Implicit numerical simulation of transonic flow through turbine cascades on unstructured grids

Y Mei and **A Guha**[*]
Aerospace Engineering Department, University of Bristol, Bristol, UK

The manuscript was received on 4 August 2004 and was accepted after revision for publication on 26 August 2004.

DOI: 10.1243/095765005X6926

Abstract: Numerical simulation of the compressible flow through a turbine cascade is studied in the present paper. The numerical solution is performed on self-adaptive unstructured meshes by an implicit method. Computational codes have been developed for solving Euler as well as Navier–Stokes equations with various turbulence modelling. The Euler and Navier–Stokes codes have been applied on a standard turbine cascade, and the computed results are compared with experimental results. A hybrid scheme is used for spatial discretization, where the inviscid fluxes are discretized using a finite volume method while the viscous fluxes are calculated by central differences. A MUSCL-type approach is used for achieving higher-order accuracy. The effects of the turbulent stress terms in the Reynolds-averaged Navier–Stokes equations have been studied with two different models: an algebraic turbulence model (Baldwin–Lomax model) and a two-equation turbulence model (k–ω model). The system of linear equations is solved by a Gauss–Seidel algorithm at each step of time integration. A new treatment of the non-reflection boundary condition is applied in the present study to make it consistent with the finite volume flux calculation and the implicit time discretization.

Keywords: upwind, CFD, computation, time-marching, Euler, Navier–Stokes, unstructured grid, self-adaptive, turbulence, turbine, implicit

1 INTRODUCTION

Computational fluid dynamics (CFD) simulation has become an essential tool in the design and analysis of modern turbomachinery components during the past decade. Steady and unsteady state flow predictions are widely studied for problems ranging in size from a single compressor or turbine blade to a complete multistage turbomachine. CFD plays an increasingly important role for it offers the following advantages.

1. Improved designs:
   (a) better understanding of flow from CFD leads to improved designs;
   (b) higher efficiency, lower losses, and wider operating characteristics.
2. More reliable design methods:
   (a) More accurate estimates of performance during the design process;
   (b) less reliance on empirical database and expert knowledge.
3. Quicker development delivery times:
   (a) design in virtual reality requires less hardware and fewer prototypes to be made.
4. Less expensive design process:
   (a) reduction in the number of expensive rig and prototype tests;
   (b) better understanding when something goes wrong.

The continual increases in engine pressure ratio and maximum temperature require accurate predictions of the aerodynamic characteristics and of the heat loads imposed on the blades.

The final purpose of the present research is to investigate the complex physical phenomenon when non-equilibrium wet steam flows through a low-pressure steam turbine. As the first part, the present paper is specifically concerned with dry gas predictions for cascade flow.

An implicit time integration is used here for better numerical stability and convergence. In the past,

[*] *Corresponding author: Aerospace Engineering Department, University of Bristol, University Walk, Bristol BS8 1TR, UK.*

various upwind schemes [**1**–**3**] have been used for calculation of the flux, while the Steger–Warming scheme [**4**] has been used for treating the non-reflecting boundary conditions and also for generating the implicit Jacobian (when implicit time integration was involved). In the present work, a new method is developed such that the calculation of the flux, the boundary conditions, and the implicit Jacobian all conform to the Roe scheme. This provides internal consistency and improves the convergence significantly.

A hybrid scheme is used for spatial discretization, where the inviscid fluxes are discretized using a finite volume method while the viscous fluxes are calculated by central differences. A MUSCL-type approach is used for achieving higher-order accuracy. The effects of the turbulent stress terms in the Reynolds-averaged Navier–Stokes equations have been studied with two different models: the Baldwin–Lomax algebraic turbulence model [**5**] and the two-equation k–ω turbulence model [**6**]. The system of linear equations is solved by a Gauss–Seidel algorithm at each step of time integration.

Unstructured grids are most suitable for complex and irregular geometries. There are two major unstructured grid generation methods: the Delaunay–Voronoi method (DVM) and the advancing front method (AFM) for triangles in two dimensions. Both of the methods are used in the present work, the initial grids are generated for inviscid and viscous computations by DVM and AFM while the self-adaptive - procedure is based on the DVM theory.

2 GOVERNING EQUATIONS

The equations used to model the flow are the compressible, Reynolds-averaged continuity, momentum, and energy equations written in an integral form, where the volume of a computational cell is denoted by V and its surface by S

$$\int_V \frac{\partial \boldsymbol{W}}{\partial t}\, \mathrm{d}V + \int_S \boldsymbol{F}\, \mathrm{d}S - \int_S \boldsymbol{G}\, \mathrm{d}S = 0 \tag{1}$$

where

$$\boldsymbol{W} = [\rho, \rho\boldsymbol{u}, e]^{\mathrm{T}}$$

$$\boldsymbol{F} = [\rho(\boldsymbol{u}\cdot\boldsymbol{n}), \rho\boldsymbol{u}(\boldsymbol{u}\cdot\boldsymbol{n}) + p\boldsymbol{n}, (e+p)(\boldsymbol{u}\cdot\boldsymbol{n})]^{\mathrm{T}}$$

$$\boldsymbol{G} = [0, \boldsymbol{\tau}^S, \boldsymbol{\tau}^S\cdot\boldsymbol{u} + \dot{\boldsymbol{q}}^S]^{\mathrm{T}}$$

$$\boldsymbol{\tau}^S = \begin{bmatrix} \tau_{xx} & \tau_{xy} \\ \tau_{xy} & \tau_{yy} \end{bmatrix}\cdot\boldsymbol{n}, \qquad \dot{\boldsymbol{q}}^S = \begin{bmatrix} \dot{q}_x \\ \dot{q}_y \end{bmatrix}\cdot\boldsymbol{n}$$

By assuming that the fluid is a perfect gas, the pressure p is calculated from

$$p = (\gamma - 1)(e - \tfrac{1}{2}\|\boldsymbol{u}\|^2)$$

The shear stress tensor, τ_{ij}, is given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)$$

The above system of equations needs initial conditions prescribing the flow state at $t = 0$ and boundary conditions. In the time marching method the final steady state is achieved as the converged solution of unsteady calculations.

3 NUMERICAL METHOD

The governing equations are treated in conservative form and discretized in time using the Euler implicit method leading to a set of non-linear finite difference equations, which are solved using a Newton procedure. In stationary simulations, convergence history is accelerated by using local time steps which are based on local stability criteria.

Applying Green's theorem, the Navier–Stokes equations can be rewritten in a differential form

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla\cdot\boldsymbol{F} = \nabla\cdot\boldsymbol{G} \tag{2}$$

3.1 Inviscid fluxes

The convective (Euler) parts are discretized using a third-order accuracy, TVD-upwind finite volume scheme. When solving transport equations for turbulent quantities, a more stable first-order upwind scheme is used. The TVD scheme is based on the MUSCL-type of upwind scheme [**7**] which consists of a projection stage and an evolution stage. In the projection stage, left and right states at each control volume interface are determined by extrapolating the node values of the conservative variables towards the control volume interface. In the evolution stage the inviscid flux is evaluated by solving the Riemann problem between left and right states using Roe's approximate Riemann solver [**2**].

3.2 Viscous fluxes

In order to construct the numerical viscous flux vector at the control volume interfaces, it is necessary to evaluate first-order derivatives of the velocity components, the speed of sound, and the turbulent quantities, which is done in a central-differences manner, using Green's theorem.

3.3 Roe approximate Riemann solver

Consider Roe's flux difference splitting of the inviscid flux vector

$$\boldsymbol{F}(\boldsymbol{W}_{\mathrm{L}},\boldsymbol{W}_{\mathrm{R}}) = \tfrac{1}{2}[\boldsymbol{F}(\boldsymbol{W}_{\mathrm{L}}) + \boldsymbol{F}(\boldsymbol{W}_{\mathrm{R}}) - \tfrac{1}{2}|\tilde{\mathbf{A}}(\boldsymbol{W}_{\mathrm{L}},\boldsymbol{W}_{\mathrm{R}})| \cdot (\boldsymbol{W}_{\mathrm{L}} - \boldsymbol{W}_{\mathrm{R}})] \tag{3}$$

where $|\tilde{\mathbf{A}}(\boldsymbol{W}_{\mathrm{L}},\boldsymbol{W}_{\mathrm{R}})|$ denotes the standard Roe matrix. It can be shown that this scheme is equivalent to the first-order finite volume upwind cell–vertex scheme based on a dual mesh. There are many different ways to achieve higher-order accuracy. In this work, a scheme of higher-order accuracy is obtained by using upwind-biased interpolations of the solution $\boldsymbol{W}_{\mathrm{L}},\boldsymbol{W}_{\mathrm{R}}$ via the MUSCL approach. Therefore, the flux function is shown as

$$\boldsymbol{F}_{\mathrm{LR}} = \tfrac{1}{2}[\boldsymbol{F}_{\mathrm{L}}^{+} + \boldsymbol{F}_{\mathrm{R}}^{-} - |\tilde{\mathbf{A}}(\boldsymbol{W}_{\mathrm{L}}^{+},\boldsymbol{W}_{\mathrm{R}}^{-})| \cdot (\boldsymbol{W}_{\mathrm{L}}^{-} - \boldsymbol{W}_{\mathrm{R}}^{+})] \tag{4}$$

where

$$\boldsymbol{F}_{\mathrm{L}}^{+} = \boldsymbol{F}(\boldsymbol{W}_{\mathrm{L}}^{+}), \qquad \boldsymbol{F}_{\mathrm{R}}^{-} = \boldsymbol{F}(\boldsymbol{W}_{\mathrm{R}}^{-})$$

and $\boldsymbol{W}_{\mathrm{L}}^{+}$ and $\boldsymbol{W}_{\mathrm{R}}^{-}$ are constructed by the MUSCL scheme as

$$\boldsymbol{W}_{\mathrm{L}}^{+} = \boldsymbol{W}_{\mathrm{L}} + \frac{S_{\mathrm{L}}}{4}[(1 - kS_{\mathrm{L}})\,\Delta_{\mathrm{L}}^{-} + (1 + kS_{\mathrm{L}})(\boldsymbol{W}_{\mathrm{L}} - \boldsymbol{W}_{\mathrm{R}})]$$
$$\boldsymbol{W}_{\mathrm{R}}^{-} = \boldsymbol{W}_{\mathrm{R}} - \frac{S_{\mathrm{R}}}{4}[(1 - kS_{\mathrm{R}})\,\Delta_{\mathrm{R}}^{+} + (1 + kS_{\mathrm{R}})(\boldsymbol{W}_{\mathrm{R}} - \boldsymbol{W}_{\mathrm{L}})]$$

where S is the flux limiter of von Albabda

$$S_i = \frac{2\Delta_i^{-}(u_j - u_i) + \varepsilon}{(\Delta_i^{-})^2 + (u_j - u_i)^2 + \varepsilon}$$
$$S_j = \frac{2\Delta_j^{+}(u_j - u_i) + \varepsilon}{(\Delta_j^{+})^2 + (u_j - u_i)^2 + \varepsilon}$$

with $\varepsilon \to 0$.

The forward and backward difference operators are given by

$$\Delta_{\mathrm{L}}^{+} = 2\boldsymbol{W}_{\mathrm{L}}, \qquad \Delta_{\mathrm{R}}^{-} = 2\boldsymbol{W}_{\mathrm{R}}$$

The parameter k is chosen to control the degree of approximation. In this paper, $k = 1/3$, which corresponds to a third-order upwind-biased scheme. When $k = 0$ it is a Fromm scheme, and $k = -1$ corresponds to a second-order upwind-biased scheme.

One of the strategies is to develop a multidimensional limiter for modifying high-order finite volume schemes in such a way that the resulting method is monotonic, thus avoiding the creation of unphysical oscillations in the numerical solution and so improving the robustness of the algorithm. One commonly used approach is the slope limiting (MUSCL) technique of van Leer, in which the limiter is applied in a geometric manner to the gradients of a piecewise linear reconstruction of the solution to create a monotonic scheme.

3.4 Boundary condition

In the present scheme, phantom cells are used to handle inlet and outlet boundaries. According to the theory of characteristics, flow angle, total pressure, total temperature, and isentropic relations are used at the subsonic axial inlet, whereas all variables are prescribed at the supersonic inlet. At the subsonic axial outlet the average value of the static pressure is prescribed, whereas all variables are calculated at the supersonic axial outlet. For Euler equations, the slippery adiabatic or isothermal condition is applied on solid walls, and for NS equations the non-slip condition is used instead. All details are shown in Fig. 1.

1. *Adiabatic impermeable solid wall*: $\boldsymbol{u} \cdot \boldsymbol{n} = 0$ for inviscid flow computation; $\boldsymbol{u} = 0$ and $\partial T/\partial n = 0$ for viscous flow computation.
2. *Periodic boundaries*. Periodic boundaries are not necessary if the whole bladerow is modelled, but, in practice, computer resources are limited, and only one blade passage of the whole flow domain is modelled to save computational time. This treatment is sufficient since flow data can be transferred directly between periodic boundaries, and the periodic boundary condition is prescribed on artificial cuts 2.

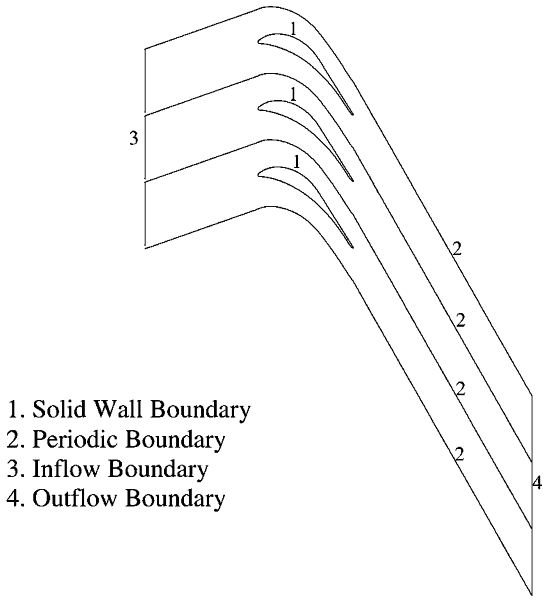

Fig. 1 Nomenclature for cascade of profiles SE1050

3. *Inlet boundary*. Total pressure p_0, total temperature T_0, and the inflow angle of attack α (or incidence angle i) are specified as the input conditions.
4. *Outlet boundary*. Outlet static pressure p_2 is specified. One option in the numerical simulation is to set the static pressure at the outlet boundary as an invariant given by the input condition; the other option is to apply non-reflection treatment on the outlet boundary as it has been successfully applied on the farfield condition of the external flow computation. This method is supported by numerical results of Yao *et al.* [**8**]. Both of these methods are used in the present paper, and the numerical results based on non-reflecting conditions are shown to be closer to that of the real flow (i.e. experimental results).

Given the incoming Mach number, M_∞, and the angle of attack, α, in the external computation, the rest of the farfield properties are fixed. This makes it easy to apply the Roe scheme on the farfield boundary calculation simply by treating the farfield phantom cells as the left-hand-side properties while treating the real values as the right-hand-side properties. This strategy makes the method that is used to compute the fluxes contributed by the boundaries the same as the method used in interior flux computation. The internal consistency of treatments results in a large CFL number in the implicit algorithm and will be discussed later in this paper. Unfortunately, unlike in the external computation, the inlet and outlet boundary conditions are expressed implicitly in the internal flow computation. The total pressure P_0, the total temperature T_0, and the incident angle i are fixed at the entrance, and the outlet static pressure p_2 is fixed at the exit. This does not provide enough information to specify the flow properties which are necessary for applying the Roe scheme at the inlet and outlet boundary. To overcome this problem and to maintain the advantage of the consistency property at the same time, a two-step prediction–correction procedure is implemented here.

1. *Prediction*. There are three inlet boundary conditions, p_0, T_0, and α, and the conservative variables will be specified explicitly if given another extra independent condition such as the inlet mach number, M_1, or the inlet static pressure, p_1. For convenience of calculation, M_1 is chosen as the extra parameter to specify the predicted values at the inlet phantom cells with the isentropic assumption here; M_1 can easily be calculated from the local inlet boundary nodes. At the outlet boundary it is found that it would cause serious reflection from the outlet boundary if p_2 were set as a constant at the outlet boundary nodes. To avoid violation of the non-reflection boundary condition, p_2 at the outlet boundary is kept flexible while p_2 at the phantom cells is fixed. There may be a loss of the total pressure after passing through the cascade profile. $p_{0.2}$ and $T_{0.2}$ at the right-hand-side of the boundary (phantom cells) are taken the same as $p_{0.2}$ and $T_{0.2}$ at the left-hand-side, which are calculated from $\boldsymbol{W}^n$ at the outlet boundary. This, together with specified p_2 at the right-hand-side, enables the Roe scheme to calculate all properties at the outlet boundary.
2. *Correction*. After upgrading the conservative variables from $\boldsymbol{W}^n$ to $\boldsymbol{W}^{n+(1/2)}$, a correction should be imposed on the inlet boundary nodes to make them satisfy the constant total pressure condition and constant total temperature at the inlet boundary. This can easily be realized by calculating the Mach number M_1 from the conservative values $\boldsymbol{W}^{n+(1/2)}$ at the inlet boundary and then updating $\boldsymbol{W}^{n+1}$ from M_1, p_0, T_0, and the inflow angle α.

Usually, the Steger–Warming scheme is used for the farfield boundary condition. In this paper, the farfield boundary condition is formulated such that it is internally consistent with the Roe scheme. For this, equation (3) is rewritten with the subscript R replaced by ∞ (the farfield condition)

$$\boldsymbol{F}(\boldsymbol{W}_\mathrm{L},\boldsymbol{W}_\infty) = \tfrac{1}{2}[\boldsymbol{F}(\boldsymbol{W}_\mathrm{L}) + \boldsymbol{F}(\boldsymbol{W}_\infty) - \tfrac{1}{2}|\tilde{\mathbf{A}}(\boldsymbol{W}_\mathrm{L},\boldsymbol{W}_\infty)| \cdot (\boldsymbol{W}_\mathrm{L} - \boldsymbol{W}_\infty)] \tag{3a}$$

The same equation applies for inflow (subscript 1) and outflow (subscript 2) farfeld conditions. Later it will be shown that this internally consistent treatment of boundary conditions improves the convergence significantly.

3.5 Implicit time integration

Equation (1) can be rewritten in semi-discrete form for each control volume

$$V_i = \frac{\partial \boldsymbol{W}}{\partial t} = R_i \tag{5}$$

where V_i is the area for two dimensions and the volume for three dimensions of the dual mesh cell, and R_i is the residual. The Euler implicit discretization and linearization of equations leads to

$$\left(\frac{V}{\Delta t} + \frac{\partial \boldsymbol{F}}{\partial \boldsymbol{W}}\right)^n \Delta \boldsymbol{W}^n = R^n \tag{6}$$

where Δt is the time increment, $\Delta \boldsymbol{W}^n$ is the difference in the conservation variable between time levels n and $n+1$, and $\partial \boldsymbol{F}/\partial \boldsymbol{W}$ represents symbolically the

Jacobian matrix. The Gauss–Seidel algorithm is applied in solving the linear equations at each step of time integration.

The present work makes novel use of the Roe scheme to calculate the implicit Jacobian matrix (in addition to its standard use for space discretization). The inviscid flux Jacobian matrix is thus written as

$$\frac{\partial \boldsymbol{F}_{\mathrm{LR}}}{\partial \boldsymbol{W}_{\mathrm{L}}} = \tfrac{1}{2}[\mathbf{A}(\boldsymbol{W}_{\mathrm{L}}) + |\tilde{\mathbf{A}}|], \quad \frac{\partial \boldsymbol{F}_{\mathrm{LR}}}{\partial \boldsymbol{W}_{\mathrm{R}}} = \tfrac{1}{2}[\mathbf{A}(\boldsymbol{W}_{\mathrm{R}}) - |\tilde{\mathbf{A}}|] \tag{7}$$

For the farfield boundary condition the Jacobian matrix is

$$\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{W}_i} = \tfrac{1}{2}[\mathbf{A}(\boldsymbol{W}_i) + |\tilde{\mathbf{A}}(\boldsymbol{W}_i, \boldsymbol{W}_\infty)|] \tag{8}$$

Therefore, all computational nodes in the flowfield, including the boundary points, are discretized by the Roe scheme.

In the Navier–Stokes calculation, the boundary conditions on the body correspond to no slip and the velocities at the solid boundary are known as 0 and should not be treated as variables on the LHS of the equations. A post-processor is utilized to implement these boundary conditions by modifying the matrix terms in equation (6) to reflect appropriately the desired boundary conditions. For clearer demonstration of this procedure, a slightly expanded representation of one of the rows in equation (6) is given by

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ A_{21} & A_{22} & A_{23} & A_{24} \\ A_{31} & A_{32} & A_{33} & A_{34} \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta e \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ R_4 \end{bmatrix} \tag{9}$$

where R represents both the residual and off-diagonal terms on the right-hand side of equation (6) and A_{ij} represents the individual components of one of the diagonal blocks in [**A**]. The density can be determined from the continuity equation during the solution process from the first row of equation (9). However, the contribution to the continuity residual along the boundary involves integration around the dual mesh surrounding the node and a segment of the body surface. The contribution from the surface, assuming zero velocity at the wall, is identically zero. The second and third rows are modified so that the solution of equation (9) maintains a zero velocity at the nodes on the solid boundaries. Furthermore, the fourth row is altered to preserve a constant temperature in the case of an isothermal wall boundary condition. The constant wall temperature assumption is used to relate the change in energy at the wall to the change in density

$$\Delta e = \frac{T_{\mathrm{wall}}}{\gamma(\gamma - 1)} \Delta\rho$$

The resulting matrices now reflect the implementation of the appropriate boundary conditions at the wall and are given by Anderson and Bonhaus [**9**].

1. Adiabatic wall

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ A_{41} & A_{42} & A_{43} & A_{44} \end{bmatrix} \begin{bmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta e \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \\ 0 \\ R_4 \end{bmatrix} \tag{10}$$

2. Isothermal wall

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} & A_{14} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -T_{\mathrm{wall}}/\gamma(\gamma - 1) & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\rho \\ \Delta\rho u \\ \Delta\rho v \\ \Delta e \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{11}$$

3.6 Turbulence model

The algebraic turbulence model and the two-equation model are used here.

3.6.1 Baldwin–Lomax algebraic turbulence model

The Baldwin–Lomax (B–L) turbulence model is the mixing length model owing to the algebraic relationship which uses Prandtl mixing length theory for determination of the eddy viscosity. This model is easy to implement and can provide accurate results for simple cases. It is a two-layer algebraic eddy viscosity model in which μ_{t} is given by

$$\mu_{\mathrm{t}} = \begin{cases} (\mu_{\mathrm{t}})_{\mathrm{inner}} y \leqslant y_{\mathrm{crossover}} \\ (\mu_{\mathrm{t}})_{\mathrm{outer}} y > y_{\mathrm{crossover}} \end{cases}$$

where y is the normal distance from the wall and the smallest value of y at which values from the inner and outer formulation are equal.

The Prandtl–van Driest formulation is used in the inner region

$$\nu_{\mathrm{T}}^{(\mathrm{i})} = \rho l^2 |\nabla \times \boldsymbol{u}|$$

where

$$l = Ky(1 - \mathrm{e}^{-(y^+/A)})$$
$$y^+ = \frac{y}{v}\sqrt{\frac{\tau_0}{\rho}}$$

and ρ is the density, $\boldsymbol{u}$ is the velocity of the field, the shear stress tangential to the wall

$$\tau_0 = \mu \left.\frac{\partial \boldsymbol{u}}{\partial y}\right|_w$$

and A is the van-Driest constant ($A = 26$).

In the outer region, the eddy viscosity is given by

$$(\mu_\mathrm{t})_\mathrm{outer} = 0.0168\beta\frac{F_\mathrm{wake}}{F_\mathrm{kleb}}$$

where F_wake is determined from

$$F_\mathrm{wake} = \min\left\{\begin{array}{c} y_\mathrm{max}F_\mathrm{max} \\ 0.25y_\mathrm{max}U_\mathrm{dif}^2/F_\mathrm{max}\end{array}\right\}$$

the quantity U_dif is the difference between the maximum and minimum total velocity in the velocity profile

$$U_\mathrm{dif} = (\sqrt{u^2 + v^2})_\mathrm{max} - (\sqrt{u^2 + v^2})_\mathrm{min}$$

y_max is the distance from the wall at which $\boldsymbol{F}(y)$ becomes maximum, and F_max is the maximum value of the function $\boldsymbol{F}(y)$, which is given as

$$\boldsymbol{F}(y) = y(1 - \mathrm{e}^{-y^+/A})|\nabla \times \boldsymbol{u}|$$

The intermittency factor is given by

$$\boldsymbol{F} = \frac{1}{1 + 5.5(0.3y/y_\mathrm{max})}$$

Generally, the BL model is designed on the basis of structured grid computation because it needs the normal distance from the wall. To adapt it for unstructured grids, a modification to the original BL model has been made. A turbulence reference grid is used to supply the necessary length scale for each cell. It is constructed by dividing the computational domain into strips, or narrow bands, which emanate from the body walls or some wake centre-lines. The width of each strip should be of the order of 1–2 cell sizes. The control volumes whose nodes reside in one particular strip are treated as one group of nodes. In each group, the distance of each node to the end-wall or the wake centre-line can be measured, which, in turn, will be used as the reference length in the turbulence model. Once the reference length for each cell is established, computation of the turbulence model can proceed on the unstructured mesh without difficulty. For example, the vorticity at each node can be obtained directly from the velocity gradient computed by the flow solver. The peak of a functional distribution can be located by searching the extremum points along each strip or within each group. The turbulence eddy viscosity at each node can then be determined easily by the two-layer algebraic mode.

Ideally, the divided strips should cover the entire computational domain without overlapping each other. However, for the flowfield with multiple walls (the pressure and suction surfaces of the blades) or wake lines, regions of overlapping strips will occur. This is equivalent to saying that the turbulence in these regions is influenced by multiple turbulence sources. In this case, the turbulence computation within each group still proceeds as before, but a weighting procedure is employed to determine the influence from each wall or wake line. A more sophisticated approach may take the distance-weighted average as the final turbulence eddy viscosity.

3.6.2 Two-equation turbulence model (k–ω model)

The main field of application of Navier–Stokes methods in aerodynamics will be for complex turbulent flows that cannot be treated by an inviscid or viscous–inviscid interaction scheme. Examples are massively separated flows, flows involving multiple length scales, flows with three-dimensional separation, and complex unsteady flows. In these flows the application of algebraic turbulence models such as the Baldwin–Lomax model becomes very complicated owing to the difficulty in defining an algebraic length scale. It is obvious that the improvement of numerical methods must be accompanied with the development of more general turbulence models and their implementation in Navier–Stokes codes.

The most popular non-algebraic turbulence models are two-equation eddy viscosity models. These models solve two transport equations, generally one for the turbulent kinetic energy and another one related to turbulence length (or time) scale. Among the two-equation models, the k–ω turbulence model of Wilcox is one of the most successful. It solves one equation for the turbulence kinetic energy, k, and a second equation for the specific turbulent dissipation rate, ω. The model does not employ damping functions and has straightforward Dirichlet boundary conditions. This leads to significant advantages in numerical stability. The standard Wilcox k–ω model is given as follows.

For the turbulence kinetic energy k transport equation

$$\frac{\partial \rho k}{\partial t}+\frac{\partial \rho \boldsymbol{u}_i k}{\partial x_i}=\tau_{ij}\frac{\partial \boldsymbol{u}_i}{\partial x_j}-\beta^* \rho\omega k+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_{k1}\mu_{\mathrm{t}})\frac{\partial k}{\partial x_j}\right]$$

For the turbulent dissipation rate ω transport equation

$$\frac{\partial \rho\omega}{\partial t}+\frac{\partial \rho \boldsymbol{u}_i \omega}{\partial x_i}=\frac{\gamma_1}{\nu_{\mathrm{t}}}\tau_{ij}\frac{\partial \boldsymbol{u}_i}{\partial x_j}-\beta_1\rho^2 k$$
$$+\frac{\partial}{\partial x_j}\left[(\mu+\sigma_{\omega1}\mu_{\mathrm{t}})\frac{\partial \omega}{\partial x_j}\right]$$

where the constants will be

$$\sigma_{k1}=0.5, \qquad \sigma_{\omega 1}=0.5, \qquad \beta_1=0.0750$$
$$\beta^*=0.09, \qquad \kappa=0.41, \qquad \gamma_1=\frac{\beta_1}{\beta^*}-\frac{\sigma_{\omega1}\kappa^2}{\sqrt{\beta^*}}$$

The model has to be supplemented by the definition of the eddy viscosity

$$\nu_{\mathrm{t}}=\frac{\mu_{\mathrm{t}}}{\rho}=\frac{k}{\omega}$$

The turbulent stress tensor $\tau_{ij}=-\rho\overline{u_i'u_j'}$ is then given by

$$\tau_{ij}=\mu_t\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}-\frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)-\frac{2}{3}\rho k\delta_{ij}$$

For the turbulent equations, the boundary conditions are needed. For the wall conditions, no slip conditions are used. The turbulent kinetic energy k is zero, and ω satisfies the following equation [**6**]

$$\omega=10\frac{6\nu}{\beta_1(\Delta y)^2}$$

where Δy is the distance to the first point away from the wall. For freestream conditions, the turbulent kinetic energy k is set at 10^{-7}, and the turbulent coefficient is 0.01 (non-dimensional), so for the freestream $\omega=Re/(M_\infty\mu_\infty)\rho_\infty\kappa_\infty$.

This model has been widely used. It is wall distance free, which is very important for the unstructured grid, and does not need damping functions in the viscous sublayer. However, this model is sensitive to the value specified for the freestream specific dissipation, ω_∞. The turbulent coefficient changes greatly, but for wall-bounded flow this is not very significant.

3.7 Computational grids

All computational unstructured grids are generated by either the Delaunay–Voronoi method (DVM) or the advancing front method (AFM). Thin grids are necessary to capture the flow details in the boundary layer, and for this purpose the advancing layer method is applied in generating skew meshes inside the boundary layer when the solver is considering the effect of viscosity. Figures 2 and 3 are the unstructured grids generated for Euler equation calculations. Figure 2 is the original grid generated

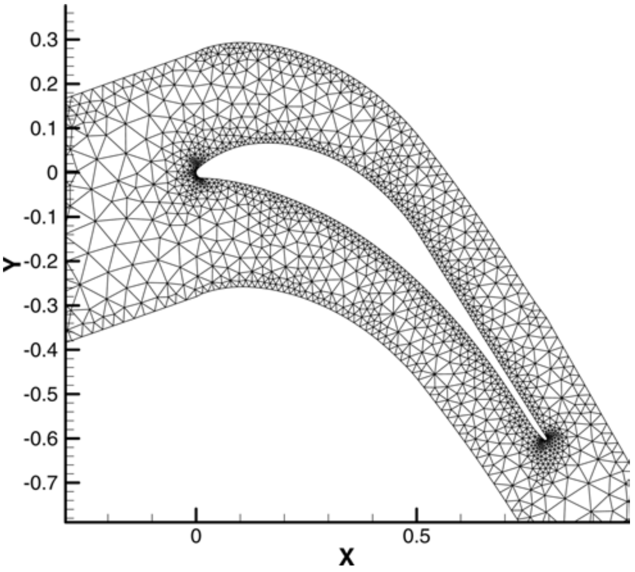

Fig. 2 Initial mesh around the SE1050 blade for the Euler solver

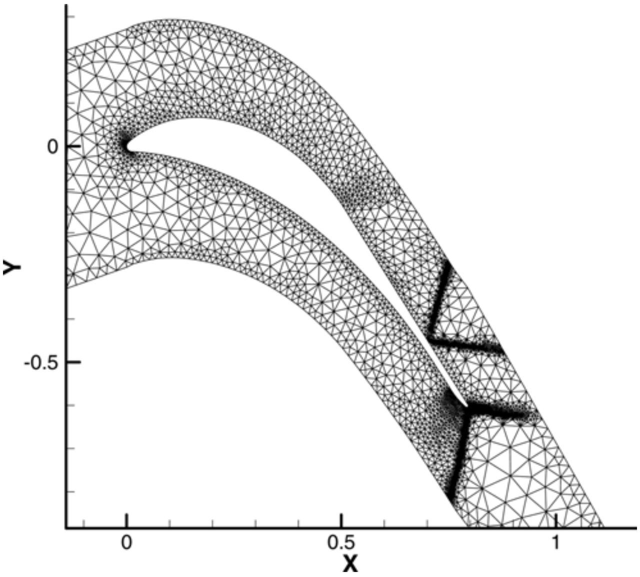

Fig. 3 Computational grid for the Euler solver after fivefold refinement

by DVM, and Fig. 3 is the grid after fivefold refinement.

4 NUMERICAL RESULTS

The final objective of the present research is to investigate the aerodynamic and thermal properties of the effect of homogeneous condensation flow on the performance of the cascade. The focus here is upon single-phase flow through a two-dimensional steam turbine cascade. The transonic rotor turbine profile cascade SE1050 [**10**], which was designed for the last stage of a ŠKODA steam turbine, is chosen as the test case here. Extensive measurements using this profile have been made for a large number of operating conditions. A rigorous definition of measured quantities is given in reference [**10**]. This provides a useful comparison with numerical computations. In addition, this cascade satisfies the velocity range used in modern gas turbines.

The physical experiment was performed under three conditions including the design condition and two off-design conditions. The numerical experiments were performed at the design condition for the following data: angle of attack $\alpha = 19.3°$ (corresponding to the incident angle $i = 0°$), total pressure $p_0 = 96\,748\,\text{Pa}$, total temperature $T_0 = 296.65\,\text{K}$, Reynolds number $Re = 1.48 \times 10^6$, and isentropic outlet Mach number $M_{2,\text{is}} = 1.198$.

4.1 Inviscid computation

These computations were carried out to validate the inviscid part of the solver. The original mesh used for the present computation consists of 2605 nodes, of which 222 nodes lie on the blade surface, and 4508 cells, as shown in Fig. 2. It takes 22 time steps to converge to −5 orders with the Gauss–Seidel implicit algorithm. The fine grid after fivefold self-adaptation contains 8427 nodes and 16 062 cells, see Fig. 3. The advantage of the interior consistency of the inviscid solver is obvious: the CFL number can be as large as 10 000 and still the scheme maintains stability. Since the computational error is proportional to $O(\Delta t)$, too large a CFL number would result in a large numerical error. Therefore, the maximum CFL number is set at 200 during the calculation to achieve a compromise between efficiency and accuracy.

The calculated surface pressure distribution along the chord is plotted in Fig. 4. The pressure coefficient C_p is defined as p/p_0. From Fig. 4 it can be seen that the pressure changes slowly along the pressure side while it changes rapidly along the suction side. The pressure downstream of the reflection shock wave that is generated from the trailing edge of the suction side of the blade corresponds to the outlet static pressure p_2 and remains approximately constant until the arc of the trailing edge is reached. The convergence histories of the computations are compared in Fig. 5. The first test case is a Roe scheme carried out on a coarse grid, the second is a Roe scheme on a fine grid, and the third is an Osher scheme on the same fine grid. The Roe method is applied at the inflow and outflow boundaries in all

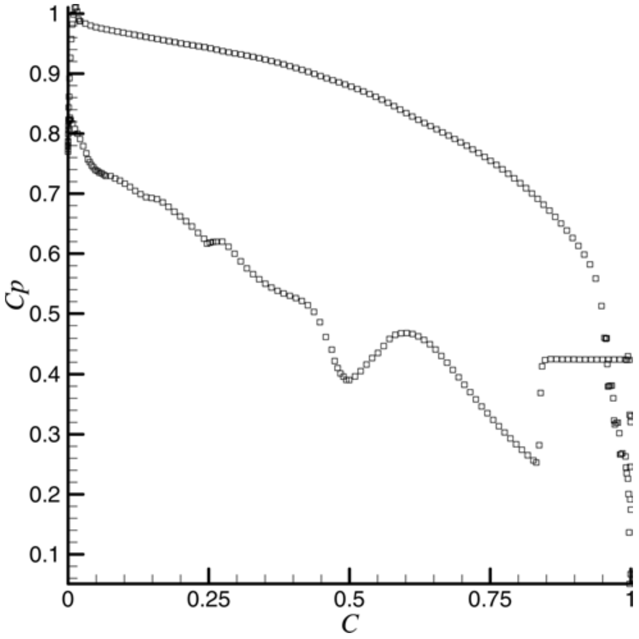

Fig. 4 Surface pressure distribution along the chord of SE1050, calculated by the present Euler solver

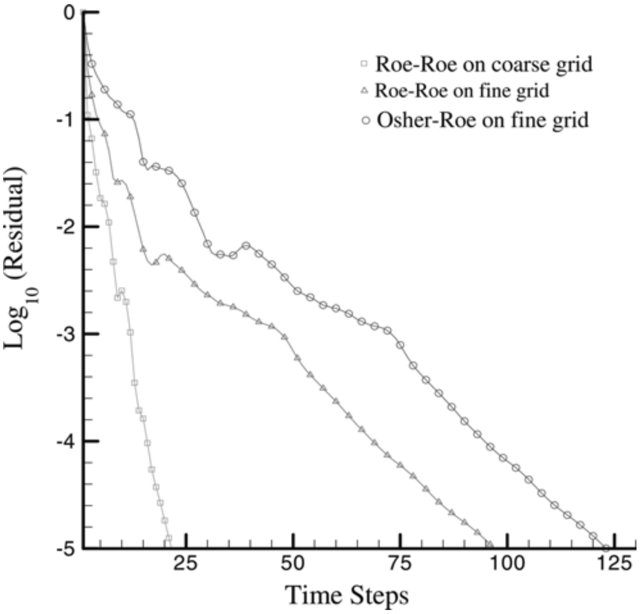

Fig. 5 Convergence history of the computations on coarse and fine grids for various schemes for internal nodes and boundary nodes

three test cases. The Roe–Roe method on a coarse grid shows the best convergence history because of the large artificial viscosity brought by the large grid. Comparison between the Roe–Roe and Osher–Roe methods on the self-adaptive grid confirms that internal consistency of the solver will provide a more efficient computation.

Šafarik suggested that the relative velocity λ be used as a standard parameter in the internal flow computation. Here, λ is defined as the ratio of velocity and the critical speed of sound

$$\lambda = \frac{u}{\sqrt{2[\gamma/(\gamma+1)]RT_0}}$$

A comparison of the computed surface relative velocity distribution with the experimental data of Štastny and Šafarik [10] is shown in Fig. 6. In this figure, the results denoted by circles have been obtained with the inviscid computation while those denoted by the squares are obtained from the experiment. Since the real velocities on the solid wall are zero, the relative velocity data in the experiment were obtained at the edge of the boundary layer.

The agreement between the computations and the experiment is good over most of the blade, although an overexpansion near the trailing edge is evident in the computations which does not appear in the experiment. The round trailing edge of the SE1050 blade profile causes this difficulty in the computation of Euler equations. Denton and Dawes [11] found that, with the coarse grid, it is not necessary to apply the Kutta condition explicitly and the numerical viscosity in the solution will automatically damp the flow and make it leave the trailing edge smoothly. However, with the refinement of the grids after several self-adaptive processes, more nodes are set on the trailing edge automatically for accurately capture of the dramatic changes. The flow then tries to approach a stagnation point on the trailing edge circle and the point where the Kutta condition should be imposed is not well defined. Then, the predicted flow is found to accelerate to high speed at the start of the trailing edge circle on both pressure and suction surfaces. From the numerical point of view, the high-speed flows from the pressure side and the suction side clash somewhere at the trailing edge with opposite directions and stop at the stagnation point all of a sudden. The dramatic change in the physical properties occurs within a few grids occupying a short distance along the trailing edge and destabilizes the numerical robustness.

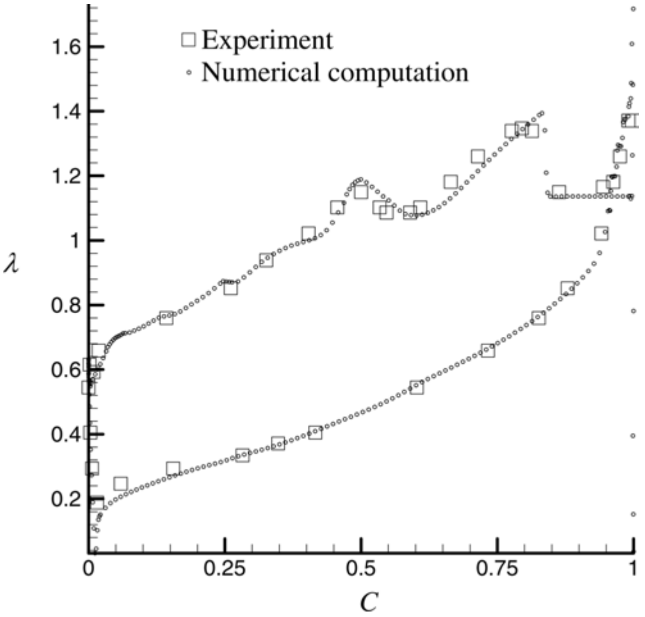

Fig. 6 Comparison between the experimental and numerical relative surface velocity λ along the chord of SE1050 ($M_{2,\mathrm{is}} = 1.198$, $\alpha = 19.3°$)

To obtain a desirable solution, Denton recommended that the trailing edge of the blade be extended along the streamline, so the round shape of the trailing edge is changed into a sharp one making the flow leave the trailing edge smoothly. This technology is known as CUSP, and it simplifies the complexity of the computation and does not influence the flow significantly. The CUSP should be unloaded, so that it only introduces blockage without exerting tangential force on the flow. Unfortunately, there is no theoretical instruction about how to create such a CUSP, and experience is needed in its proper implementation.

Figures 7 and 8 represent the Mach contours of the numerical and experimental results respectively. Figure 8 shows the interferometric picture with lines of constant Mach number. Several shock waves, boundary layers, wakes, and expansion waves are observed. (The figure presents the frame around trailing edges of a cascade of profiles.) The ‘fishtail’ shock waves are seen from both the pressure and suction sides of the trailing edge circle. The shock wave generated from the pressure side is very strong and hits the suction side of its neighbouring blade. It can be seen from a comparison of Figs 7 and 8 that the solver captured exact positions of the reflection shock wave.

4.2 Viscous computation

Many (~20 in the present calculation) layers of nodes are distributed in the boundary layer to simulate the real flow more accurately. Both the Baldwin–Lomax algebraic turbulence model and the two-equation k–ω model are incorporated in the developed solver. The initial computational grid is plotted in Fig. 9, while the surface pressure distribution and the Mach number contour are shown in

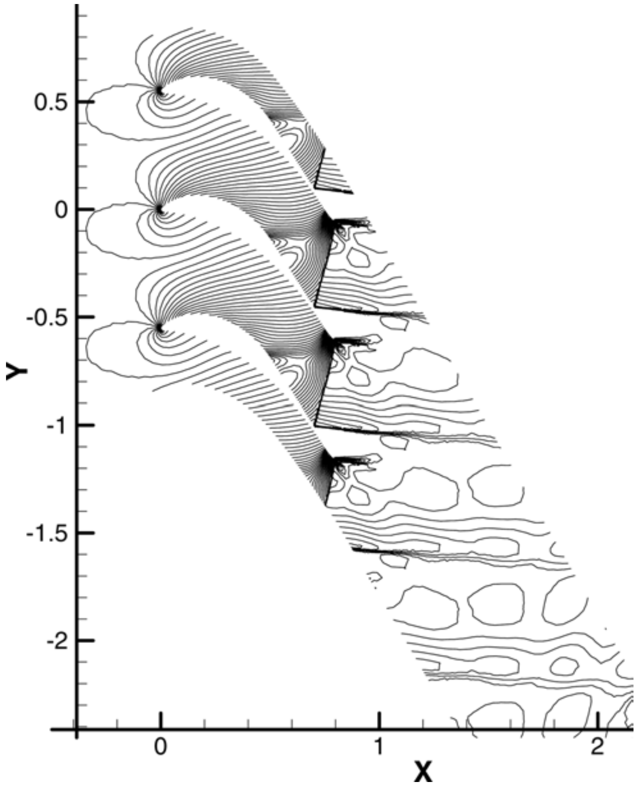

Fig. 7 Mach contour on the SE1050 blade cascade, calculated by the present Euler solver ($\alpha = 0$, $M_{2,\text{is}} = 1.198$)

Figs 10 and 11 respectively. The operating condition is the same as the inviscid computation, with the Reynolds number as an additional parameter: incident angle $i = 19.3^\circ$, $M_{2,\text{is}} = 1.198$, $Re = 1.48 \times 10^6$, total pressure $P_0 = 99\,805$ Pa, and total temperature $T_0 = 300$ K.

Comparing the pressure distribution of the inviscid result in Fig. 4 with the viscous result in Fig. 10, it is interesting to see that the oscillation at the trailing edge disappeared. In the viscous computation, the velocity on the solid wall is fixed at zero, so it reduces non-physical oscillation and enforces stability of the calculation. Comparison between Figs 4 and 10 shows that the shock wave moves towards the leading edge, which results from boundary layer blockage.

Different self-adaptive sensors are applied in the grid refinement procedure for efficiently detecting the positions of the shock wave and the wake. The shock is characterized as an abrupt jump in pressure, density, and other flow variables. Therefore, the density gradient is chosen as the shock wave sensor to indicate the position of the shock wave. Unfortunately, this sensor does not detect the wake, because neither the pressure nor the density changes significantly in the wake. To capture the wake, which is dominated by the viscosity, various rotation-related sensors are constructed and tested as follows.

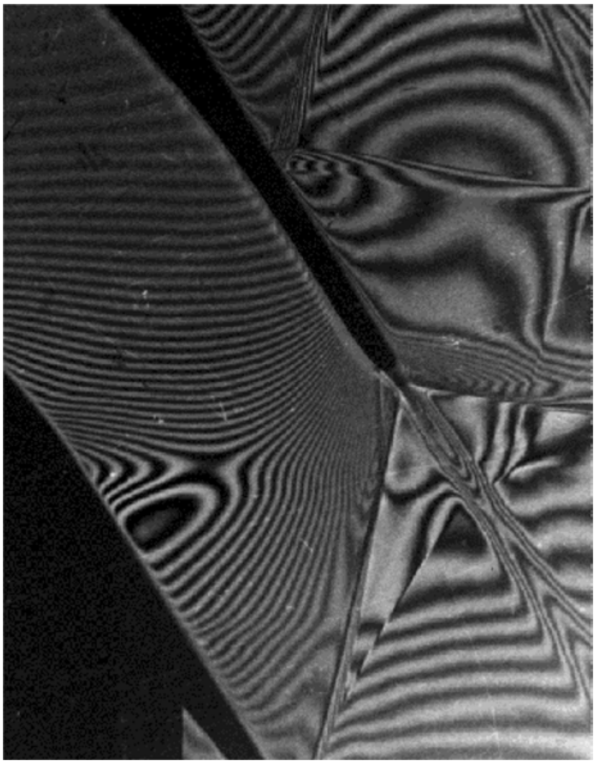

Fig. 8 Interferometric picture of flow in the SE1050 cascade ($\alpha = 0$, $M_{2,\text{is}} = 1.198$) (courtesy of P. Šafarik)

1. *Primitive rotation variable ω.* The primitive rotation variable, ω, is chosen as the first wake sensor. As expected, the wake is captured in the

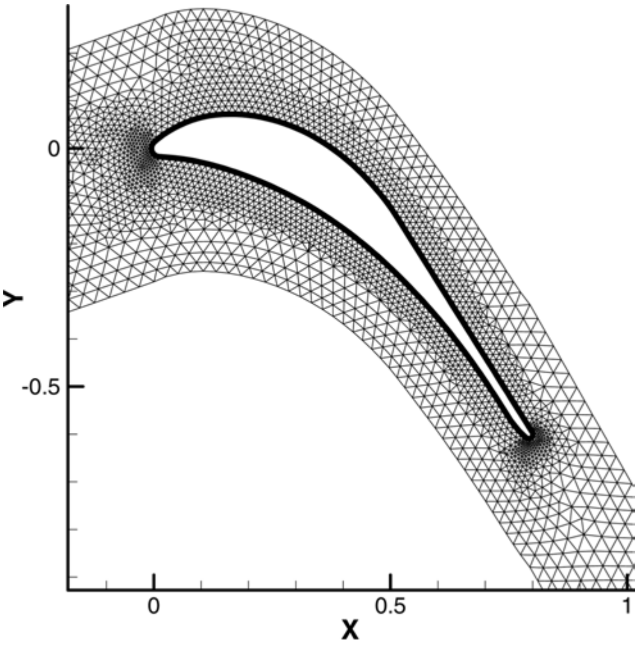

Fig. 9 Initial computational grid around the SE1050 blade for the NS solver

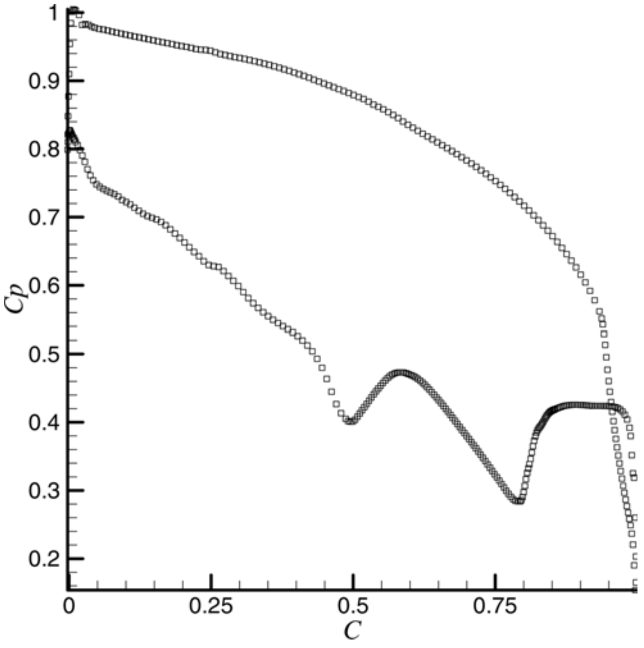

Fig. 10 Pressure distribution along the chord of the SE1050 blade with a k–ω model based on the initial grid

first three generations of self-adaptation. However, unfortunately, the new nodes are added such that the dense region of grids extends like a fork within the wake, the effect becoming more prominent as more generations of self-adaptation are employed. The reason for this non-uniform distribution is that the transverse gradient of speed near the centre-line of the wake decreases in the axial downstream direction. This means that the rotation variable ω near the centre-line of the wake decreases more rapidly than that of the other area in the wake.

2. *New wake sensor* $\omega/(u^2 + V^2)$. To get rid of the 'fork' and predict an accurate position of the wake, a new wake sensor is created. Considering that the velocities on the centre-line are relatively small compared with those on the other area of the wake, the new wake-sensitive sensor is devised as $\omega/(u^2 + v^2)$. The numerical tests in the present study showed that this sensor is more effective for self-adaptation of grids within the wake.

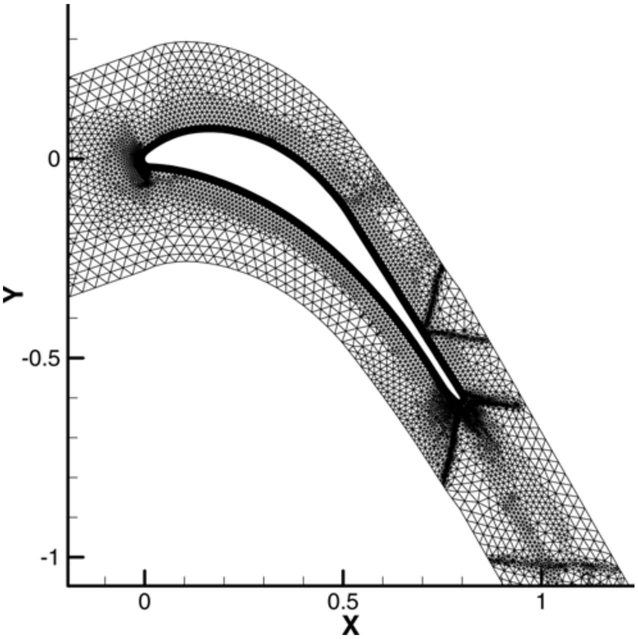

Fig. 11 Refined grid for N–S computation

Figure 11 shows the computational mesh after three rounds of refinement, two of them based on the shock sensor and one based on the wake sensor. The pressure distribution along the blade surface is plotted in Fig. 12. A comparison with Fig. 10 shows that the shock wave is much sharper after the grid refinement. Figure 13 plots the Mach contour based on the refined grid and provides an instant view of the wake of the flow.

The numerical results of the inviscid and viscous flow computations are listed in Table 1 and compared with experimental results [**12**]. Five numerical solutions from the present study are included: Euler (on a coarse grid), Euler (on a fine grid), Navier–Stokes (NS) (laminar), NS (BL turbulent model), and NS (k–ω turbulent model).

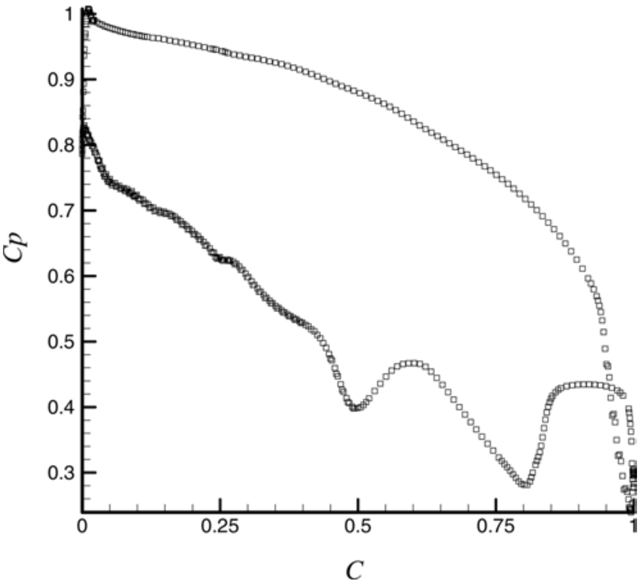

Fig. 12 Pressure distribution along the blade chord based on the refined grid, calculated by the present N–S solver with a k–ω model

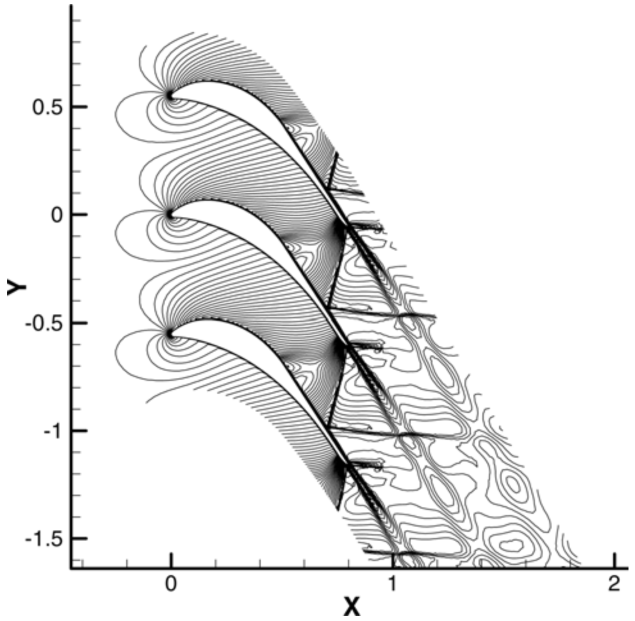

Fig. 13 Mach contour on the SE1050 blade cascade with a $k-\omega$ model based on the refined grid

The energy loss coefficient, ξ, is defined [12] as

$$\xi = 100\left(1 - \frac{\lambda_2^2}{\lambda_{2,\mathrm{is}}^2}\right)$$

where $\lambda_{2,\mathrm{is}}$ is the isentropic outlet relative speed and λ_2 is the real outlet relative speed. The ξ values calculated in different computations are listed in Table 1. As expected the viscous results give a higher energy loss than do the inviscid results. The loss coefficient predicted by the inviscid computation depends on the grid size: $\xi = 2.1$ per cent in the fine grid and $\xi = 2.6$ per cent in the coarse grid. When the grid size is small, the shocks are sharper and the numerical viscosity is lower. For the viscous computations, the $k-\omega$ model predicts the highest energy loss (which is also the closest to the experiment result), and ξ of the laminar flow is the smallest because of the absence of turbulence viscosity.

Table 1 Comparison of the experimental and various numerical results

	Inlet Mach number, M_1	Exit angle, β_2 (deg)	Energy loss coefficient, ξ (%)
Experimental [12]	0.375	29.80	4.6
Computational [12]	0.329	32.02	Not available
Euler (on a coarse grid)	0.329	31.94	2.5
Euler (on a fine grid)	0.329	31.91	2.1
NS (laminar)	0.328	31.80	2.6
NS (BL turbulent model)	0.327	31.95	2.9
NS ($k-\omega$ turbulent model)	0.328	31.93	3.1

5 CONCLUSION

Computational solvers of Euler and Navier–Stokes equations have been developed. A hybrid scheme is used for spatial discretization, where the inviscid fluxes are discretized using a finite volume method while the viscous fluxes are calculated by central differences. A MUSCL-type approach is used for achieving higher-order accuracy. The effects of the turbulent stress terms in the Reynolds-averaged Navier–Stokes equations have been studied with two different models: an algebraic turbulence model (Baldwin–Lomax model) and a two-equation turbulence model ($k-\omega$ model). The calculations have been carried out on unstructured grids. The initial grids for inviscid and viscous computations are generated by the advancing front method (AFM) and Delaunay–Veronoi method (DVM), while the self-adaptive procedure is based on the Delaunay–Veronoi method (DVM). A new rotation-based sensor has been devised to capture the wake region more accurately.

The numerical solvers have been applied to the transonic rotor turbine profile cascade SE1050. The numerical results of the inviscid and viscous flow computations are listed in Table 1 and compared with experimental results [12]. Five numerical solutions from the present study are included: Euler (on a coarse grid), Euler (on a fine grid), NS (laminar), NS (BL turbulent model), and NS ($k-\omega$ turbulent model). Comparisons with experiment include relative velocity distributions and Mach number contours. In general, the comparisons with experiment are good.

In the present work, a new method is developed such that the calculation of the flux, the boundary conditions, and the implicit Jacobian all conform to the Roe scheme. This provides internal consistency and improves the convergence significantly. Many numerical tests have been performed with different expressions for the inlet and outlet boundaries. The traditional Steger–Warming scheme and the new Roe scheme are tested, combined with different inviscid flux solvers: the Osher scheme and the Roe scheme. The present numerical study (e.g. Fig. 5) shows that the new treatment of the boundary conditions at the inlet and outlet boundaries improves the computational efficiency significantly.

Work is at hand to augment the Euler and Navier–Stokes solvers to simulate non-equilibrium wet steam flow through turbine cascades. References [13] and [14] present an Eulerian–Lagrangian approach for wet steam calculations with many novel features (e.g. accurate coupling of unsteady fluid dynamics, a new averaging procedure for retaining a poly-dispersed droplet spectrum etc.), whereas the two-phase solvers under development will use an Eulerian–Eulerian scheme.

ACKNOWLEDGEMENTS

The authors wish to express their gratitude to P. Šafarik who provided the experimental data and helpful comments. YM was supported by a University of Bristol Scholarship and an ORS award.

REFERENCES

1 **van Leer, B.** Towards the ultimate conservative scheme, monotony and conservation combined in a scheme II. *J. Comput. Phys.*, 1974, **14**, 361–370.

2 **Roe, P.** Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 1981, **43**, 357–372.

3 **Osher, S.** and **Salomon, F.** Upwind difference schemes for hyperbolic systems of conservation laws. *Math. Computation*, 1982, **28**, 339–374.

4 **Steger, J. L.** and **Warming, R. F.** Flux-vector splitting of the inviscid gas dynamic equations with applications to finite-difference methods. *J. Comput. Phys.*, 1981, **40**, 263–293.

5 **Baldwin, B.** and **Lomax, H.** Thin-layer approximation and algebraic model for separated turbulent flows. AIAA paper 78–257, 1978.

6 **Wilcox, D. C.** *Turbulence Modelling for CFD*, 1993 (DCW Industries, Canada, California), ISBN 0-9636051-0-0.

7 **van Leer, B.** Towards the ultimate conservative difference scheme V, a 2-order sequel to Godunov's method. *J. Comput. Phys.*, 1979, **32**, 101–136.

8 **Yao, J., Jameson, A., Alonso, J. J.,** and **Liu, F.** Development and validation of massively parallel flow solver for turbomachinery flows. AIAA paper 00-0882, 2000.

9 **Anderson, W.** and **Bonhaus, D.** An implicit upwind algorithm for computing turbulent flows on unstructured grids. *Computers Fluids*, 1994, **23**(1), 1–21.

10 **Štastny, M.** and **Šafarik, P.** Experimental analysis data on the transonic flow past a plane turbine cascade. ASME paper 90-GT-313, 1990.

11 **Denton, J. D.** and **Dawes, W. N.** Computational fluid dynamics for turbomachinery design. *Proc. Instn Mech. Engrs, Part C: J. Mechanical Engineering Science*, 1999, **213**, 107–124.

12 **Štastny, M.** and **Šafarik, P.** Boundary layer effects on the transonic flow in a straight turbine cascade. ASME paper 92-GT-155, 1992.

13 **Guha, A.** and **Young, J. B.** Time-marching prediction of unsteady condensation phenomena due to supercritical heat addition. In Proceedings of IMechE Conference on *Turbomachinery: Latest Developments in a Changing Scene*, London, 1991, pp. 167–177.

14 **Guha, A.** Two-phase flows with phase transition. VKI Lecture Series 1995–06 (ISSN 0377-8312), von Karman Institute for Fluid Dynamics, Belgium, 1995, 1–110.

APPENDIX

Notation

$\mathbf{A}$	Jacobian matrix of flow equations
c	chord length
C	non-dimensional co-ordinate along the chord
c_p	specific heat at constant pressure
C_p	pressure coefficient (defined as p/p_0)
e	total energy per unit volume
i	incident angle
k	turbulent kinetic energy
M	Mach number
$M_{2,\mathrm{is}}$	isentropic outlet Mach number
$\boldsymbol{n}$	outward unit vector normal to the cell interface
p	static pressure
p_0	total pressure
Pr	Prandtl number
q_i	components of the heat flux vector
$\dot{\boldsymbol{q}}^S$	cell interface heat flux
Re	Reynolds number
t	time
T	temperature
T_0	total temperature
$\boldsymbol{u}$	velocity vector
V	volume
$\boldsymbol{W}$	conservative variables vector
X	non-dimensional coordinate $= x/c$
Y	non-dimensional coordinate $= y/c$
α	inflow angle
β	exit angle
γ	ratio of specific heats
λ	relative velocity
μ	coefficient of viscosity
ξ	energy loss coefficient
ρ	density
$\boldsymbol{\tau}_{ij}$	components of the shear stress tensor
$\boldsymbol{\tau}^S$	cell interface shear stress vector
ω	rotation $= \sqrt{(\partial u/\partial y - \partial v/\partial x)^2}$

Subscripts

is	isentropic flow
L	left of interface
R	right of interface
1	inlet
2	outlet

Superscripts

n	time step