these applications, this suggests that convergence may not be achieved; and hence, the results may be severely modified by numerical errors.

Significant improvements have been achieved in the last several years for the Spalart-Allmaras (SA) turbulence model [8, 9]. The improved solvability of the single transport equation of the model using the SA negative form is a major factor for the substantial increase in using this turbulence model. Considering the results of the AIAA Drag Prediction and High-Lift workshops of the last few years, there are more computer codes appearing using this model that can also converge several orders of magnitude. In some cases, even fully converged solutions (i.e., solutions in which the residuals are reduced to near machine zero) are achieved. Perhaps, this is at least part of the reason for the focus on introducing correction terms (e.g., streamline curvature and rotation effects) in the SA model [10, 11]. Such correction terms can provide improvement in representation of the flow physics.

During roughly the same time period as for the SA model, a number of improvements were also made for two-equation turbulence models of the  $k-\omega$  type (i.e., Wilcox model and Menter SST model). Also, there was some effort to improve the solver for the RANS and two-equation models. There is a need for improvements in the convergence behavior and reliability. Thus, solving the RANS equations with such models is frequently avoided due to high difficulty in achieving reliable convergence. Typically, to converge the RANS equations in combination with  $k-\omega$  type models is a much more difficult task compared to a one-equation model.

Such two-equation turbulence models have been successfully applied to airfoil flows with small to moderate separation (e.g., see Menter [12], Swanson [13]). However, the difficulty of simulating a broad range of separated flows, including onset and extent of separation, still remains with this class of two-equation models, as well as other eddy viscosity models. Due to the demands on the data required for a certification process together with the observations

from the workshops, it is clear that there is still a great need for the development of solution methods for the RANS equations, in particular for two- equation turbulence models.

From the perspective of the authors of this paper, to design such a numerical method providing both sufficiently accurate data together with a quantifiable bound on the error, the following four points are a minimum standard one has to consider:

- (a) The full differential or integral formulation of the equations together with the turbulence modelling equations is given.
- (b) Its exact implementation is documented.
- (c) A solution algorithm that is able to compute for a given number of degrees of freedom a solution free of numerical integration errors.
- (d) The solution algorithm works for systematic mesh refinement, that is, one can obtain mesh converged results.

As soon as one of these criteria is not satisfied, certain doubts about the assertions made from approximate solutions computed with the considered numerical method arise.

From the four criteria mentioned above, obviously the first two criteria are the simplest to satisfy. Straightforwardly, one simply has to write down the actual implementation of the equations. Nothing should be hidden such as cut-off values for certain variables. This fact is mentioned here explicitly, though often not mentioned, because a significant number of implementations use certain strategies to cut-off or to restrict several of the variables. As examples we refer to the technical documentations [14, 15]. A final solution is then generally not examined with respect to activity of such limiting processes. To assess the impact corresponding to such limitings, there must be transparency of the solutions. Since often authors do not discuss or do not even mention these intrusions into the equations, only conjectures about the reasons for discrepancies between the approximate solution and experimental data can be made. One conjecture of relevance is that c) mentioned above is not independent of b).

The reason for this, which is possibly the major issue in constructing a reliable algorithm to approximately solve the RANS equations and the two transport equations of  $k$ - $\omega$  type turbulence models, is to ensure positivity of the turbulence kinetic energy  $k$  and the energy dissipation rate  $\omega$ . Perhaps, the most relevant publications dealing with this topic are [16–19]. They apply an implicit algorithm to solve the RANS equations and the two equations of a  $k$ - $\omega$  model in a loosely coupled manner. The advantage of this approach is that only the Jacobian matrix for the turbulence equations is manipulated such that it is an M-matrix for each time step. Hence, starting with a positive  $k$  and  $\omega$ , all successive values of these two dependent variables remain positive throughout the iteration process. Another idea to ensure positivity of  $k$  and  $\omega$  uses variable substitution (logarithmic form), which was suggested by Ilinca et al. [20]. A number of other finite element methods (FEMs), not only Continuous Petrov-Galerkin methods but also Discontinuous Galerkin (DG) methods, also employ this approach (e.g., [21–23]). In some FEMs realizability conditions are also imposed [21, 22]. As a consequence, it is not obvious if the transformed boundary-value problem is still equivalent to the original one. As discussed in [24], there are possible effects due to smoothness of the solution as well as the modified boundary condition that may have an impact on the resulting boundary-value problem. In this article, we suggest a rather simple algorithm to deal with this positivity issue.

Given the wide range of applications for which solutions to the RANS equations are required, it is desirable that a computer code should meet the following requirements:

- It works for a large variety of parameters defining the boundary-value problem to solve, such as

- a variety of geometries,
- a large number of inflow conditions, which includes a range from very low Mach number to hypersonic flows,
- a broad range of Reynolds numbers,
- a large number of different boundary conditions;
- It works for a broad range of parameters determining the actual solution method, such as
  - variation in CFL number,
  - inner linear solution methods,
  - linear and nonlinear multigrid as well as cycling strategies;
- It always converges to a steady-state solution, if a steady state solution exists;
- It does not show significant loss in convergence rates with systematic mesh refinement studies, for example, an increase in the number of degrees of freedom are considered.

From the point of view of the authors, at this time, the design of a solution method for the RANS equations satisfying all these conditions is an open problem. It can be assumed that various interventions (using cutoff values or restriction of variables) into several solution methods have been incorporated to be in a position to compute steady-state solutions, at least for a small number of problems with a specific choice of parameters.

Considering this background, the primary objective of this paper is to propose a numerical algorithm that efficiently solves the RANS equations in combination with two-equation models of the  $k$ - $\omega$  type on structured and unstructured meshes. Numerical examples demonstrate that the number of nonlinear iterations is comparable to state-of-the-art methods for one-equation models. In this regard, we refer to [25–31], where a similar number of nonlinear iterations is required using similar methods for the one-equation model of Spalart and Allmaras. We illustrate that a reliable behavior of the solution algorithm, that is without oscillations or a deterioration in the final convergence rate is not only possible for basic 2-D test cases but also for 3-D transonic test cases at the design point and for a 3-D high lift configuration. Beyond the pure heuristics and test cases, an analysis is performed in particular for the two-equation turbulence model, which reveals some of the special characteristics of these models and what one needs to consider in order to implement them successfully.

In Sects. 2 and 3 of the paper, the integral form of the RANS equations and the turbulence modeling equations, respectively, are defined. This includes the boundary conditions necessary for a well defined boundary-value problem. Section 4 considers the components of the numerical algorithm, which are the focus of the paper. Then, in Sect. 5, analysis of properties, including linear stability, of the solution algorithm is considered. Both two-dimensional (2-D) and three-dimensional (3-D) examples of applications of the numerical solution algorithm are shown and discussed in Sect. 6. Concluding remarks of the paper are made in Sect. 7.

## 2 Governing Equations of Fluid Flow

To describe flow behavior we consider for the domain  $G \subset \mathbb{R}^m$ ,  $m = 2, 3$ , i.e., an open and connected set, and an interval  $[0, T) \subset \mathbb{R}$ ,  $T > 0$ , the conservative form of the RANS equations is expressed in integral form by

$$0 = \frac{d}{dt} V_G(W)(t) + R_{\partial G}(W)(t), \quad t \in [0, T), \quad (1a)$$

where the integral operators  $V_G$  and  $R_{\partial G}$  are given by

$$V_G(W)(t) := \int_G W(x, t) dx \tag{1b}$$

$$R_{c, \partial G}(W)(t) := \int_{\partial G} \langle f_c(W(y, t)), n(y) \rangle ds(y), \tag{1c}$$

$$R_{v, \partial G}(W)(t) := \int_{\partial G} \langle f_v(W(y, t)), n(y) \rangle ds(y), \tag{1d}$$

$$R_{\partial G} := R_{c, \partial G} - R_{v, \partial G}, \tag{1e}$$

and  $W : G \times [0, T) \rightarrow \mathbb{R}^{m+2}$ ,

$$W(x, t) := (\rho(x, t), \rho(x, t)u(x, t), \rho(x, t)E(x, t))^T, \tag{2}$$

denotes the vector field of conserved variables and  $n$  is the unit outward normal on  $\partial G$ . The terms  $f_c$  and  $f_v$  describe the convective and viscous contributions

$$f_c(W) := \begin{pmatrix} \rho u \\ \rho u_1 u + p e_1 \\ \vdots \\ \rho u_m u + p e_m \\ \rho H u \end{pmatrix},$$

$$f_v(W) := \begin{pmatrix} 0 \\ \tau_1(W) \\ \vdots \\ \tau_m(W) \\ \theta(W) \end{pmatrix}, \quad m = 2, 3.$$

Here  $e_i$  is the  $i$ th unit vector. The quantities  $\rho, u = (u_1, \dots, u_m)^T, E$  and

$$H := E + p/\rho \tag{3}$$

are the density, the velocity, the specific total energy, and the enthalpy of the fluid. The equation of state

$$p = (\gamma - 1)\rho \left( E - \frac{\|u\|_2^2}{2} \right) \tag{4}$$

defines the pressure  $p$ , and  $\gamma$  is the gas dependent ratio of specific heats, which is 1.4 for air. Assuming that an effective viscosity  $\mu_{\text{eff}} := \mu_{\text{eff}}(W) = \mu_{\text{eff}}(W(x, t))$  is given and using Stoke’s hypothesis, the bulk viscosity vanishes and the second coefficient of viscosity  $\lambda = -2/3\mu_{\text{eff}}$ . Then, the viscous stress tensor is given by

$$\tau(W) := \mu_{\text{eff}}\mathcal{S} + \lambda \operatorname{div}(u)\operatorname{Id} = 2\mu_{\text{eff}}\bar{\mathcal{S}}, \quad \bar{\mathcal{S}} := \mathcal{S} - \frac{1}{3} \operatorname{div}(u)\operatorname{Id}, \tag{5}$$

and  $\mathcal{S}$  denotes the mean strain-rate tensor, which is given by the symmetric part of the total derivative of flow velocity vector  $u$ ,

$$\mathcal{S} := \frac{1}{2} \left( \frac{du}{dx} + \left( \frac{du}{dx} \right)^T \right), \quad \text{i.e.,}$$

$$S_{ij} = \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right). \tag{6}$$

Throughout the paper  $\text{Id}(x) = x$  denotes the identity operator. The viscous flux term for the energy equation is given by

$$\theta(W) := \tau(W)u + q(W), \tag{7a}$$

$$q(W) := \kappa_{\text{eff}} \text{grad } T. \tag{7b}$$

The effective viscosity  $\mu_{\text{eff}}$  and effective conductivity  $\kappa_{\text{eff}}$  are computed by

$$\mu_{\text{eff}} := \mu_l + \mu_t, \quad \kappa_{\text{eff}} := \kappa_l + \kappa_t, \tag{8}$$

and the laminar viscosity is given by Sutherland’s law

$$\mu_l(W) := \mu_{l,\infty} \left( \frac{T}{T_\infty} \right)^{3/2} \frac{T_\infty + \bar{T}}{T + \bar{T}}, \quad \mu_{l,\infty} := \frac{\rho_\infty u_\infty L}{Re}, \tag{9}$$

$$\begin{aligned} \kappa_l(W) &:= \frac{c_p \mu_l(W)}{Pr_l} \quad \text{and} \\ c_p &:= \Re \frac{\gamma}{\gamma - 1}, \end{aligned} \tag{10}$$

whereby  $T_\infty > 0$ ,  $\rho_\infty > 0$  and  $u_\infty > 0$  denote some constant reference temperature, density and velocity,  $L > 0$  is some constant reference length scale,  $Re > 0$  is the corresponding Reynolds number,  $\bar{T} := 110.4^\circ\text{K}$  is Sutherland’s constant,  $\Re$  is the universal gas constant and the laminar Prandtl number is given by  $Pr_l := 0.72$ .

In this article, we restrict consideration to linear, two equation turbulence models, represented by differential or integral equations stated in Sect. 3. The solutions of these equations reveal additional quantities in the considered fluid. These occurring variables extend the degrees of freedom given by the conservative variables  $W$  by a further unknown function

$$W_t : G \times [0, T] \rightarrow \mathbb{R}^2.$$

The additional variables are used to determine the eddy viscosity,

$$\begin{aligned} \mu_t &= \mu_t(W_t(x, t), W(x, t)) \\ &\geq 0 \quad \text{for all } (x, t) \in G \times [0, T], \end{aligned}$$

required for (8). Given the eddy viscosity  $\mu_t$  the turbulent thermal conductivity is described by the algebraic relation

$$\kappa_t := c_p \frac{\mu_t}{Pr_t}, \quad Pr_t := 0.9. \tag{11}$$

The dependent variables of the turbulence models are the turbulence kinetic energy per unit mass  $k$  and the dissipation rate  $\omega$  of the turbulence kinetic energy. Before we state the two transport equations for  $k$  and  $\omega$ , we define  $t = (t_{ij})_{1 \leq i, j \leq m}$  and the Reynolds stress tensor  $\tau^{\text{RS}} = (\tau_{ij}^{\text{RS}})_{1 \leq i, j \leq m}$ , using the strain-rate tensors  $S$  and  $\bar{S}$ , given in (6) and (5), and the mean molecular stress tensor,  $t = (t_{ij})_{1 \leq i, j \leq m}$ . Then,

$$\begin{aligned} t &= 2\mu_{\text{eff}}\bar{S}, \quad \rho\tau^{\text{RS}} \\ &= 2\mu_t\bar{S} - \frac{2}{3}\rho k\text{Id}. \end{aligned} \tag{12}$$

Additionally, we define the mean rotation tensor  $\Omega$ , which is the skew-symmetric part of the total derivative of flow velocity  $u$ ,

$$\begin{aligned} \Omega &:= \frac{1}{2} \left( \frac{du}{dx} - \left( \frac{du}{dx} \right)^T \right), \quad \text{i.e. } \Omega_{ij} \\ &= \frac{1}{2} \left( \frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i} \right). \end{aligned} \tag{13}$$

### 3 Two-Equation Turbulence Models of $k$ - $\omega$ Type

The principal elements of the  $k$ - $\omega$  type turbulence models being considered are described in this section. The emphasis is on the 1988 and 2006 Wilcox models and the 2003 Menter Shear Stress Transport (SST) model. Presentation and extensive discussion of these models is given in references [32–34] and [12, 35–37]. An additional resource providing information on these and other turbulence models is NASA’s Turbulence Modeling Resource (TMR) website [38]. Important details of these models, such as parameters and closure coefficients, can be found in the TMR or in the original publications.

#### 3.1 Wilcox $k$ - $\omega$ Model

In this article we consider  $k$ - $\omega$  models of the form

$$\frac{\partial (\rho k)}{\partial t} + \text{div} (\rho k u) = \text{div} ((\mu_l + \sigma_k \mu_t) \text{grad } k) + \rho Q_{k,(k,\omega)}, \tag{14a}$$

$$\frac{\partial (\rho \omega)}{\partial t} + \text{div} (\rho \omega u) = \text{div} ((\mu_l + \sigma_\omega \mu_t) \text{grad } \omega) + \rho Q_{\omega,(k,\omega)}. \tag{14b}$$

The presented and used 1988 and 2006 models of Wilcox are named as “Wilcox1988” and “Wilcox2006” on NASA’s (TMR) website [38]. The eddy viscosity in these models is computed by

$$\mu_t = \mu_t(W, k, \omega) = \rho k / z(W, k, \omega) \tag{15}$$

with a given function  $z$ . For example, in the 1988 and 2006 models of Wilcox, we have

$$z(W, k, \omega) = \omega, \quad (1988 \text{ model}), \tag{16}$$

$$z(W, k, \omega) = \max \left\{ \omega, C_{\text{lim}} \sqrt{\frac{2\Omega : \Omega}{\beta^*}} \right\}, \quad (2006 \text{ model}). \tag{17}$$

The symbol  $:$  denotes a double dot tensor product. The source terms for the  $k$  and  $\omega$  equations include production ( $Pr$ ) and destruction ( $De$ ) terms. A cross-diffusion term ( $Di$ ) is also included in the source term of the  $\omega$  equation for the 2006 Wilcox model ( $\Psi = 1$ ), the 1988 model does not have such a term ( $\Psi = 0$ ),

$$\begin{aligned} Q_{k,(k,\omega)} &:= \tilde{Pr}_{k,(k,\omega)} - De_{k,(k,\omega)}, \quad Q_{\omega,(k,\omega)} \\ &:= Pr_{\omega,(k,\omega)} - De_{\omega,(k,\omega)} + \Psi Di_{\omega,(k,\omega)}, \end{aligned} \tag{18}$$

where

$$Pr_{k,(k,\omega)} := \tau^{\text{RS}} : \frac{du}{dx}, \quad De_{k,(k,\omega)} := \beta^* k \omega \tag{19a}$$

$$Pr_{\omega,(k,\omega)} := \alpha \frac{\omega}{k} \tau^{RS} : \frac{du}{dx}, \quad De_{\omega,(k,\omega)} := \beta \omega^2. \tag{19b}$$

$$Di_{\omega,(k,\omega)} := \sigma_d \frac{1}{\omega} (\text{grad } k, \text{grad } \omega). \tag{19c}$$

In the 1988 Wilcox model, the production term of the  $k$  equation is generally limited according to

$$\tilde{Pr}_{k,(k,\omega)} := \min \{ Pr_{k,(k,\omega)}, 20De_{k,(k,\omega)} \}, \tag{20}$$

so as to prevent very large  $P_k$  during the initial phases of a calculation. An example to demonstrate the impact and influence of (20) on possible solutions was demonstrated in [24, Section 7]. This change is not included in the 2006 Wilcox model due to the introduction of a stress limiter. This stress limiter can have a significant effect on a shock location. The effect of varying the stress limiter on the shock location for the RAE 2822 airfoil (Case 10) is shown in Swanson and Rossow [39].

### 3.2 Menter $k-\omega$ Model (SST Model)

The presented and used model of Menter is named as "SST-2003" on NASA's (TMR) website [38] with the only difference that (20) is used, that is the 20 was not replaced by 10. In the 2003 Menter SST model the constants  $\sigma_k$  and  $\sigma_\omega$  in the two transport equations (14a) and (14b) are replaced with blending functions  $\Phi_k$  and  $\Phi_\omega$ . These blending functions depend on the values  $\sigma_k$  and  $\sigma_\omega$  in the  $k-\omega$  and  $k-\epsilon$  models. The source terms are also changed due to blending functions for the  $k-\omega$  and  $k-\epsilon$  models. The eddy viscosity is determined from (15) with  $z(W, k, \omega)$  given by

$$\begin{aligned} z(W, k, \omega) &= \frac{1}{\min \left\{ \frac{1}{\omega}, \frac{a_1}{F_2 \sqrt{2S:S}} \right\}} \\ &= \max \left\{ \omega, \frac{F_2 \sqrt{2S:S}}{a_1} \right\}. \end{aligned} \tag{21}$$

Here,  $F_2$  is a blending function defined by

$$F_2 := \tanh \left( (\max \{ \Gamma_1, 2\Gamma_3 \})^2 \right) \tag{22}$$

where

$$\begin{aligned} \Gamma_1 &:= \frac{C_{\Gamma_1} \nu_l}{d^2 \omega}, \\ \Gamma_3 &:= \frac{\sqrt{k}}{\beta^* \omega d}, \quad \beta^* := 0.09, \end{aligned} \tag{23}$$

and  $d$  is the distance to the closest no-slip wall, and  $\nu_l$  is the kinematic viscosity. According to [12], the constants are  $a_1 = 0.31$  and  $C_{\Gamma_1} = 500$ . The source terms, having the same components as in (18), are given by

$$\begin{aligned} Q_{k,SST} &= \tilde{Pr}_{k,SST} - De_{k,SST}, \\ Q_{\omega,SST} &= Pr_{\omega,SST} - De_{\omega,SST} + Di_{\omega,SST}, \end{aligned} \tag{24}$$

where

$$Pr_{k,SST} := \tau^{RS} : \frac{du}{dx}, \quad De_{k,SST} := \beta^* k \omega, \tag{25a}$$



$$Pr_{\omega, \text{SST}} := \Phi_{\gamma} \frac{1}{\nu_t} \tau^{\text{RS}} : \frac{du}{dx}, \quad De_{\omega, \text{SST}} := \Phi_{\beta} \omega^2, \tag{25b}$$

$$Di_{\omega, \text{SST}} := 2(1 - F_1) \sigma_{\omega_2} \frac{1}{\omega} \langle \text{grad } k, \text{grad } \omega \rangle. \tag{25c}$$

The blending of the SST model is controlled by a function  $\Phi = \Phi(x; \varepsilon_1, \varepsilon_2)$ . This function is designed to detect the edge of the boundary layer, such that the SST model behaves inside the boundary layer like a  $k$ - $\omega$  model and outside like a  $k$ - $\varepsilon$  model, exploiting the convex combination  $\Phi : [0, 1] \rightarrow [\varepsilon_1, \varepsilon_2]$ ,

$$\Phi(F_1; \varepsilon_1, \varepsilon_2) := F_1 \varepsilon_1 + (1 - F_1) \varepsilon_2. \tag{26}$$

### 3.3 Simplifications of $k$ - $\omega$ Models

Although these models are formulated in their compressible form (14), their actual implementation and usage is often based on their incompressible version. In fact, this occurs even when they are used with respect to compressible flow. Then, one assumes

$$\text{div}(u) = 0, \tag{27}$$

yielding for the equation of mass conservation  $0 = \frac{\partial \rho}{\partial t} + \langle \text{grad } \rho, u \rangle$ , and hence

$$\frac{\partial(\rho k)}{\partial t} + \text{div}(\rho k u) = \rho \left( \frac{\partial k}{\partial t} + \text{div}(k u) \right), \tag{28}$$

$$\frac{\partial(\rho \omega)}{\partial t} + \text{div}(\rho \omega u) = \rho \left( \frac{\partial \omega}{\partial t} + \text{div}(\omega u) \right). \tag{29}$$

Inserting (28) and (29) into (14) gives

$$\frac{\partial k}{\partial t} + \text{div}(k u) = \frac{1}{\rho} \text{div}((\mu_l + \sigma_k \mu_t) \text{grad } k) + Q_{k,(k,\omega)}, \tag{30a}$$

$$\frac{\partial \omega}{\partial t} + \text{div}(\omega u) = \frac{1}{\rho} \text{div}((\mu_l + \sigma_{\omega} \mu_t) \text{grad } \omega) + Q_{\omega,(k,\omega)}, \tag{30b}$$

and the independent variables  $W_t = (\rho k, \rho \omega)$  are replaced by  $W_t = (k, \omega)$ . To be consistent using assumption (27) we have

$$\rho \tau = 2 \mu_t \bar{S}. \tag{31}$$

In a second step, assumption (27) can be integrated into (31). This yields  $\bar{S} = S$  and finally

$$\rho \tau = 2 \mu_t S. \tag{32}$$

#### Remark

*Often, it is not indicated if the assumption (27) is included into the formulation of the turbulence model, that is, if (31) or (32) is used for the formulation of the production terms. For the implementation considered here we chose (31).*

### 3.4 Integral Form of $k$ - $\omega$ Models

When integrating the diffusive terms in (30) over a control volume, they cannot be rewritten as a surface integral because of division with density  $\rho$ . Hence, a further approximation is

introduced,

$$\frac{1}{\rho} \operatorname{div} ((\mu_l + \sigma_k \mu_t) \operatorname{grad} k) \approx \operatorname{div} ((\nu_l + \sigma_k \nu_t) \operatorname{grad} k), \tag{33a}$$

$$\frac{1}{\rho} \operatorname{div} ((\mu_l + \sigma_\omega \mu_t) \operatorname{grad} \omega) \approx \operatorname{div} ((\nu_l + \sigma_\omega \nu_t) \operatorname{grad} \omega). \tag{33b}$$

Integration of (30) using the approximation of (33) gives the integral equation

$$\begin{aligned} &V_G(Q_{(k,\omega)}(W_t, W))(t) \\ &= \frac{d}{dt} V_G(W_t)(t) \\ &\quad + R_{\partial G, (k,\omega)}(W_t, W)(t), \end{aligned} \tag{34}$$

where the integral operators are  $R_{\partial G, (k,\omega)} := R_{c, \partial G, (k,\omega)} - R_{v, \partial G, (k,\omega)}$ , and

$$\begin{aligned} &R_{c, \partial G, (k,\omega)}(W_t, W)(t) \\ &:= \int_{\partial G} \langle f_{c, (k,\omega)}(W_t(y, t), W(y, t)), n(y) \rangle ds(y), \\ &R_{v, \partial G, (k,\omega)}(W_t, W)(t) \\ &:= \int_{\partial G} \langle f_{v, (k,\omega)}(W_t(y, t), W(y, t)), n(y) \rangle ds(y). \end{aligned}$$

Here the convective  $f_{c, (k,\omega)}$  and viscous  $f_{v, (k,\omega)}$  contributions as well as the source terms  $Q_{(k,\omega)}$  are summarized by

$$f_{c, (k,\omega)}(W_t, W) := \begin{pmatrix} ku \\ \omega u \end{pmatrix}, \tag{35a}$$

$$f_{v, (k,\omega)}(W_t, W) := \begin{pmatrix} (\nu_l + \sigma_k \nu_t) \operatorname{grad} k \\ (\nu_l + \sigma_\omega \nu_t) \operatorname{grad} w \end{pmatrix}, \tag{35b}$$

$$Q_{(k,\omega)}(W_t, W) := \begin{pmatrix} Pr_{k, (k,\omega)} - De_{k, (k,\omega)} \\ Pr_{\omega, (k,\omega)} - De_{\omega, (k,\omega)} + Di_{\omega, (k,\omega)} \end{pmatrix}. \tag{35c}$$

### 3.5 Nondimensionalization

For a numerical implementation a suited scaling and nondimensionalization is often crucial. Due to the no-slip wall boundary condition (see [24]) for  $\omega$ ,

$$\lim_{h \rightarrow 0^+} \omega(x - hn(x)) h^2 = \frac{6\nu_l}{\beta}, \quad x \in \partial D_{\text{no-slip}}, \tag{36}$$

typically the values for  $\omega$  related to  $k$  and the other conservative variables (2) may differ in orders of magnitude. For this reason, a scaling of  $\omega$  is introduced.

Throughout this subsection, we denote dimensional variables using the sign  $\hat{\cdot}$ , for example dimensional density is denote by  $\hat{\rho}$ . To nondimensionalize  $\hat{k}$  and  $\hat{\omega}$  we choose as reference values

$$k_{\text{ref}} = u_{\text{ref}}^2 \quad \text{and} \quad \omega_{\text{ref}} = \frac{u_{\text{ref}}}{L_{\text{ref}}}. \tag{37}$$

Then the nondimensional variables are given by

$$k = \frac{\hat{k}}{u_{\text{ref}}^2} \quad \text{and}$$

$$\begin{aligned} \omega &= \frac{\hat{\omega}}{\omega_{sc}\omega_{ref}}, \\ \omega_{sc} &= \frac{Re}{\sqrt{\gamma}M_\infty L}, \end{aligned} \tag{38}$$

where  $\omega_{sc}$  denotes the additional scaling for  $\omega$ . Such scaling may be motivated by the following argumentation. Boundary condition (36) is realized by

$$\begin{aligned} \omega_{no-slip}(p_{i,bdry}) &= \frac{60\nu_l(W(p_{i,bdry}))}{\beta \|p_{i,bdry} - p_{i,n}\|_2^2}, \\ p_{i,bdry} &\in \partial D_{no-slip}, \end{aligned} \tag{39}$$

where  $p_{i,bdry}$  denotes the point on the no-slip wall and  $p_{i,n}$  the closest, next discrete point in direction  $-n(p_{i,bdry})$ . Obviously, with respect to possible mesh refinements the distance to the closest wall satisfies

$$d_{h,i} \approx \|p_{i,bdry} - p_{h,i,n}\|_2 \rightarrow 0, \quad h \rightarrow 0,$$

and hence possibly numerical instabilities emerge due to a significant increase in the boundary values  $\omega_{no-slip}$ . The above choice of  $\omega_{sc}$  serves a some normalization such that the largest values for  $\omega$  are in a range of the other variables. Naturally, a choice of  $\omega_{sc} = 1$  or any other reasonable choice is also possible.

To obtain an equivalent system of equations, this scaling parameter needs to be correctly incorporated in all terms and equations. For a detailed study we refer to [40, 41], here we just report the results. For example, to obtain the scaled, nondimensional version of the eddy viscosity (15) together with (21) for the SST model we have

$$\mu_t = \frac{1}{\omega_{sc}} \rho k \min \left\{ \frac{1}{\omega}, \frac{\omega_{sc} a_1}{F_2 \sqrt{2\Omega} : \Omega} \right\}.$$

Then, introducing (37) into (35a), (35b) and (35c), we compute

$$f_{c,(k,\omega)}(\hat{W}_t, \hat{W}) = u_{ref} \begin{pmatrix} k_{ref} & 0 \\ 0 & \omega_{ref}\omega_{sc} \end{pmatrix} f_{c,(k,\omega)}(W_t, W), \tag{40}$$

$$f_{v,(k,\omega)}(\hat{W}_t, \hat{W}) = u_{ref} \begin{pmatrix} \frac{k_{ref}}{\omega_{sc}} & 0 \\ 0 & \omega_{ref} \end{pmatrix} \tilde{f}_{v,(k,\omega)}(W_t, W), \tag{41}$$

where

$$\tilde{f}_{v,(k,\omega)}(W_t, W) := \left( \begin{matrix} (\Gamma(T) + \sigma_k \nu_t) \text{grad } k \\ (\Gamma(T) + \sigma_\omega \nu_t) \text{grad } \omega \end{matrix} \right), \quad \Gamma(T) = T^{3/2} \left( \frac{1 + C_{suth}}{T + C_{suth}} \right),$$

and  $C_{suth}$  denotes Sutherland’s constant. For the source terms (35c) we obtain

$$Pr_{k,(k,\omega)}(\hat{W}_t, \hat{W}) = \frac{u_{ref}^3}{L_{ref}\omega_{sc}} Pr_{k,(k,\omega)}(W_t, W), \tag{42a}$$

$$De_{k,(k,\omega)}(\hat{W}_t, \hat{W}) = \frac{u_{ref}^3 \omega_{sc}}{L_{ref}} De_{k,(k,\omega)}(W_t, W), \tag{42b}$$

$$Pr_{\omega,(k,\omega)}(\hat{W}_t, \hat{W}) = \frac{u_{ref}^2}{L_{ref}^2} Pr_{\omega,(k,\omega)}(W_t, W), \tag{42c}$$

$$De_{\omega,(k,\omega)}(\hat{W}_t, \hat{W}) = \frac{u_{ref}^2 \omega_{sc}^2}{L_{ref}^2} De_{\omega,(k,\omega)}(W_t, W), \tag{42d}$$

$$Di_{\omega,(k,\omega)}(\hat{W}_t, \hat{W}) = \frac{u_{ref}^2}{L_{ref}^2} Di_{\omega,(k,\omega)}(W_t, W). \tag{42e}$$

Introducing the mapping  $g : D \rightarrow \hat{D}, x \mapsto L_{ref}x$ , which maps the computational domain  $D$  to its physical domain  $\hat{D}$ , and application of substitution formulae

$$\int_{\hat{D}} v(x)dx = L_{ref}^m \int_D v(g(y))dy, \tag{43a}$$

$$\int_{\partial \hat{D}} \langle v(y), n(y) \rangle ds(y) = L_{ref}^{m-1} \int_{\partial D} \langle v(g(y)), n(y) \rangle ds(y). \tag{43b}$$

integral equation (34) turns into using (40), (41) and (42a)– (42e),

$$\begin{aligned} & \left( \begin{array}{c} \frac{L_{ref}^2 u_{ref}^3}{\omega_{sc}} \int_D Pr_{k,(k,\omega)} dy - L_{ref}^2 u_{ref}^3 \omega_{sc} \int_D De_{k,(k,\omega)} dy \\ L_{ref} u_{ref}^2 \int_D Pr_{\omega,(k,\omega)} dy - L_{ref} u_{ref}^2 \omega_{sc}^2 \int_D De_{\omega,(k,\omega)} dy + L_{ref} u_{ref}^2 \int_D Di_{\omega,(k,\omega)} dy \end{array} \right) \\ &= \begin{pmatrix} L_{ref}^2 u_{ref}^3 & 0 \\ 0 & \omega_{sc} L_{ref} u_{ref}^2 \end{pmatrix} \left\{ \frac{d}{dt} V_D(W_t)(t) + R_{c,\partial G,(k,\omega)}(W_t, W)(t) \right\} \\ & - \begin{pmatrix} \frac{1}{\omega_{sc}} L_{ref}^2 u_{ref}^3 & 0 \\ 0 & L_{ref} u_{ref}^2 \end{pmatrix} \int_{\partial D} \langle \tilde{f}_{v,(k,\omega)}(W_t, W), n \rangle ds(y). \end{aligned}$$

Multiplication of the whole system with the diagonal matrix  $\text{diag} \left( \frac{1}{L_{ref}^2 u_{ref}^3}, \frac{1}{\omega_{sc} L_{ref} u_{ref}^2} \right)$  gives the mathematically equivalent system of equations

$$\begin{aligned} & \left( \begin{array}{c} \omega_{sc}^{-1} \int_D Pr_{k,(k,\omega)} dy - \omega_{sc} \int_D De_{k,(k,\omega)} dy \\ \omega_{sc}^{-1} \int_D Pr_{\omega,(k,\omega)} dy - \omega_{sc} \int_D De_{\omega,(k,\omega)} dy + \omega_{sc}^{-1} \int_D Di_{\omega,(k,\omega)} dy \end{array} \right) \\ &= \frac{d}{dt} V_D(W_t)(t) + R_{c,\partial G,(k,\omega)}(W_t, W)(t) \\ & - \begin{pmatrix} \omega_{sc}^{-1} & 0 \\ 0 & \omega_{sc}^{-1} \end{pmatrix} \int_{\partial D} \langle \tilde{f}_{v,(k,\omega)}(W_t, W), n \rangle ds(y). \end{aligned} \tag{44}$$

The system of equations (44) is the actual system of equations which is implemented and solved. For a closed formulation of the complete boundary-value problem we refer to [24].

## 4 Solution Algorithm

### 4.1 Multigrid and Implicit Smoother

The discretization strategy followed is based on a finite-volume formulation. The inviscid terms are discretized using a central difference scheme with an added matrix-valued artificial viscosity [29]. A first-order upwind scheme is applied to the convective part of the turbulent flow equations. Gradients required for the viscous terms and for the source terms are computed using a Green-Gauss formulation (see [42]). For a detailed description of the discretization strategy and boundary conditions applied, we refer to [29, 30, 40]. To clarify that we deal now with the discretized equations we use bold letters from now on.

The discretization of the mean flow equations (1) together with the turbulent flow equations (34) yields the system of ordinary differential equations

$$\frac{d}{dt} \begin{pmatrix} \mathbf{W}(t) \\ \mathbf{W}_t(t) \end{pmatrix} = \begin{pmatrix} -\mathbf{M}_{\text{mean}}^{-1} \mathbf{R}_{\text{mean}}(\mathbf{W}(t), \mathbf{W}_t(t)) \\ -\mathbf{M}_{\text{turb}}^{-1} \mathbf{R}_{\text{turb}}(\mathbf{W}(t), \mathbf{W}_t(t)) \end{pmatrix}, \tag{45}$$

where

$$\begin{aligned} \mathbf{M}_{\text{mean}} &:= \text{diag}(\text{diag}(\text{vol}(\Omega_i))) \in \mathbb{R}^{5N_{\text{elem}} \times 5N_{\text{elem}}} \\ \mathbf{M}_{\text{turb}} &:= \text{diag}(\text{diag}(\text{vol}(\Omega_i))) \in \mathbb{R}^{2 \cdot N_{\text{elem}} \times 2 \cdot N_{\text{elem}}} \end{aligned}$$

denote the mass matrix for mean and turbulent flow equations. To approximately solve (45) we assume that the mean flow equations depend only on  $\mathbf{W}$ , and  $\mathbf{W}_t$  acts only as a parameter here, whereas the turbulent flow equations depend only on  $\mathbf{W}_t$ , and  $\mathbf{W}$  acts as a parameter. Hence, we rewrite system (45) as

$$\frac{d}{dt} \mathbf{W}(t) = -\mathbf{M}_{\text{mean}}^{-1} \mathbf{R}_{\text{mean}}(\mathbf{W}(t); \mathbf{W}_t(t)) \tag{46a}$$

$$\frac{d}{dt} \mathbf{W}_t(t) = -\mathbf{M}_{\text{turb}}^{-1} \mathbf{R}_{\text{turb}}(\mathbf{W}_t(t); \mathbf{W}(t)). \tag{46b}$$

Equations (46a) and (46b) are then solved in a loosely coupled manner (see loosely and fully coupled methods [43]).

To approximate a steady-state solution, we apply a nonlinear multigrid method (see [44]), which is called the Full Approximation Scheme (FAS), to the mean flow equation (46a). The turbulent flow equation (46b) is solved in a single grid mode, only. A multistage diagonally implicit Runge–Kutta method is used as smoother (see [40])

$$\begin{aligned} \mathbf{W}^{(0)} &:= \mathbf{W}^n \\ \mathbf{W}^{(j)} &= \mathbf{W}^{(0)} - \alpha_{j+1,j} \mathbf{P}_j \left( \mathbf{W}^{(j-1)} \right)^{-1} \mathbf{R} \left( \mathbf{W}^{(j-1)} \right), \quad j = 1, \dots, s \tag{47} \\ \mathbf{W}^{n+1} &= \mathbf{W}^{(s)}, \end{aligned}$$

$$\mathbf{P}_j \left( \mathbf{W}^{(j-1)} \right) := (\Delta T)^{-1} \mathbf{M} + \frac{d\mathbf{R}}{d\mathbf{W}} \left[ \mathbf{W}^{(j-1)} \right], \quad \Delta T := \text{diag}(\text{diag}(\Delta t_i)), \tag{48}$$

where we have neglected the subindices mean and turb, and  $\Delta t_i$  denotes the local time step. To apply (47), the linear equation

$$\mathbf{Prec}_j^{-1} \mathbf{P}_j \mathbf{x} = \alpha_{j+1,j} \mathbf{Prec}_j^{-1} \mathbf{R}(\mathbf{W}^{(j-1)}). \tag{49}$$

needs to be solved. Here,  $\mathbf{Prec}_j$  denotes some preconditioner, which is the major ingredient of the algorithm. To find an approximation of (49), we apply a (left) preconditioned GMRES method (see for example [45]) with initial guess  $\mathbf{x}^{(0)} = 0$ :

- Solve (approximately)  $\mathbf{Prec}_j \mathbf{r}_0 = \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)})$
- Compute  $\beta := \|\mathbf{r}_0\|_2, \mathbf{v}_1 := \frac{1}{\beta} \mathbf{r}_0$
- for  $k = 1, \dots, m$ 
  - Solve (approximately)  $\mathbf{Prec}_j \mathbf{w} = \mathbf{P}_j \mathbf{v}_k$
  - for  $i = 1, \dots, k$ 
    - \*  $h_{i,k} := \langle \mathbf{w}, \mathbf{v}_i \rangle$
    - \*  $\mathbf{w} := \mathbf{w} - h_{i,k} \mathbf{v}_i$
  - $h_{k+1,k} = \|\mathbf{w}\|_2, \mathbf{v}_{k+1} = \frac{1}{h_{k+1,k}} \mathbf{w}$

- $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_m)$ ,  $H_m = (h_{i,k})_{1 \leq i \leq k+1, 1 \leq k \leq m}$
- Solve  $\mathbf{y}_m := \operatorname{argmin}_{\mathbf{y}} \|\beta e_1 - H_m \mathbf{y}\|_2$  by Given's-rotations
- $\mathbf{x}^{(m)} := \mathbf{x}^{(0)} + \mathbf{V}_m \mathbf{y}_m$

In the GMRES method, the matrix–vector multiplication of the operator given in (49) applied to a vector is approximated by a finite-difference of the residual operator,

$$\frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}) \mathbf{h} \approx \frac{1}{2\epsilon} (\mathbf{R}(\mathbf{W} + \epsilon \mathbf{h}) - \mathbf{R}(\mathbf{W} - \epsilon \mathbf{h})).$$

The choice of a suitable  $\epsilon > 0$  is not trivial, as cancellation and approximation errors need to be balanced. A possible method can be found in [42, Chapter 6.2.5].

### 4.2 Construction of Preconditioner

The preconditioner has two contributions:

- (a) the linear operator  $\mathbf{Prec}_j$  itself,
- (b) an iterative solution method for approximately solving the linear systems

$$\mathbf{Prec}_j \mathbf{w} = \mathbf{P}_j \mathbf{v}_k. \tag{50}$$

For the construction of the linear operators we follow the approach presented in [29, 40]:

$$\mathbf{Prec}_{j,\text{mean}} := (\Delta T)^{-1} \mathbf{M}_{\text{mean}} + \epsilon \alpha_{jj} \frac{d\tilde{\mathbf{R}}_{\text{mean}}^{\text{comp}}}{d\mathbf{W}}, \tag{51}$$

$$\mathbf{Prec}_{j,\text{turb}} := (\Delta T)^{-1} \mathbf{M}_{\text{turb}} + \epsilon \alpha_{jj} \frac{d\tilde{\mathbf{R}}_{\text{turb}}^{\text{comp},(k,\omega)}}{d\mathbf{W}_t}. \tag{52}$$

Here,  $d\tilde{\mathbf{R}}_{\text{prec}}^{\text{comp}}/d\mathbf{W}$  is a linearization of a residual based on a compact discretization scheme.

For details of  $\mathbf{Prec}_{j,\text{mean}}$  we refer to [29, 40]. To obtain  $\frac{d\tilde{\mathbf{R}}_{\text{turb}}^{\text{comp},(k,\omega)}}{d\mathbf{W}_t}$  the upwind scheme is exactly differentiated. For the derivative of the viscous terms a thin shear layer assumption is assumed for the gradients. The source terms are also exactly differentiated, also the diffusion source terms which typically give entries to the off-diagonal entries of the matrix. But, to realize the preconditioner, the derivatives of the source terms to construct  $\mathbf{Prec}_{j,\text{turb}}$  do not include the destruction terms

$$\frac{\partial D e_{k,(k,\omega)}}{\partial k_i} \quad \text{and} \quad \frac{\partial D e_{\omega,(k,\omega)}}{\partial \omega_i}. \tag{53}$$

The necessity for this modification is discussed in Sect. 5.1. Additionally, the viscosity  $\nu_t$  is also exactly differentiated and the arising terms are completely included. For the Wilcox 1988 model the approach is described in detail in [40, Chapter 4].

### 4.3 Solving Linear Systems and Truncation Criteria

To realize the preconditioned GMRES method approximately solving (49), we need to define appropriate truncation criteria. To this end, we distinguish two principal implementations of (47):

- (M1) When the number of GMRES steps is 0 and only the preconditioning step is evaluated,
- (M2) When the number of GMRES steps is  $> 0$ .

To approximately solve (49) for the mean flow equations we follow both strategies M1 and M2:

- When choosing M1, we apply either at most 250 symmetric Gauss–Seidel sweeps or iterate until

$$\frac{\|\mathbf{Prec}_j \mathbf{r}_0 - \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)})\|_2}{\|\alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)})\|_2} < 10^{-2}.$$

- When choosing M2, we truncate the GMRES iteration either after at most 20 steps or if

$$\frac{\|\mathbf{P}_j \mathbf{x}^{(m)} - \alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)})\|_2}{\|\alpha_{j+1,j} \mathbf{R}(\mathbf{W}^{(j-1)})\|_2} < 10^{-2}.$$

For approximately solving the linear systems (50), we apply five symmetric Gauss–Seidel sweeps.

#### 4.4 Choice of Time Step

To compute the local time steps  $\Delta t_i$  in (48) for both the mean flow equations and for the turbulence modeling equations, we use an approximation to the spectral radius of the diagonal blocks of  $\frac{d\mathbf{R}}{d\mathbf{W}}$ , that is,

$$\Delta t_i := \text{CFL} \cdot \text{vol}(D_i) \left[ \sum_{j \in \mathcal{N}(i)} \text{svol}(e_{ij}) \left( \rho \left( \frac{\partial \mathcal{H}^{1st, \text{Roe}}}{\partial W_i} \right) + C_v \rho \left( \frac{\partial \langle f_v(W_i, W_j), n_{e_{ij}} \rangle^{\text{TSL}, \mu=\text{const}}}{\partial W_i} \right) \right) \right]^{-1}, \quad C_v := 8.$$

The CFL number is chosen according to

$$\text{CFL} = \min \{ \text{CFL}_{\text{init}} \cdot f(n), \text{CFL}_{\text{max}} \}, \tag{54a}$$

$$f(n) = \begin{cases} 1, & n < 10, \\ \alpha^{n-10}, & n \geq 10, \end{cases} \quad \alpha > 1. \tag{54b}$$

Typical values are  $\text{CFL}_{\text{init}} = 3$  and  $\text{CFL}_{\text{max}} = 1000$ . For more details we refer to [29, 40, 46].

#### 4.5 Positivity of $k$ and $\omega$

To deal with the problem of positivity of  $k$  and  $\omega$  we simply introduced a damping of the updates. For example, Algorithm (47) gives for the variables  $k_i$  and  $\omega_i$ ,  $i = 1, \dots, N_{\text{elem}}$ , the updates

$$k_i^{(j)} = k_i^{(0)} - \Delta k_i, \tag{55a}$$

$$\omega_i^{(j)} = \omega_i^{(0)} - \Delta \omega_i, \tag{55b}$$

where  $(\Delta k_i, \Delta \omega_i)$  denotes the symbol for  $i$ th entry of the solution vector that one obtains by evaluating

$$\alpha_{j+1,j} \mathbf{P}_j^{-1} \mathbf{R}(\mathbf{W}^{(j-1)}).$$

Direct application of (55) often yields negative values, in particular for  $k$ . Most often this is observed for high-lift test cases, but for almost all test cases negative values show up for  $k$  and  $\omega$  at least during the starting phase of the iteration. Therefore, we replaced the update (55) by an application of Algorithm 1.

---

**Algorithm 1** Update for  $k$ - $\omega$  model
 

---

```

1: procedure LOOP OVER ALL MESH POINTS TO UPDATE  $k$  AND  $\omega$ 
2:   for  $i = 1, \dots, N_{elem}$  do
3:      $s_n = 1$ 
4:     for  $n = 1, 2, \dots$  do
5:        $k_i^{new} = k_i^{(0)} - s_n \Delta k_i$ 
6:       if  $k_i^{new} > 0$  then
7:          $k_i^{(j)} = k_i^{new}$ 
8:         break
9:       else
10:         $s_{n+1} = \frac{s_n}{2}$ 
11:      end if
12:    end for
13:  end for
14:  for  $i = 1, \dots, N_{elem}$  do
15:     $s_n = 1$ 
16:    for  $n = 1, 2, \dots$  do
17:       $\omega_i^{new} = \omega_i^{(0)} - s_n \Delta \omega_i$ 
18:      if  $\omega_i^{new} > 0$  then
19:         $\omega_i^{(j)} = \omega_i^{new}$ 
20:        break
21:      else
22:         $s_{n+1} = \frac{s_n}{2}$ 
23:      end if
24:    end for
25:  end for
26: end procedure

```

---

Algorithm 1 represents a kind of damped Newton method introducing a further effect of regularization. Expressed in formulae, Algorithm 1 realizes the following condition:

$$s_{n,i}^{(k)} = \min_{n \in \mathbb{N}_0} \left\{ \frac{1}{2^n} \right\} \quad \text{such that } k_i^{(j)} > 0,$$

$$s_{n,i}^{(\omega)} = \min_{n \in \mathbb{N}_0} \left\{ \frac{1}{2^n} \right\} \quad \text{such that } \omega_i^{(j)} > 0.$$

A side effect of this approach is the fact that the updates may become arbitrarily small, yielding an overall convergence corruption. However, so far, for none of the considered test cases has a stall of convergence been observed. Compared with many others methods tried to ensure positivity of  $k$  and  $\omega$ , Algorithm 1 was found to be superior with the present implementation. The simplicity of Algorithm 1 is another argument for its use. Nevertheless, Algorithm 1 cannot guarantee convergence. Hence, future work may require focus on other mechanisms to ensure positivity of  $k$  and  $\omega$  without reformulating the  $k$ - $\omega$  model itself. On the other hand, the damping of updates

$$k_i^{new} = k_i^{(0)} - s_n \Delta k_i$$

$$\omega_i^{new} = \omega_i^{(0)} - s_n \Delta \omega_i$$



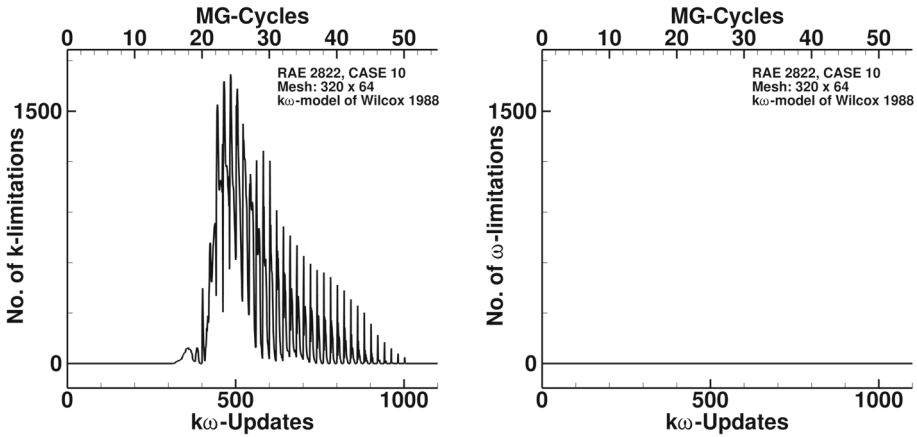


Fig. 1 Number of limitations for the 1988 Wilcox  $k$ - $\omega$  model

is not a severe restriction and is justified in the following sense. Using in general (47) to compute the updates, we have

$$(\Delta k_i, \Delta \omega_i) = \alpha_{j+1,j} \left[ ((\Delta t)^{-1} \mathbf{M} + \alpha_{jj} \frac{d\mathbf{R}}{d\mathbf{W}} (\mathbf{W}^{(j-1)}))^{-1, \text{app}} \mathbf{R}(\mathbf{W}^{(j-1)}) \right]_i.$$

A necessary criterion for convergence is  $\|\mathbf{R}(\mathbf{W}^{(j-1)})\| \rightarrow 0$ , that is in particular

$$\left( \mathbf{R}(\mathbf{W}^{(j-1)}) \right)_i \rightarrow 0.$$

If the solution of the turbulence modeling equations converges, then at some iterate the updates  $\Delta k_i$  and  $\Delta \omega_i$  are so small that additional damping is not necessary, and at the same time, positivity of  $k$  and  $\omega$  is ensured. In this sense, Algorithm 1 is not a severe restriction.

To illustrate the mode of operation of Algorithm 1, a plot of the number of  $k$  limitations and  $\omega$  limitations is given in Figs. 1 and 2 for the Wilcox model of 1988 and for the SST model, respectively. To approximate a solution, we performed for each multigrid cycle on the mean flow equations 20 subiterations on the turbulence modeling equations. The number of multigrid cycles is plotted on the upper  $x$ -axis, the total number of subiterations on the lower  $x$ -axis. From Fig. 1 (left) we can observe that not only the total number of limitations for  $k$  goes to zero, but also within each subiteration the number of limitations is significantly reduced. For the Wilcox model of 1988, no limitation of the  $\omega$  variable is required, which is clearly seen in Fig. 1 (right).

For the considered example, the number of limitations required for the SST-model is significantly smaller. Moreover, Fig. 2 (right) shows that for the SST-model also limitations for  $\omega$  are performed. As expected from considerations above, at some level in convergence the number of limitations for both the  $k$  and  $\omega$  variables is 0. This means, that discrete, positive solutions for  $k$  and  $\omega$  are obtained. This confirms numerically that the suggested limitation given by Algorithm 1 yields discrete solutions satisfying positivity for  $k$  and  $\omega$  if convergence of the equations is observed. However, as long as one of the functions for  $k$  or  $\omega$  is locally negative, the algorithm cannot converge.

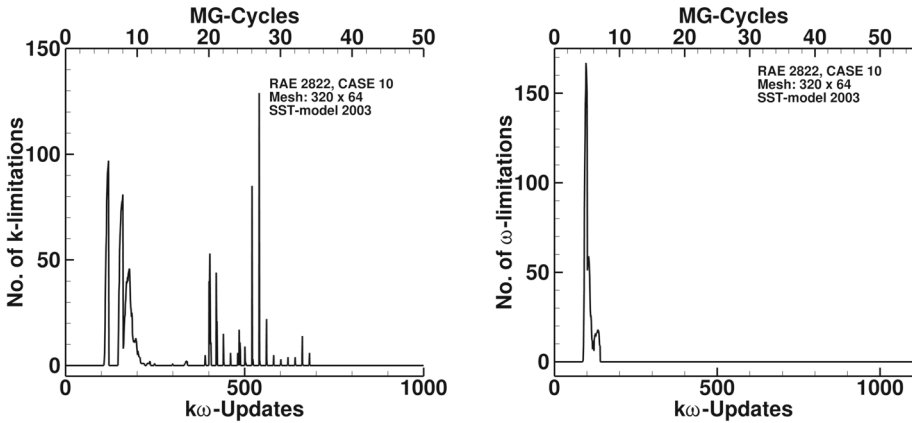


Fig. 2 Number of limitations for the Menter SST model

### 5 Considerations for the $k-\omega$ Model

In this section, at least for some of the heuristics and arguments previously given, we are going to present both an analysis for the linear and nonlinear parts of the solution algorithm.

#### 5.1 Analysis of Linear Part

In Sect. 4.1 it is stated that the derivatives of the destruction terms (53) are neglected for the construction of the preconditioner. To illustrate this necessity, we use that the splitting method

$$x_{m+1} = Mx_m + Nb$$

converges if and only if  $\rho(M) < 1$  (see e.g. [47]). For the Jacobi and Gauss–Seidel method the iteration matrix  $M$  is given by

$$M_{Jac} = D^{-1} (D - A) \quad \text{and} \quad M_{GS} = -(D + L)^{-1} R.$$

Using Arnoldi’s method, as in the inner part of the GMRES method [29, 31], we can approximate the spectrum of these operators for the preconditioner (52), including and neglecting (53). Figures 3 and 4 show the approximated eigenvalue distributions at the beginning and end of a nonlinear iteration, respectively. For each Figure, (53) is included on the left and excluded on the right. In particular, at the beginning of the iteration an inclusion of (53) changes the spectrum of the iteration matrices such that convergence of both the Gauss–Seidel and the Jacobi methods is not possible, whereas exclusion of (53) gives a converging method. For the final state the situation improves, but as one can observe in Fig. 4 (left), there are still a few approximate eigenvalues outside the range of stability. Hence, for the overall nonlinear iteration process it is necessary and beneficial to neglect (53). Moreover, when comparing the spectra of the Jacobi and the Gauss–Seidel methods, we observe a more equally distributed one for the latter one.

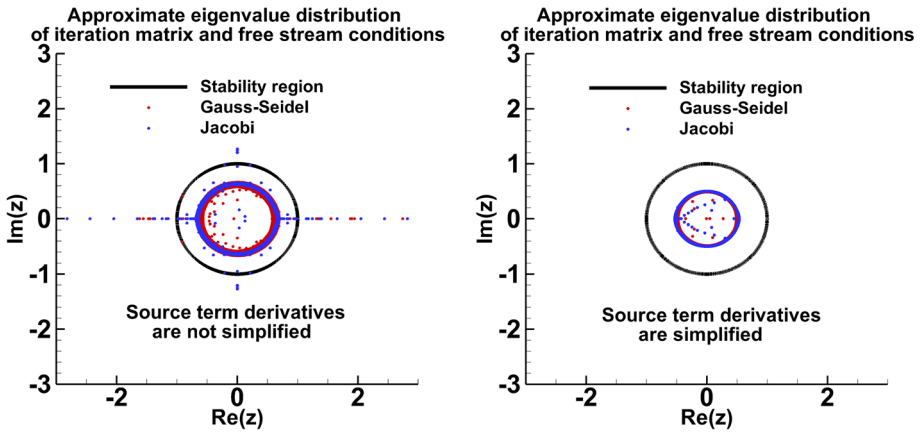


Fig. 3 Eigenvalue distribution of Jacobi and Gauss–Seidel method in the beginning of nonlinear iteration

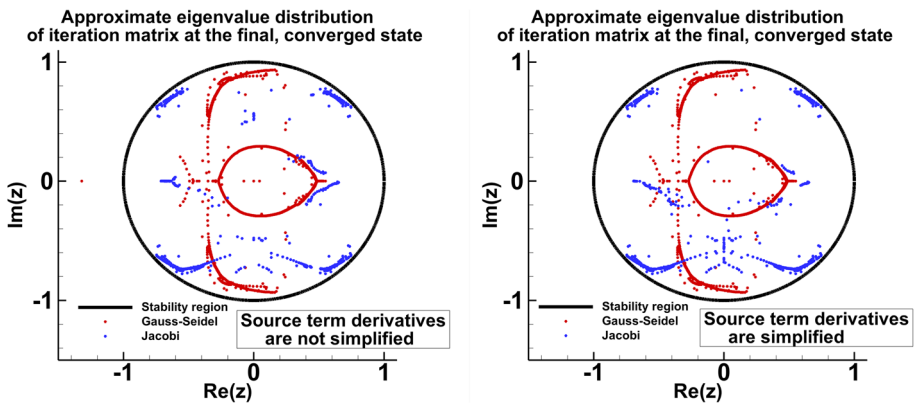


Fig. 4 Eigenvalue distribution of Jacobi and Gauss–Seidel method for the final nonlinear iteration

### 5.2 Analysis of Nonlinear Part

To understand if we can expect convergence of the solution for the turbulence modeling equations for the outer nonlinear iteration, at least in a small neighborhood of a solution  $(\mathbf{W}^\dagger, \mathbf{W}_t^\dagger)$ , for example,

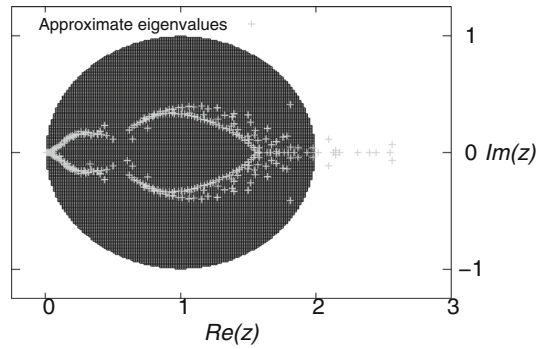
$$\mathbf{R}_{\text{turb}}(\mathbf{W}_t^\dagger, \mathbf{W}^\dagger) = 0.$$

We assume a small perturbation, that is,  $\|\mathbf{W}_t\| < \epsilon$  of  $\mathbf{W}_t^\dagger$ . Using a Taylor series expansion and neglecting terms of higher order, we approximate

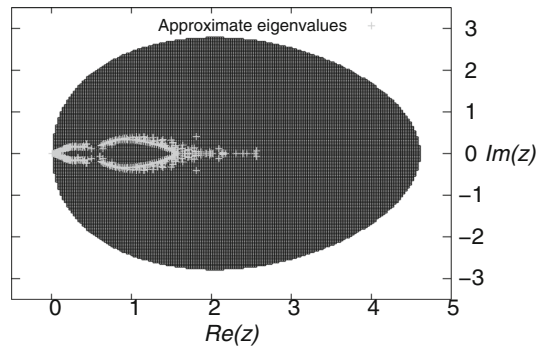
$$\begin{aligned} \frac{d\mathbf{W}(t)}{dt} &= \frac{d(\mathbf{W}^\dagger + \mathbf{W}(t))}{dt} = -\mathbf{M}^{-1}\mathbf{R}(\mathbf{W}^\dagger + \mathbf{W}) \\ &\approx -\mathbf{M}^{-1}\left(\mathbf{R}(\mathbf{W}^\dagger) + \frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^\dagger)\mathbf{W}\right) \end{aligned}$$

**Fig. 5** Range of stability for multistage schemes and approximate spectrum for two equation  $k$ - $\omega$  model of CASE 9

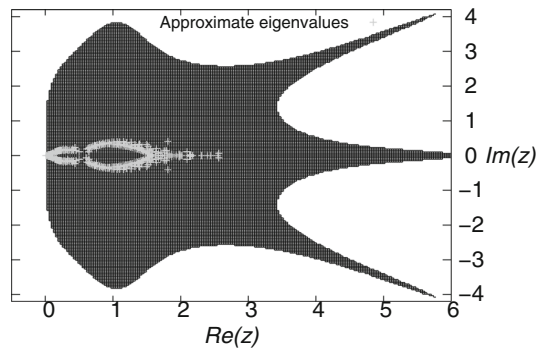
Linear stability range for one-stage scheme



Linear stability range for three-stage scheme



Linear stability range for five-stage scheme



$$= -\mathbf{M}^{-1}\mathbf{A}\mathbf{W}(t), \quad \mathbf{A} := \frac{d\mathbf{R}}{d\mathbf{W}}(\mathbf{W}^\dagger).$$

Then, the update of the multistage Runge–Kutta scheme (47) can be expressed by a polynomial expression,

$$\mathbf{W}^{n+1} = q_s(\mathbf{Prec}^{-1, \text{app}} \mathbf{A}) \mathbf{W}^n,$$

$$q_s(z) = 1 + \sum_{j=1}^s (-1)^j z^j \prod_{i=s-j+1}^s \alpha_{i+1,i}. \tag{56}$$

Thus, the multistage Runge–Kutta scheme (47) is stable, that is, it defines a contracting operator in a linear sense for all  $z \in \mathbb{C}$ , if  $|q_s(z)| < 1$ .

To compute approximations to the eigenvalues of  $\mathbf{P}^{-1,app} \mathbf{A}$ , we again exploit the GMRES method and its connection to the Arnoldi process (see [29, 31]). As a test case, we consider Case 9 from the experiments of Cook, McDonald and Firmin [48] and a  $320 \times 64$  C-type mesh. This mesh is a member of a family of meshes used later in Sect. 6 on Numerical Examples. To transform the spectrum we restrict ourselves to the one-stage scheme with coefficients  $\alpha_{21} = 1.0$ , the three-stage scheme with coefficients  $\alpha_{21} = 0.15$ ,  $\alpha_{32} = 0.4$ ,  $\alpha_{43} = 1$  and the five-stage scheme with coefficients  $\alpha_{21} = 0.0695$ ,  $\alpha_{32} = 0.1602$ ,  $\alpha_{43} = 0.2898$ ,  $\alpha_{54} = 0.5060$  and  $\alpha_{65} = 1.0$  (see [25]). Figure 5 shows the range of stability of these multistage schemes together with the approximate spectrum. For this test case and for both the three-stage and the five-stage schemes, we can expect a stable method, whereas the one stage scheme is expected to diverge.

### 6 Numerical Examples

To demonstrate the applicability of the suggested algorithms, we consider some standard 2-D and challenging 3-D test cases, which have been computed frequently in the literature. The 2-D cases are transonic flows over an airfoil. The 3-D cases include relatively low-speed flow over a wing-body geometry at a high angle of attack and transonic flow over a wing-body configuration. For the 2-D cases, we consider both the Wilcox and SST  $k-\omega$  models. The SST model is used in the 3-D computations. The density residual is evaluated by

$$\text{Density residual}(n) := \sqrt{\sum_{j=1}^{N_{elem}} \frac{(\mathbf{R}_{j,\text{mean},\rho}(\mathbf{W}^{T_n}))^2}{(\text{vol}(\Omega_j))^2}} / \sqrt{\sum_{j=1}^{N_{elem}} \frac{(\mathbf{R}_{j,\text{mean},\rho}(\mathbf{W}_\infty))^2}{(\text{vol}(\Omega_j))^2}},$$

for  $k-\omega$  model we evaluate both corresponding residuals

$$k - \text{residual}(n) := \sqrt{\sum_{j=1}^{N_{elem}} \frac{(\mathbf{R}_{j,\text{turb},k}(k^{T_n}, \omega^{T_n}))^2}{(\text{vol}(\Omega_j))^2}} / \sqrt{\sum_{j=1}^{N_{elem}} \frac{(\mathbf{R}_{j,\text{turb},k}(k_\infty, \omega_\infty))^2}{(\text{vol}(\Omega_j))^2}},$$

$$\omega - \text{residual}(n) := \sqrt{\sum_{j=1}^{N_{elem}} \frac{(\mathbf{R}_{j,\text{turb},\omega}(k^{T_n}, \omega^{T_n}))^2}{(\text{vol}(\Omega_j))^2}} / \sqrt{\sum_{j=1}^{N_{elem}} \frac{(\mathbf{R}_{j,\text{turb},\omega}(k_\infty, \omega_\infty))^2}{(\text{vol}(\Omega_j))^2}}.$$

To satisfy near machine zero, a computation is truncated, and the result is accepted as soon as  $\text{density residual}(n) < 10^{-14}$ . The computations on unstructured grids were performed in parallel using MPI and either the C<sup>2</sup>A<sup>2</sup>S<sup>2</sup>E<sup>2</sup> or CARA cluster of DLR.

#### 6.1 RAE 2822 Airfoil

The first examples considered correspond to the RAE 2822 airfoil. These examples have been chosen because they are frequently considered when attempting to validate turbulence

**Table 1** Flow Conditions for RAE 2822 airfoil

Cases	$M_\infty$	AoA	$Re$
Case 9	0.73	2.79°	$6.5 \cdot 10^6$
Case 10	0.75	2.81°	$6.2 \cdot 10^6$

**Table 2** Mesh data for RAE 2822 airfoil

	Coarse	Medium	Fine
Mesh size	$320 \times 64$	$640 \times 128$	$1280 \times 256$
No. of quadrilaterals	20,480	81,920	327,680
No. of cells on the airfoil	256	512	1024

**Table 3** Case 9: Computed lift and drag coefficients with 1988  $k-\omega$  model

Model	Grid	$C_L$	$C_D$	$(C_D)_p$	$(C_D)_v$
$k\omega$ -1988	$320 \times 64$	0.792254	0.0195732	0.0120721	0.0075010
$k\omega$ -1988	$640 \times 128$	0.804623	0.0191719	0.0122047	0.0069673
$k\omega$ -1988	$1280 \times 256$	0.806221	0.0189818	0.0121783	0.0068035

**Table 4** Case 9: Computed lift and drag coefficients with SST model

Model	Grid	$C_L$	$C_D$	$(C_D)_p$	$(C_D)_v$
SST	$320 \times 64$	0.742041	0.0163938	0.0103164	0.0060774
SST	$640 \times 128$	0.761013	0.0164595	0.0106297	0.0058298
SST	$1280 \times 256$	0.766296	0.0164743	0.0107338	0.0057406

models. There are two cases, Case 9 and Case 10, for transonic flow over the RAE 2822 airfoil [48]. The flow conditions for these cases are given in Table 1. In Case 9 there is a fairly strong shock wave occurring on the upper surface of the airfoil, whereas in Case 10 there is a sufficiently strong shock on the upper surface to cause significant separation of the flow behind the shock.

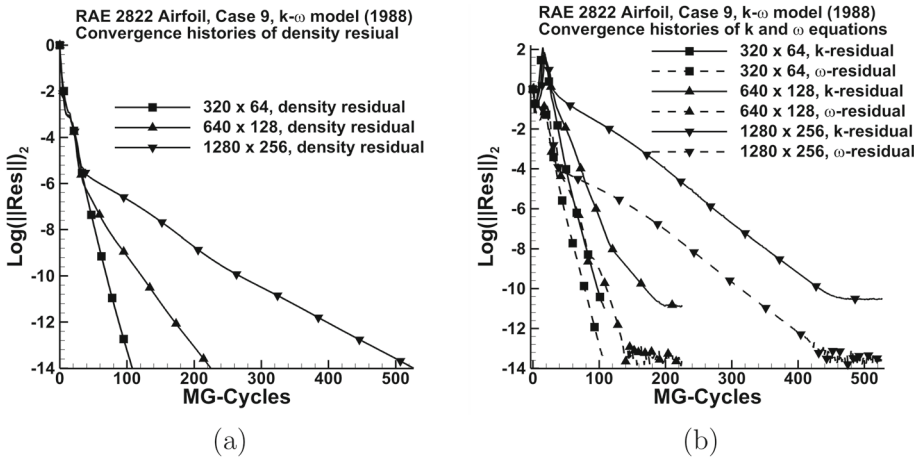
We perform the computations on a sequence of C-type structured meshes described in Table 2. The meshes have a C-type topology. The finest mesh consists of 1280 cells around the airfoil (1024 cells on the airfoil) and 256 cells in normal direction. The normal mesh spacing at the surface of the finest mesh is approximately  $3 \times 10^{-6}$ , and the maximum surface cell aspect ratio is about 560.

**Table 5** Case 10: Computed lift and drag coefficients with 1988  $k-\omega$  model

Model	Grid	$C_L$	$C_D$	$(C_D)_p$	$(C_D)_v$
$k\omega$ -1988	$320 \times 64$	0.800997	0.0309963	0.0236907	0.0073056
$k\omega$ -1988	$640 \times 128$	0.814742	0.0300047	0.0232394	0.0068074
$k\omega$ -1988	$1280 \times 256$	0.815570	0.0298718	0.0232158	0.0066560

**Table 6** Case 10: Computed lift and drag coefficients with SST model

Model	Grid	$C_L$	$C_D$	$(C_D)_p$	$(C_D)_v$
SST	$320 \times 64$	0.722263	0.0241748	0.0187140	0.0054608
SST	$640 \times 128$	0.742187	0.0246749	0.0194276	0.0052473
SST	$1280 \times 256$	0.743644	0.0248062	0.0194821	0.0053241



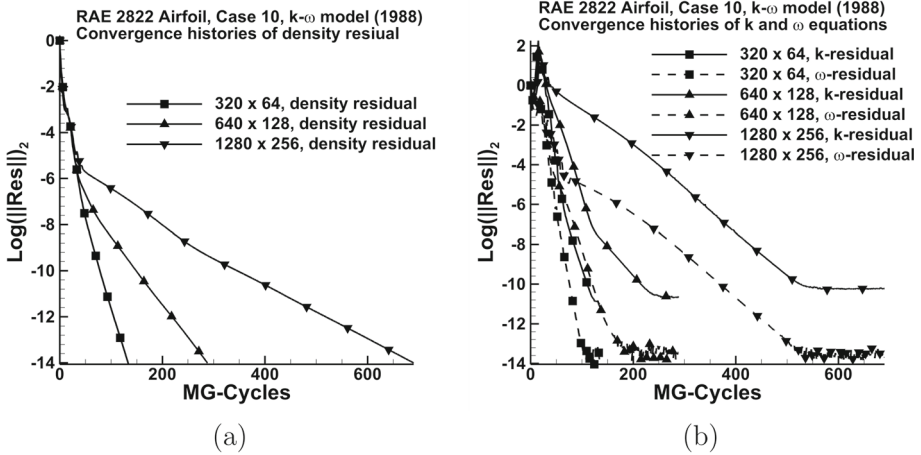
**Fig. 6** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 9). The effects of turbulence are represented with the Wilcox  $k-\omega$  model (1988). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** Mean flow equations, **b**  $k-\omega$  equations. (unstructured grid code)

The initial numerical calculations for the two RAE 2822 cases were performed with the complete solution algorithm of this paper, which includes agglomerative multigrid and GMRES, in the framework of an unstructured grid computer code. Tables 3, 4, 5, 6 include the predicted lift and total drag coefficients, including the pressure and skin-friction contributions. In Figs. 6, 7, 8 and 9, convergence histories using the 1988 version of the Wilcox  $k-\omega$  and the 2003 Menter SST turbulence models are presented. The final plots, which are included in Fig. 10, show the computed surface pressure distributions for the two RAE cases compared with the experimental data of Cook, McDonald and Firman [48].

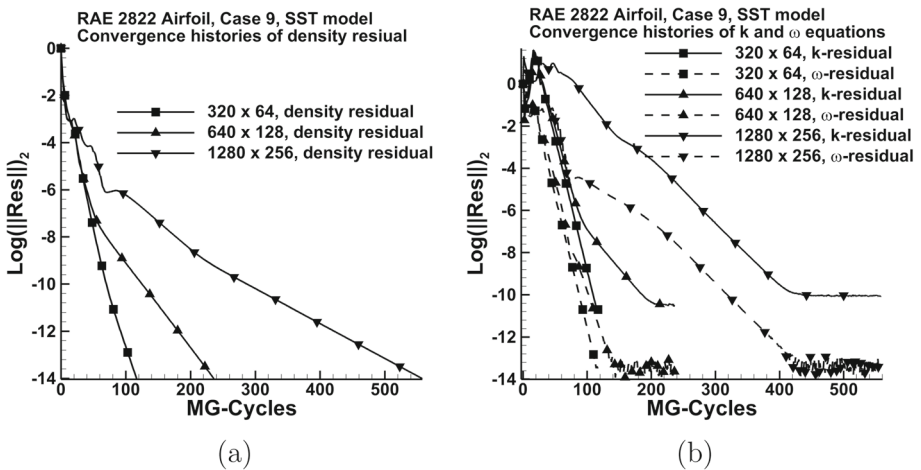
In the second set of results for Cases 9 and 10, the numerical computations were performed using the *core* of the solution algorithm described previously in the framework of a structured grid computer code. Results are for both the 2006 version of the Wilcox  $k-\omega$  and the 2003 Menter SST turbulence models. The convergence history results from these calculations are displayed in Figs. 11, 12, 13, 14. As shown in these figures, the residuals for both the RANS equations and the transport equations of the turbulence models indicate a consistent, reliable linear convergence rate and solutions are attained in only 125 to 150 multigrid cycles. Additional discussion of applications of the structured grid code are given in [13].

## 6.2 Transonic Turbulent Flow over a Common Research Model

The next test case is the first of the two 3-D flow computations in this paper. This first case is turbulent flow over the Common Research Model (CRM). It was considered at the fifth AIAA



**Fig. 7** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 10). The effects of turbulence are represented with the Wilcox  $k-\omega$  model (1988). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** Mean flow equations, **b**  $k-\omega$  equations. (unstructured grid code)

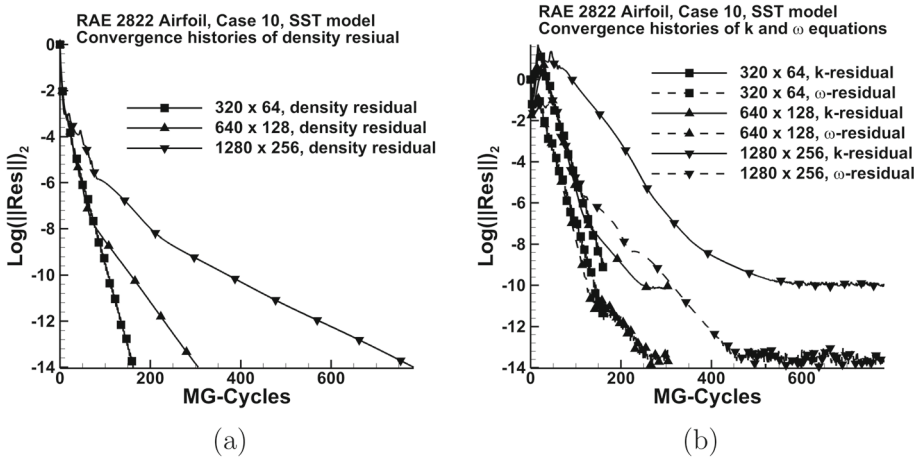


**Fig. 8** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 9). The effects of turbulence are represented with the Menter SST model (2003). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** mean flow equations, **b**  $k-\omega$  equations. (unstructured grid code)

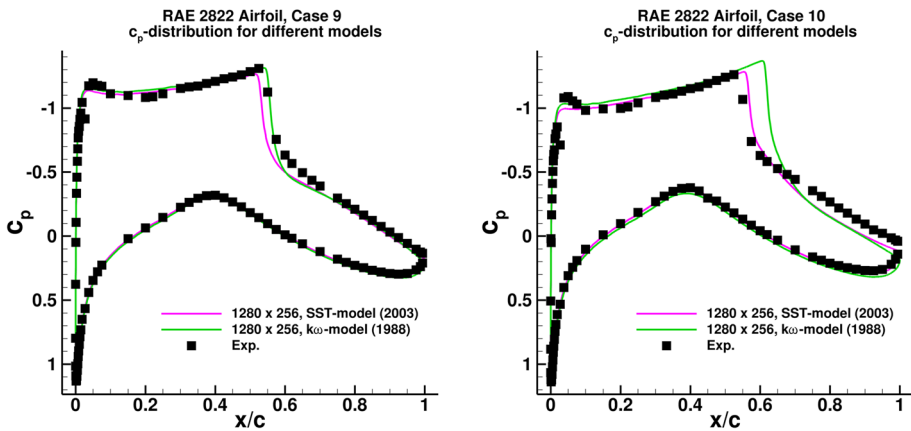
Drag Prediction Workshop. The meshes provided by the workshop are block-structured. A sequence of hybrid meshes was generated from the pure hexahedral meshes. Table 7 provides information about the grids used in the computations. The relevant physical conditions for this case are also given. The computed lift and drag coefficients as well as computational times for both mesh sequences are given in Table 8. There is a detailed description of the meshes given in [49]. The computations were performed on DLR's  $C^2A^2S^2E$  cluster. For the L4, L3, L2, and L1 meshes, we used 192, 72, 48 and 24 domains, resp. This corresponds to about 92000, 71000, 46000 and 28000 mesh points per domain.

- Geometry: Wing-body configuration, fifth AIAA Drag Prediction Workshop
- Reynolds number:  $Re = 5.0 \cdot 10^6$





**Fig. 9** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 10). The effects of turbulence are represented with the Menter SST model (2003). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** Mean flow equations, **b**  $k$ - $\omega$  equations (unstructured grid code)

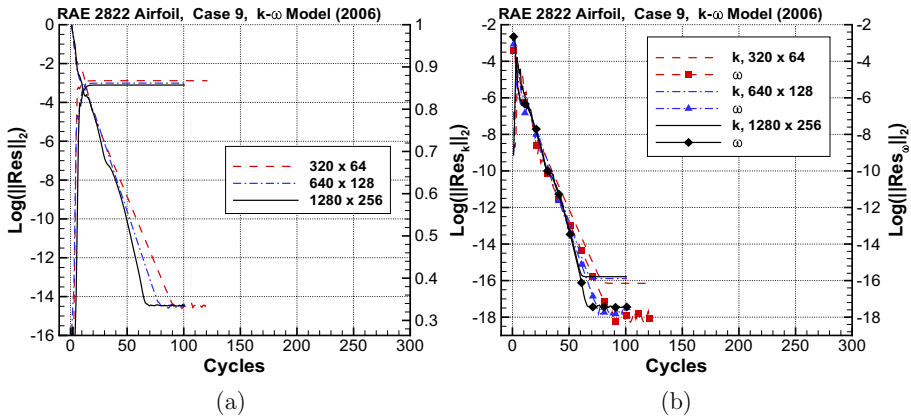


**Fig. 10**  $C_p$ -distribution for Case 9 and Case 10 and comparison with experimental data

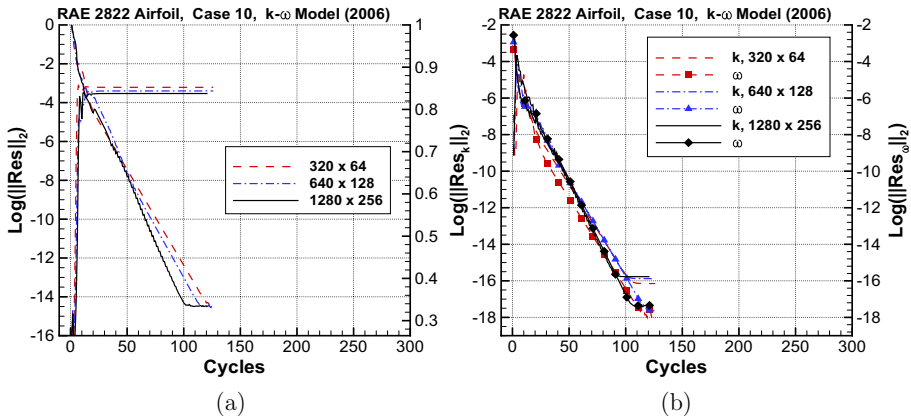
- Inflow Mach number:  $M_\infty = 0.85$
- Angle of attack  $2.15^\circ$

It should be pointed out that it is difficult to find reports in the literature in which systematic mesh refinement studies together with two-equation turbulence models (e.g., the SST-model) are shown for 3-D turbulent flows and the solution algorithm shows a consistent and reliable convergence behavior. In this paper the objective is not to perform extensive mesh refinement studies and attempt to evaluate accuracy, but rather to demonstrate that the algorithm presented is effective in reducing residuals well below the truncation error of the numerical algorithm.

Figure 15 shows the convergence histories for the sequence of meshes for the considered test case. Application of the algorithm presented in Sect. 4 made it possible to reach similar consistent convergence behavior for both the hexahedral and hybrid sequence of meshes and the stopping criterion was met in only a few hundred multigrid cycles. In Fig. 16, the surface



**Fig. 11** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 9). The effects of turbulence are represented with the Wilcox  $k-\omega$  model (2006). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** Mean flow equations, **b**  $k-\omega$  equations. (structured grid code)

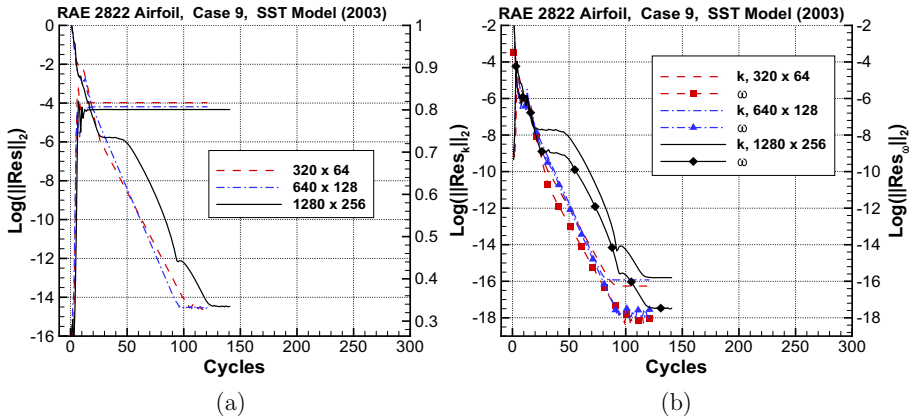


**Fig. 12** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 10). The effects of turbulence are represented with the Wilcox  $k-\omega$  model (2006). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** Mean flow equations, **b**  $k-\omega$  equations. (structured grid code)

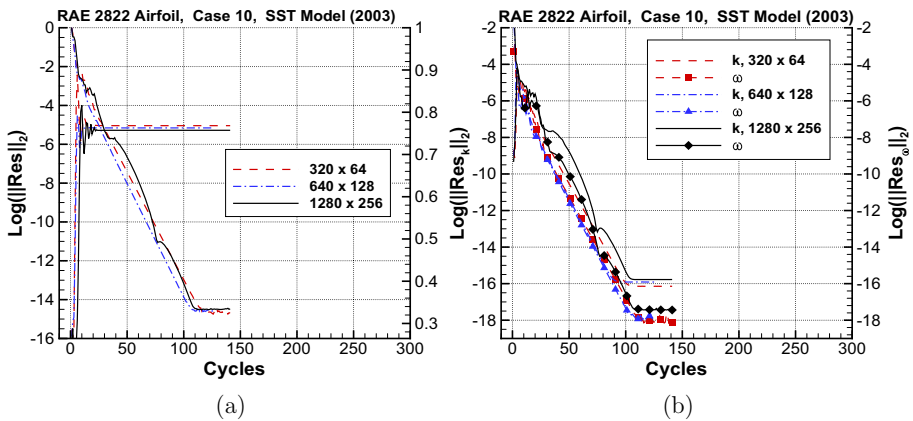
$C_p$  distribution for the Common Research Model using hybrid and hexahedral meshes is displayed.

### 6.3 NASA TRAP Wing

As a final example, to investigate the behavior of the proposed solution algorithm towards the incompressible limit, we examine the NASA Trap Wing considered at the first AIAA High-Lift prediction workshop [5]. For this test case, the importance of an adequate low-speed preconditioner for the governing flow equations must be emphasized, as there are low-speed regions as well as regions in which the flow is significantly accelerated. The low-speed preconditioner applied in this algorithm has a numerical dissipation that is a function of the local Mach number (see [50] for details).



**Fig. 13** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 9). The effects of turbulence are represented with the Menter SST model (2003). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** mean flow equations, **b**  $k$ - $\omega$  equations. (structured grid code)



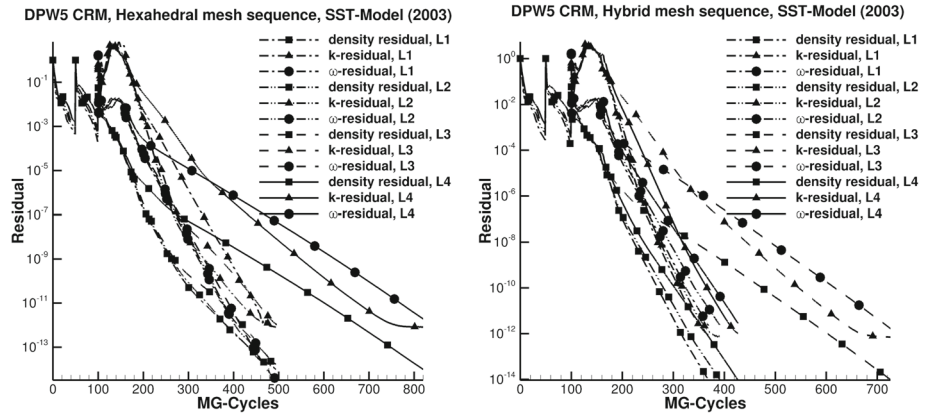
**Fig. 14** Convergence histories of solution computed for turbulent flow over RAE 2822 airfoil (Case 10). The effects of turbulence are represented with the Menter SST model (2003). A family of grids is considered:  $320 \times 64$ ,  $640 \times 128$ , and  $1280 \times 256$  cells. **a** Mean flow equations, **b**  $k$ - $\omega$  equations. (structured grid code)

**Table 7** Mesh data for DPW5 CRM

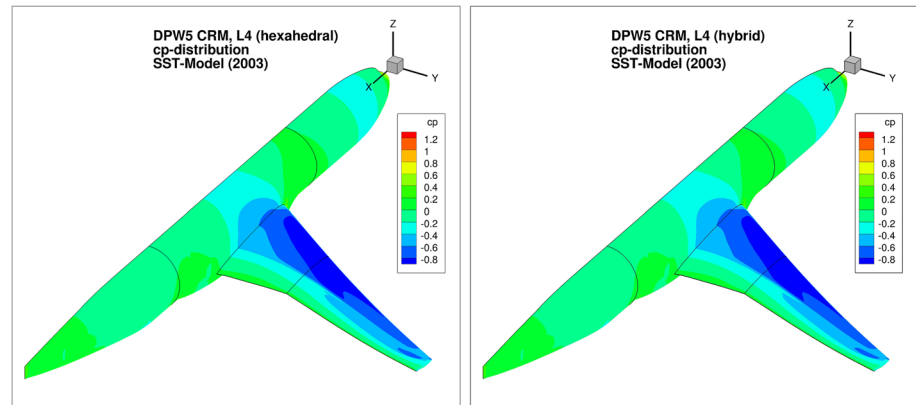
Level	Hybrid meshes		Hexahedral meshes	
	No. of Tetrahedra	No. of Prisms	No. of Hexahedrons	No. of points
L1	2,555,904	425,984	638,976	660,177
L2	8,626,176	1,437,696	2,156,544	2,204,089
L3	20,766,720	3,301,376	5,111,808	5,196,193
L4	69,728,256	11,261,952	17,252,352	17,441,905

**Table 8** lift and drag coefficient for CRM and SST model and computational times

	Hexahedral meshes			Hybrid meshes		
	$C_L$	$C_D$	Comp. time	$C_L$	$C_D$	Comp. time
L1	0.48636	0.02783	≈ 5.5 h (24 domains)	0.50094	0.03012	≈ 6 h (24 domains)
L2	0.46915	0.02505	≈ 10h (48 domains)	0.47431	0.02589	≈ 9.5h (48 domains)
L3	0.46985	0.02470	≈ 18h (72 domains)	0.47168	0.02494	≈ 18 h (72 domains)
L4	0.47426	0.02467	≈ 42h (384 domains)	0.47633	0.02488	≈ 41 h (384 domains)



**Fig. 15** Convergence histories for DPW5 CRM and SST-model



**Fig. 16** Surface  $C_p$  distribution for DPW5 CRM and SST-model on L4 hybrid and hexahedral mesh

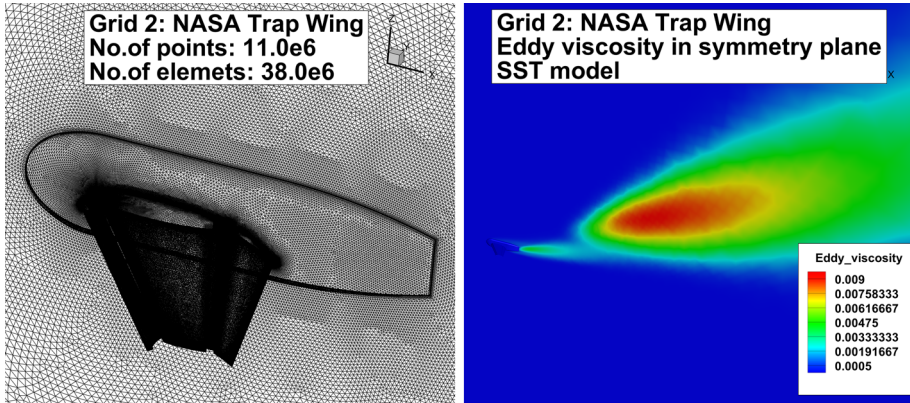
For the numerical computations, we used two meshes. The meshes were generated using VGrid and are marked as UH6 in Table 2 of [5]. Characteristics of these meshes are given in Table 9. Figure 17 provides a sense of the gridpoint distribution on the wing-body surface and symmetry plane for Grid 2. In addition, this figure shows the eddy viscosity contours corresponding to this grid on the symmetry plane.

The flow conditions for this 3-D example are given by

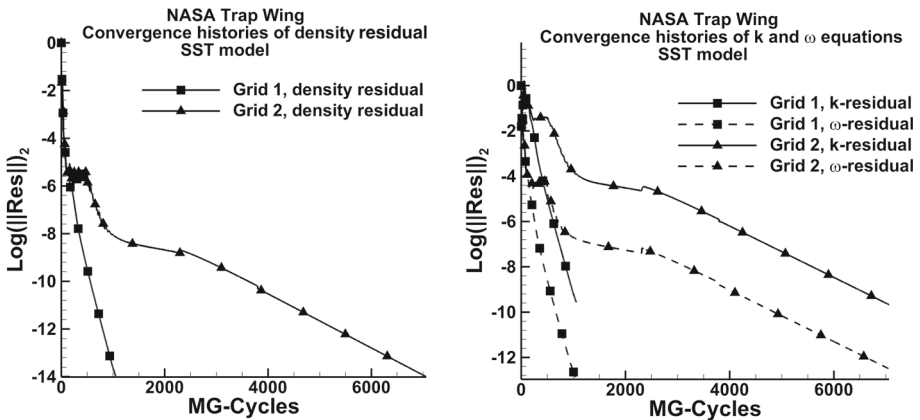
- Geometry: NASA TRAP Wing

**Table 9** NASA TRAP Wing: Mesh data

Mesh	No. of points	No. of elements
Grid 1	3,727,008	10,169,092
Grid 2	11,047,965	38,017,477



**Fig. 17** Left: Grid of NASA Trap Wing; Right: Computed eddy viscosity in the symmetry plane with the SST model



**Fig. 18** Convergence histories of mean flow and turbulent flow equations for the NASA Trap Wing and the SST model

- Reynolds number:  $Re = 4.3 \cdot 10^6$
- Inflow Mach number:  $M_\infty = 0.2$
- Angle of attack  $28^\circ$

The high angle of attack ( $\alpha$ ) is an especially important flow condition, due to the numerical challenge that it represents in achieving a fully converged solution, as evident in the literature. In addition, high  $\alpha$  aerodynamics can introduce numerous complexities into the flow field, including recirculation, vortical flow regions and various strong viscous types of interactions. Convergence histories for these computations with the SST model are displayed in Fig. 18. As in the results for the previous examples, these histories also exhibit near machine zero convergence.

## 7 Concluding Remarks

In this paper an effective numerical algorithm for solving the RANS equations and the two transport equations of  $k$ - $\omega$  type turbulence models has been presented. The integral form of the RANS and  $k$ - $\omega$  equations along with the corresponding boundary conditions for a well defined boundary-value problem have been presented and described. Details of the  $k$ - $\omega$  type two-equation models, which include all elements exactly as implemented, have been given.

It is important to emphasize that the two-equation models considered in the paper have been implemented according to the original papers of Wilcox [32, 34] and Menter [37]. Consequently, there are no additional limiters on the dependent variables, which are often used in the application of the models and not discussed. Furthermore, there are no realizability conditions being imposed, as seen in some finite-element methods (e.g., [21, 22]), which use a logarithmic variable substitution. While one can certainly argue that changing the turbulence model for the purpose of what one can call *numerical compatibility* is appropriate, from the viewpoint of this paper, this should only be acceptable if the integrity of the model is not compromised.

To demonstrate the effectiveness and capability of the numerical solution algorithm, several examples of aerodynamic applications have been presented. Both 2-D and 3-D flows have been considered with varying degrees of difficulties due to the complexity of the turbulent flows being considered. In all cases, the convergence histories have been shown. The computed lift and drag coefficients have also been given.

An analysis of the numerical algorithm is addressed, which reflects some of the properties and stability of the solution algorithm. An approximate eigenspectrum of the preconditioner for the inner part of the GMRES method has been determined using Arnoldi's method. The impact of the destruction terms, which are part of the source terms of a two-equation turbulence model, using Jacobi and Gauss–Seidel methods has been presented. Neglecting these terms in the Jacobian of the implicit preconditioner is necessary, in general, and highly beneficial. The distribution of the eigenvalues of the amplification matrix has been obtained for two-stage, three-stage and five-stage Runge–Kutta schemes with the implicit preconditioner. The advantage of the present technique with respect to local mode analysis [51] is that the stability is determined for a specific problem being solved.

As a final remark, in Sects. 3.3 and 3.4 the incompressible version of the considered models was introduced and finally implemented. When comparing with results from other codes using the original compressible version of the models for subsonic and transonic flow cases, no major differences between the results were observed. However, at the outset there seem to be no obvious obstacles to apply the methods proposed in this article also directly to the compressible version of the models. Future work will address how the methods proposed here can be directly applied to these compressible versions.

**Funding** Open Access funding enabled and organized by Projekt DEAL. The authors have not disclosed any funding.

**Data Availability** The data of this study is available from the authors upon reasonable request.

## Declarations

**Conflict of interest** The authors have no conflicts of interest to declare that are relevant to the content of this article.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Levy, D.W., Zickuhr, T., Vassburg, J., Agrawal, S., Wahls, R.A., Pirzadeh, S., Hensch, M.J.: Summary of Data from the First AIAA CFD Drag Prediction Workshop. AIAA Paper 2002-0841 (2002)
2. Levy, D.W., Laffin, K.R., Tinoco, E.N., Vassberg, J.C., Mani, M., Rider, B., Rumsey, C.L., Wahls, R.A., Morrison, J.H., Brodersen, O.P., Crippa, S., Mavriplis, D.J., Murayama, M.: Summary of Data from the Fifth AIAA CFD Drag Prediction Workshop. In Proc. 51st AIAA Aerospace Sciences Meeting, Grapevine (Dallas/Ft. Worth Region), Texas, January 2013, Number 2013-0046 in Conference Proceeding Series. AIAA (2013)
3. Tinoco, E.N., Brodersen, O.P., Keye, S., Laffin, K.R., Feltrop, E., Vassberg, J.C., Mani, M., Rider, B., Wahls, R.A., Morrison, J.H., Hue, D., Roy, C.J., Mavriplis, D.J., Murayama, M.: Summary data from the 6th AIAA CFD drag prediction workshop: CRM cases. *J. Aircr.* **55**(4), 1352–1379 (2018)
4. Derlaga, J.M., Morrison, J.H.: Statistical analysis of solutions from the sixth AIAA computational fluid dynamics drag prediction workshop. *J. Aircr.* **55**(4), 1388–1400 (2019)
5. Rumsey, C.L., Slotnick, J.P., Long, M., Stuever, A.R., Wayman, T.R.: Summary of the first AIAA CFD high-lift prediction workshop. *J. Aircr.* **48**(6), 2068–2079 (2011)
6. Rumsey, C.L., Slotnick, J.P.: Overview and Summary of the Second AIAA High Lift Prediction Workshop. AIAA Paper 2014-0747 (2014)
7. Rumsey, C.L., Slotnick, J.P., Sclafani, A.J.: Overview and summary of the third AIAA high lift prediction workshop. *J. Aircr.* **56**(2), 621–644 (2019)
8. Spalart, P.R., Allmaras, S.R.: A one-equation turbulence model for aerodynamic flows. In: AIAA Computational Fluid Dynamics Conference, number 1992-439 in Conference Proceeding Series. AIAA (1992)
9. Allmaras, S.R., Johnson, F.T., Spalart, P.R.: Modifications and clarifications for the implementation of the Spalart–Allmaras turbulence model. In: International Conference on Computational Fluid Dynamics 7, Hawaii, number ICCFD7-1902 in Conference Proceeding Series (2012)
10. Shur, M., Spalart, P., Strelets, M., Travin, A.: Turbulence modeling in rotating and curved channels: assessing the spalart-shur correction. *AIAA J.* **38**(5), 784–792 (2000)
11. Spalart, P.R.: Strategies for turbulence modelling and simulations. *Int. J. Heat Fluid Flow* **21**(3), 252–263 (2000)
12. Menter, F.R.: Two-equation Eddy-viscosity turbulence models for engineering applications. *AIAA J.* **32**(8), 1598–1605 (1994)
13. Swanson, R.C.: Solving two-equation turbulence models: with a perspective on solving transport equations. NASA Technical Memorandum, NASA/TM–20210016636, NASA Langley Research Center (2021)
14. Krist, S.L., Biedron, R.T., Rumsey, C.L.: CFL3D User's Manual (version 5.0). Technical Memorandum, NASA/TM–1998-208444, NASA Langley Research Center, Hampton, VA (1998)
15. Gerhold, T., Evans, J., Galle, M.: Technical Documentation of the DLR  $\tau$ -Code. Technical report, Deutsches Zentrum für Luft- und Raumfahrt (1997). LIDO-Berichtsjahr=1999
16. Mor-Yossef, Y., Levy, Y.: Unconditionally positive implicit procedure for two-equation turbulence models: application to  $k$ - $\omega$  turbulence models. *J. Comput. Phys.* **220**(1), 88–108 (2006)
17. Mor-Yossef, Y., Levy, Y.: The unconditionally positive-convergent implicit time integration scheme for two-equation turbulence models: Revisited. *Comput. Fluids* **38**(10), 1984–1994 (2009)
18. Wasserman, M., Mor-Yossef, Y., Yavneh, I., Greenberg, J.B.: A robust implicit multigrid method for rans equations with two-equation turbulence models. *J. Comput. Phys.* **229**(16), 5820–5842 (2010)
19. Wasserman, M., Mor-Yossef, Y., Greenberg, J.B.: A positivity-preserving, implicit defect-correction multigrid method for turbulent combustion. *J. Comput. Phys.* **316**, 303–337 (2016)
20. Ilinca, F., Hétu, J.-F., Pelletier, D.: A unified finite element algorithm for two-equation models of turbulence. *Comput. Fluids* **27**(3), 291–310 (1998)

21. Bassi, F., Crivellini, A., Rebay, S., Savini, M.: Discontinuous Galerkin solution of the Reynolds-averaged Navier–Stokes and  $k$ - $\omega$  turbulence model equations. *Comput. Fluids* **34**(4–5), 507–540 (2005)
22. Hartmann, R., Held, J., Leicht, T.: Adjoint-based error estimation and adaptive mesh refinement for the RANS and  $k$ - $\omega$  turbulence model equations. *J. Comput. Phys.* **230**(11), 4268–4284 (2011)
23. Venkatakrishnan, V., Allmaras, S.R., Kamenetskii, D.S., Johnson, F.T.: Higher order schemes for the compressible Navier–Stokes Equations. AIAA Paper 2003-3987 (2003)
24. Langer, S., Swanson, R.C.: On boundary-value problems for RANS equations and two-equation turbulence models. *J. Sci. Comput.* **85**(20), 1 (2020)
25. Swanson, R.C., Turkel, E., Rossow, C.-C.: Convergence acceleration of Runge–Kutta schemes for solving the Navier–Stokes equations. *J. Comput. Phys.* **224**(1), 365–388 (2007)
26. Swanson, R.C., Rossow, C.-C.: An efficient solver for the RANS equation and a one-equation turbulence model. *Comput. Fluids* **42**(1), 13–25 (2011)
27. Nishikawa, H., Diskin, B., Thomas, J.L., Hammond, D.H.: Recent advances in agglomerated multigrid. In: 51st AIAA Aerospace Sciences Meeting, AIAA Paper 2013-863, Texas (2013)
28. Nishikawa, H., Diskin, B., Thomas, J.L., Hammond, D.H.: Recent advances in agglomerated multigrid. In: 51st AIAA Aerospace Sciences Meeting, AIAA Paper 2013-863, Texas (2013)
29. Langer, S.: Agglomeration multigrid methods with implicit Runge–Kutta smoothers applied to aerodynamic simulations on unstructured grids. *J. Comput. Phys.* **277**, 72–100 (2014)
30. Langer, S., Schwöppe, A., Kroll, N.: The DLR flow solver TAU—status and recent algorithmic developments. In: Proceedings 52nd aerospace sciences meeting, January 2014, number 2014-0080 in Conference Proceeding Series. AIAA (2014)
31. Langer, S., Schwöppe, A., Kroll, N.: Investigation and comparison of implicit smoothers applied in agglomeration multigrid. *AIAA J.* **53**(8), 2080–2096 (2015)
32. Wilcox, D.C.: Reassessment of the scale-determining equation for advanced turbulence models. *AIAA J.* **26**(11), 1299–1310 (1988)
33. Wilcox, D.C.: *Turbulence Modeling for CFD*. CA, 3rd edition, DCW Industries Inc, La Cañada (2006)
34. Wilcox, D.C.: Formulation of the  $k$ - $\omega$  turbulence model revisited. *AIAA J.* **46**(11), 2823–2838 (2008)
35. Menter, F.R.: Improved Two-Equation  $k$ - $\omega$  Turbulence Models for Aerodynamic Flows. Technical report, NASA (1992)
36. Menter, F.R.: Zonal Two Equation  $k$ - $\omega$  Turbulence Models for Aerodynamic Flows. Technical report, NASA (1993)
37. Menter, F.R., Kuntz, M., Langtry, R.: Ten years of industrial experience with the SST turbulence model. *Turbulence Heat Mass Transf.* **4**, 625–632 (2003)
38. <http://turbmodels.larc.nasa.gov>. Turbulence modeling resource website (2013)
39. Swanson, R.C., Rossow, C.-C.: An initial investigation of the effects of turbulence models on the convergence of the RK/implicit scheme. National Aeronautics and Space Administration, Scientific and Technical Information Division (2008)
40. Langer, S.: Preconditioned Newton methods to approximate solutions of the Reynolds averaged Navier–Stokes equations. Technical report, Institut für Aerodynamik und Strömungstechnik (2018)
41. Langer, S.: An initial investigation of solving RANS equations in combination with two-equation turbulence models. Technical report, Institut für Aerodynamik und Strömungstechnik (2019)
42. Blazek, J.: *Computational Fluid Dynamics: Principles and Applications*. Elsevier Science Ltd., Oxford (2001)
43. Langer, S., Martinez, G.S.: Loosely coupled and coupled solution methods for the RANS equations and a one-equation turbulence model. *Comput. Fluids* **232**, 1–17 (2021)
44. Trottenberg, U., Oosterlee, C.W., Schuller, A.: *Multigrid*. Academic Press Inc, Orlando, FL (2001)
45. Saad, Y.: *Iterative Methods for Sparse Linear Systems*. International Thomson Publishing, Boston (1996)
46. Langer, S.: Hierarchy of Preconditioning Techniques for the solution of the Navier–Stokes equations discretized by 2nd order unstructured finite volume methods. In: ECCOMAS 2012 CONFERENCE, Conference Proceeding Series (2012)
47. Golub, G.H., van Loan, C.F.: *Matrix Computations*, 2nd edn. The John Hopkins University Press, Baltimore (1983)
48. Cook, P.H., McDonald, M.A., Firmin, M.C.P.: Aerofoil RAE 2822 pressure distributions and boundary layer and wake measurements. AGARD-AR, 138 (1979)
49. Vassberg, J.C.: A unified baseline grid about the common research model wing-body for the 5th AIAA CFD drag prediction workshop. In: Proceedings of 29th AIAA Applied Aerodynamics Conference, Honolulu, HI, June 2011, number 2011-3508 in Conference Proceeding Series. AIAA (2011)
50. Langer, S.: Investigations of a compressible second order finite volume code towards the incompressible limit. *Comput. Fluids* **149**, 119–137 (2017)



51. Swanson, R.C., Turkel, E., Yaniv, S.: Analysis of a RK/implicit smoother for multigrid. In: Computational Fluid Dynamics 2010: Proceedings of the Sixth International Conference on Computational Fluid Dynamics, ICCFD6, St Petersburg, Russia, on July 12–16, 2010, pp. 409–417. Springer, Berlin (2011)

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.