Multall—An Open Source, Computational Fluid Dynamics Based, Turbomachinery Design System

John D. Denton
Whittle Laboratory,
Cambridge University Engineering Department,
Cambridge CB3 ODY, UK
e-mail: jdd1@cam.ac.uk

Turbomachinery design systems are usually the jealously guarded property of large companies, and the author is not aware of any for which the source code is freely available. This paper is aimed providing a freely available system that can be used by individuals or small companies who do not have access to an in-house system. The design system is based on the three-dimensional (3D) computational fluid dynamics (CFD) solver Multall, which has been developed over many years. Multall can obtain solutions for individual blade rows or for multistage machines, and it can also perform quasi-3D (Q3D) blade-to-blade calculations on a prescribed stream surface and axisymmetric throughflow calculations. Multall is combined with a one-dimensional (1D) mean-line program, Meangen, which predicts the blading parameters on a mean stream surface and writes an input file for Stagen. Stagen is a blade geometry generation and manipulation program which generates and stacks the blading, combines it into stages, and writes an input file for Multall. The system can be used to design the main blade path of all types of turbomachines. Although it cannot design complex features such as shroud seals and individual cooling holes, these features can be modeled, and their effect on overall performance predicted. The system is intended to be as simple and easy to use as possible, and the solver is also very fast compared to most CFD codes. A great deal of user experience ensures that the overall performance is reasonably well predicted for a wide variety of machines. This paper describes the system in outline and gives an example of its use. The source codes are written in FORTRAN77 and are freely available for other users to try. [DOI: 10.1115/1.4037819]

Introduction

Turbomachinery design systems differ greatly in their details but the general approach is likely to follow the following guidelines:

(1) Specify overall parameters such as mass flow rate, mean diameter, rotational speed, inlet flow conditions, and exit pressure.
(2) Perform a one-dimensional (1D) mean-line calculation to obtain the annulus shape and midspan blade angles.
(3) Perform a two-dimensional axisymmetric throughflow calculation in the inverse (design) mode to obtain the variation of flow angles along the span.
(4) Repeat the throughflow calculation in analysis mode to predict the blade losses, machine efficiency, and stream surface thickness distributions.
(5) Perform quasi-3D (Q3D) blade-to-blade calculations at several spanwise sections on each blade row to design the blade shapes.
(6) Perform relatively coarse grid, multistage, three-dimensional (3D), viscous calculations for the main flow path of the whole machine to optimize the blade stacking.
(7) Perform more detailed 3D calculations to include the effects of leakage flows, endwall bleeds and cavities, and coolant flows. These will give the final prediction of machine overall performance.

Each of these stages is likely to be repeated several times with return to repeat previous steps often being necessary.

The author has long maintained [1] that steps 3–5 should be omitted and that, given a flexible geometry manipulation program and a fast 3D solver; it is not only simpler, but also faster, to go straight from the mean-line calculation to coarse grid 3D calculations. These can then be used to optimize the blade shapes and stacking at the same time. The main time saving comes from omitting the numerous Q3D blade-to-blade calculations, which are anyhow of limited accuracy, because the change in blade loading produced by redesigning a section on a Q3D stream surface will not be the same as that obtained when the same blade section is calculated as part of a 3D solution. This is because of the well-known phenomenon of three-dimensional relief [2], spanwise changes in blade loading must be gradual in the real 3D flow, while there can be large changes between adjacent sections on Q3D surfaces. Figure 1 illustrates this for a highly loaded low-

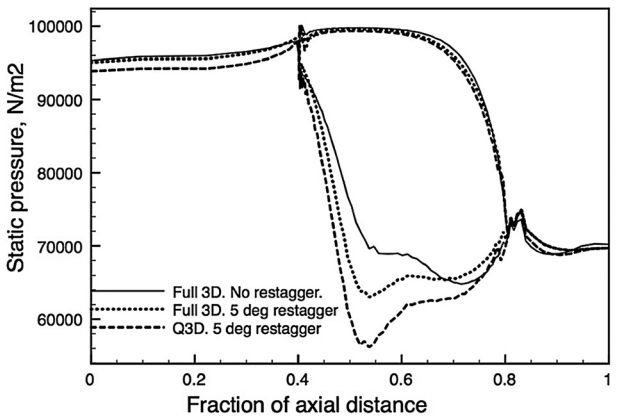

Fig. 1 Effects of local restagger on an LP turbine blade

Contributed by the International Gas Turbine Institute (IGTI) of ASME for publication in the JOURNAL OF TURBOMACHINERY. Manuscript received July 11, 2017; final manuscript received August 24, 2017; published online September 26, 2017. Editor: Kenneth Hall.

pressure (LP) turbine blade, the midspan section has been restaggered by 5 deg from its design setting, and its predicted loading distribution from a Q3D and a 3D calculation are compared. The trend of the change in loading is the same in the Q3D and 3D calculations, but the magnitude is much less for the 3D calculation.

To satisfy these requirements, the author has, over many years, developed a system based on the 3D solver Multall and simple geometry generation and manipulations programs, Meangen and Stagen. This system is described in this paper. The programs are all written in FORTRAN77 and the source codes, user manuals and sample test case data sets are freely available and can be downloaded as described in the Appendix. The only known freely available design system is that described by Turner et al. [3]. It is a quasi-3D system, based around the blade-to-blade solver Mises, and only the executable codes are available.

Historical Background

The heart of the system is the 3D program Multall. This is a multistage, 3D, steady flow, program for the main blade path, based on the finite volume time marching method. Multall assumes smooth axisymmetric annulus boundaries and so does not include hub or shroud cavities or detailed blade tip or shroud geometries. However, tip and shroud leakages can be modeled. The interfaces between blade rows are treated by mixing planes.

Multall is the latest version of a CFD code that has been continuously developed for over 40 years [4]. Its original version was probably the first 3D turbomachinery CFD code to be used by industry. At that stage it was inviscid, for a single blade row, with extremely coarse grids and was based on the "opposed difference" numerical scheme. The first mixing plane model was added in 1979 [5] to permit complete stage calculations. The details of the mixing plane model have undergone many improvements over the years, and the latest version is described later in this paper. A particularly simple form of multigrid was added in the early 1980s and gave a significant increase in convergence rate. Viscous effects, with gradually increasing levels of complexity, were added throughout the 1980s, as were numerous other refinements. The most enduring viscous model is a thin shear level approximation to the Navier–Stokes (N–S) equations, based on the mixing length model, which, despite the availability of more complex models, continues to give excellent results. The resulting code was widely used in the industry, especially for multistage calculations.

The solution algorithm was changed to a new method, the "scree" scheme around 2000. This is simpler, faster, and requires lower levels of artificial viscosity than other schemes with which the author is familiar. It also works better than other methods at low Mach numbers and is the basis of the present code. A full N–S version of the mixing length model and the Spalart–Allmaras turbulence model were added between 2000 and 2010, as was an improved mixing plane method. In the last few years, the code has been extended to allow Q3D blade-to-blade calculations and axisymmetric throughflow calculations to be performed within the same code. These additions, together with the speed and simplicity of the code make it particularly suitable for use as a design tool.

Although it has been, and still is, quite widely used, many of the details of Multall have not been published in the open literature and some of the more interesting details are given in this paper. More details can be found in the user manual.

The origins of the blade section generating and stacking program, Stagen, go back to 1994 and are described in Ref. [1]. It has also undergone continuous development to keep it compatible with Multall and to increase the options for generating and stacking the blade profiles. Although it is very easy to modify the geometry once a data set for Stagen has been set up, setting up of the initial data set can take some time, of order half an hour per blade row. Hence, a preprocessor called Meangen has been developed. This has been written to use the minimum possible input data, of order 20 lines of data per blade row, and an initial data set can be generated by answering questions on the screen. The data set for Meangen can therefore be generated in a matter of minutes. Meangen then performs a mean-line design calculation, twists the blades to generate a free vortex flow, and writes a complete data set for Stagen. Stagen will run on this data set without the need to make any changes to it and writes a data set for Multall. Using the combination an initial 3D data set for Multall can be generated in a matter of minutes, and an initial 3D solution obtained in the order of 15 min per stage.

Meangen

Meangen is designed to use the minimum possible input data and so many of the parameters needed, such as grid point numbers, gas properties, blade thicknesses, blade row, and stage spacings are set by default. When starting a new design, these defaults should be changed to fit requirements and the program recompiled. The number of blades is fixed by default values of the Zweifel coefficient, modified to allow for changes in radius and stream surface thickness. Meangen can obtain its data from either the screen or from a file. When first run on a new design, the user will usually answer questions, such as "Input the required mass flow rate," presented to them on the screen. The data input is saved in a file, so that the next time Meangen is run on the same design small changes can most easily be made by editing the file.

Meangen is really designed to design complete stages and multiple stages, although data sets for single bade rows can be produced if requested. There are two options for choosing the mean stream surface. For an axial flow machine with small radius changes, it is possible to specify a fixed design radius for each stage; this can be either the hub, mid, or tip radius and can vary from stage to stage. For a machine with significant radius change, such as a mixed flow or centrifugal compressor or a radial inflow turbine, the coordinates of a design stream surface must be specified and again this may be the hub, mid, or tip stream surface.

For the first option, the velocity triangles at the design radius can be chosen using the conventional equations for a repeating stage with constant axial velocity. To do this, it is necessary to specify any three variables chosen from

- stage flow coefficient, ϕ;
- stage loading coefficient, ψ;
- stage reaction, λ;
- stator inlet angle, α_1;
- stator exit angle, α_2;
- rotor relative inlet angle, β_1; and
- rotor relative exit angle, β_2.

Having chosen three of these values, all other parameters can be obtained from standard velocity triangle equations. The author finds the equation

$$\psi = 2(1 - \lambda - \varphi \tan \alpha_0) \tag{1}$$

particularly useful as it enables the flow angles to be determined from the basic design parameters, ψ, ϕ, and λ. It is valid for both turbines and compressors. It should be emphasized that these equations can only be used for a repeating stage with constant axial velocity.

When there is a significant radius change through the stage, the design stream surface coordinates must be input, and the velocity triangles are specified either by inputting all four flow angles, α_1, α_2, β_1, β_2, or by providing the absolute flow angles, α_0, α_4, at stage inlet and exit, together with the flow coefficient and the stage loading coefficient. The variation of meridional velocity through the stage is also specified. This type of input may be used to generate data sets for centrifugal compressors or radial inflow turbines. However, if multiple stages are generated in this way, there is no guarantee of repeating velocity triangles.

When designing a multistage machine, if a stage has the same design radius and velocity triangles as the previous stage, it can be designed simply by pressing "Y," otherwise the design radius and velocity triangles must be specified for each stage. Although the velocity triangles are only set at one radius, their variation along

the blade span is calculated assuming free vortex flow, i.e., $rV_\theta =$ constant, this should produce a design with fairly uniform meridional velocity.

Stagen

Meangen writes a data file, called "stagen.dat" that can be immediately read by Stagen. The file contains enough data to generate and stack preliminary blade sections, and Stagen can be run without any editing of the file to produce a preliminary 3D data set for Multall. However, this initial data set is based upon simple estimates of many parameters, and so, inspecting the 3D solution will very likely show that some details need to be improved, and this is done by editing the input file to Stagen.

The blade profiles are input on a set of axisymmetric streamwise surfaces, the first of which must be the hub and the last the casing of the machine. Typically 5 surfaces would be sufficient to specify a twisted blade. These surfaces need not be true streamlines, they are automatically generated by Meangen but can be changed within Stagen if it is required to change the annulus lines. The blade sections can be generated in a variety of ways.

(1) A pre-existing blade shape may be input by reading in its coordinates.
(2) The blade shape is input as a table of camber line angle and blade thickness against fraction of meridional chord.

If the second option is chosen, then typically 5–10 points are used and the tangent of the angle is taken to vary linearly between the input points. This produces a parabolic curve between each pair of input points. The blade thickness can be generated either by a series of equations or by inputting a table of the thicknesses above and below the camber line, against fraction of chord. The former method is more convenient and three numbers give a great deal of control over the blade shape. The range of thickness profiles that can be generated by the system of equations is illustrated in Fig. 2. The thickness can be added either perpendicular to the camber line or as a tangential thickness.

The general principle used when designing a blade section is to increase the local surface curvature to increase the local surface velocity and vice-versa. The curvature can be changed either by changing the camber line angles or by changing the thickness distribution. The method of inputting the blade thickness as data gives more control over this, simply increasing the local thickness above or below the camber line increases the curvature and hence the local surface velocity. The leading and trailing edge thicknesses are input separately to the main blade thickness and are blended to the main blade thickness at a specified fraction of chord. As shown in Fig. 2, the leading edge thickness distribution does not cause a discontinuity in the surface curvature.

The blade profiles are generated on a plane (x, y) surface but are then transformed into the (m, θ) coordinate system of the streamwise surfaces such that the x value becomes the meridional distance, m, and the y value defines the value of θ on the streamwise surface using

$$\theta - \theta_1 = \int_{r1}^{r} dy/r \tag{2}$$

This ensures that a flat plate on the (x, y) surface transforms into a log spiral on the (m, θ) surface, and an unloaded flat plate remains unloaded.

The blade sections on the streamwise surfaces are initially stacked on their centroids but can then be moved either tangentially or axially or along or perpendicular to their chord line to vary the stacking. Figure 3 shows the sections generated for a LP steam turbine rotor with converging–diverging sections for very high Mach numbers.

The number and spacing of grid points on the blade and upstream and downstream of it is controlled by input data to Stagen, as is the number and spacing of the pitchwise and spanwise grid points. However, these can easily be changed at a later stage within Multall. In a multistage machine, the grids generated by Stagen may overlap either side of the mixing plane, and the matching of the grids at the mixing plane is then done automatically within Multall.

Inlet and outlet boundary conditions for the flow are provided by Meangen. These initially assume spanwise uniform inlet

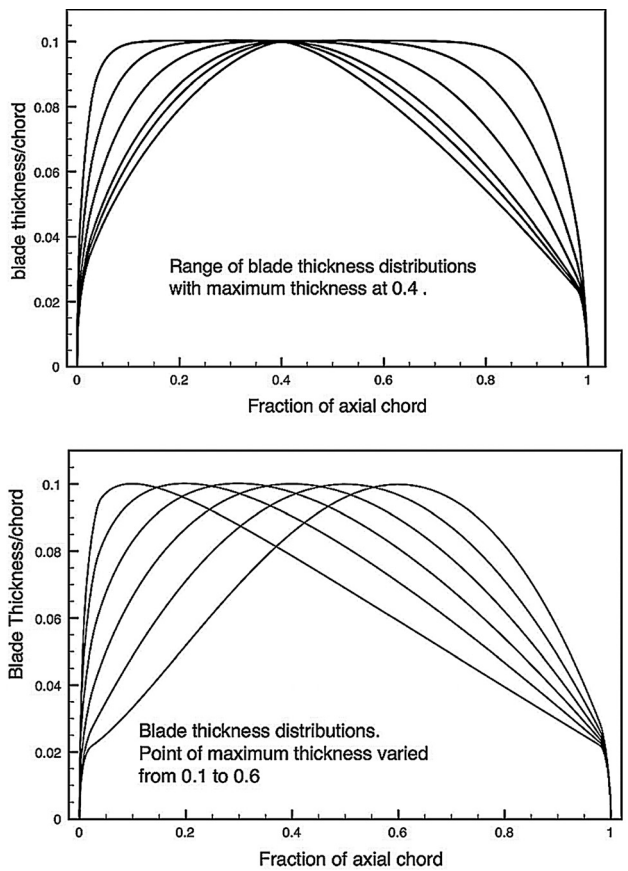

Fig. 2 Blade thickness distributions obtained from Stagen

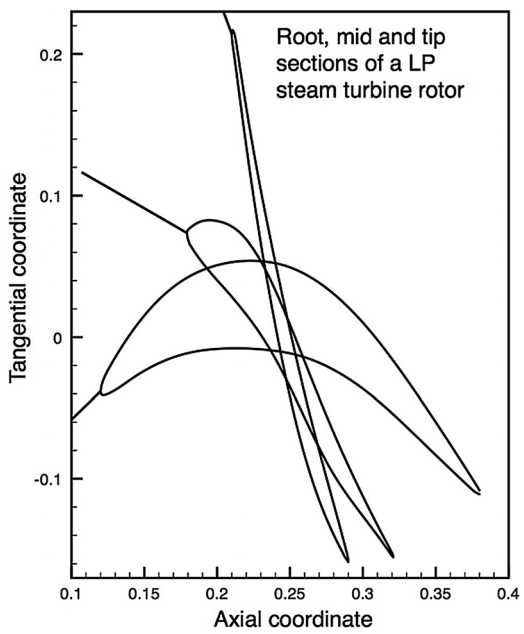

Fig. 3 Root, mean and tip blade sections for a last-stage steam turbine rotor

conditions, but they can easily be changed by editing the Stagen data file.

Multall

As mentioned previously, Multall has evolved over many years, and the main objectives throughout its development have been speed and simplicity. The code is written specifically for turbomachinery using a cylindrical x, r, θ coordinate system. The inner and outer boundaries must be surfaces of revolution, and the circumferential extent is the local blade pitch. The program is not multiblock, and the whole flow field is covered by a single grid, with special treatment at the mixing planes. This simplifies the coding and speeds up the calculation. Some of its details have been published previously [6] when the program was called "Multip," but details of many of the recent additions have not been published.

Features included in Multall are

- mixing plane model between blade rows,
- automatic matching of the grids at the mixing plane,
- quasi-3D blade-to-blade calculations,
- axisymmetric throughflow calculations,
- ability to refine the grids within the code,
- three turbulence models,
- specified boundary layer transition point, or a simple transition model,
- surface roughness effects,
- cooling flow addition through the blades or endwalls,
- temperature-dependent gas properties,
- bleed flows from the hub or casing,
- artificial compressibility for low Mach number and incompressible flows,
- pinched tip model for plain tip clearances,
- shroud leakage model for shrouded blades,
- automatic trailing edge cusp generation,
- modification of the inlet boundary conditions to simulate a repeating stage, and
- ability to perform limited blade redesign within the 3D code.

Space does not allow description of all these options and only some of the most interesting of them will be described in this paper. More details of all options can be obtained from the user manual.

Solution Algorithm. The code is based on an explicit time-marching finite volume method for solving the N–S equations. Various solution algorithms have been tried in the past, but the present one is considered to be by far the best. It is a single step scheme which works by storing the time derivatives from each time-step and using them as part of the next time-step. The conservation equations for mass, momentum, and energy (and turbulence quantities if used) integrated over a time-step Δt for a finite volume cell can be written as

$$\Delta F = \frac{\Delta t}{\text{vol}} \sum_{\text{cellfaces}} \text{fluxes} \tag{3}$$

where F is any one of the conserved variables, vol is the volume of a cell, and "flux" means the flow of mass, momentum or energy through a cell face.

The value of ΔF is evaluated for every conserved variable on every time-step. If the calculated value of ΔF was simply added to the value of F at the start of the time-step, the scheme would be completely unstable. However, if the value of dF/dt from the previous $(n-1)$ time-step is combined with the current value (time step n) to extrapolate the value of dF/dt to the end of the step, and we use the extrapolated value to update the variables, we get

$$F_{n+1} = F_n + \Delta F, \text{ where } \Delta F = \left[2\left(\frac{dF}{dt}\right)_n - \left(\frac{dF}{dt}\right)_{n-1}\right]\Delta t \tag{4}$$

which is stable, and very robust, for Courant–Friedrichs–Levy numbers up to about 0.5. The scheme requires only a single evaluation of the fluxes and updating of the variables per time-step. Its efficiency is probably slightly less than that of a four step Runge–Kutta algorithm, which requires four flux evaluations and variable updates but permits about five times the CFL number, but in the author's view its simplicity more than makes up for that. It has been called the "scree" scheme because of the two steps forward, one step backward, which will be familiar to any mountaineer who has climbed a scree slope. In addition to its speed and simplicity, the scheme requires very little artificial viscosity to stabilize it, and it works at lower Mach numbers, down to about 0.15, than most other schemes.

A simple extension to the scheme, requiring only two extra lines of coding, is available as an option in the program. This is called the "SSS" scheme, and it permits larger CFL numbers, up to about 0.8, but is less robust than the basic "scree" scheme and so is not usually used.

Grid and Multigrid. The code uses a simple H grid with cell corner storage of the flow properties. Use of an H grid is much maligned in the CFD literature, because it inevitably leads to highly sheared finite volume cells. However, the author's experience is that, with cell corner storage, it works remarkably well, even with highly sheared cells. The numerical errors always vary as the square of the grid spacing and the number of cells affected by them varies inversely as the grid spacing, so smaller grid spacings always reduce the cumulative error. Hence, the loss of accuracy can be compensated by using more grid points in highly sheared region, and the speed that comes from simplicity makes this option competitive with more complex grids. An H mesh is anyhow necessary at the mixing planes. To illustrate this point, Fig. 4 shows the grid and solution for the flow around the leading edge of a turbine cascade. The grid is extremely sheared but despite this the stagnation point is well captured. Similar results, including a comparison to experiment on the leading edge of an aerofoil, are shown by Bryanston-Cross and Denton [7].

The shock capturing ability of the code may be seen in Fig. 5 which shows the solution for the staggered wedge test case from Ref. [8] which has a 60 deg sheared grid. The code contains an option to prevent overshoots and undershoots at shock waves, which was used in this case. The solution agrees very well with the exact solution given in Ref. [8]. The inlet Mach number is 1.6, and the bow shock is reflected off the lower surface and is exactly canceled when its reflection meets the corner on the upper surface. The Mach numbers after each reflection are almost exactly correct.

The multigrid method differs from the conventional one [9], being much simpler, but it seems to be equally effective for codes with cell corner storage. However, Peterson [10] was unable to get the method to work on his code with cell center storage. The multigrid blocks are formed by combining a group of cells into a block as illustrated in Fig. 6. The block need not be a simple $2 \times 2 \times 2$ block, as is usual, in fact a $3 \times 3 \times 3$ block is often optimum but larger blocks are perfectly acceptable. The sum of the fluxes around the faces of the block is found simply by summing the flux sums for the cells within it, which have already been obtained to calculate the primary changes, i.e.,

$$\sum_{\text{block}} \text{fluxes} = \sum_{\text{block}} \sum_{\text{cell}} \text{fluxes} \tag{5}$$

This is because all the fluxes across the internal cell faces cancel out leaving only those around the edges of the block. Each cell within the block is then given an update

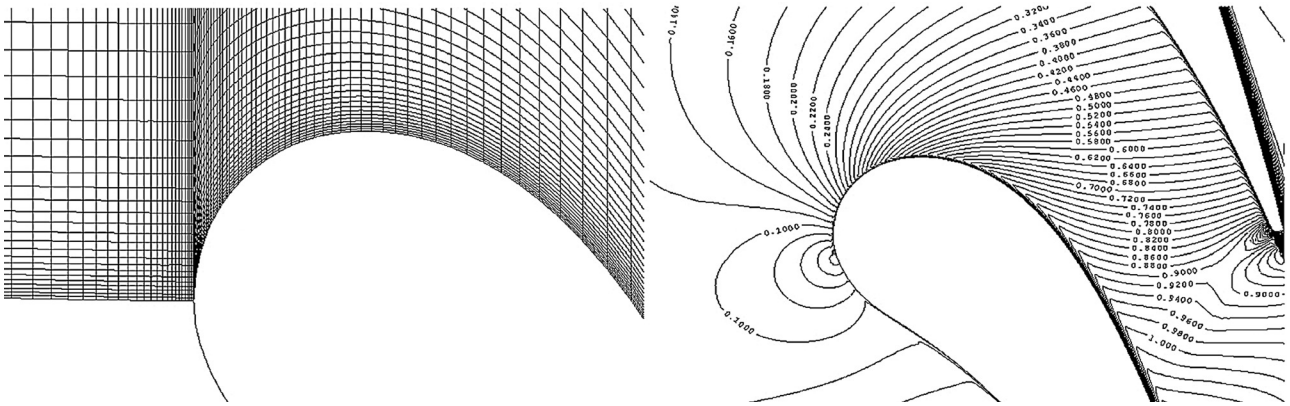

Fig. 4 Prediction of the stagnation point on a turbine blade

$$\Delta F = \frac{\Delta t}{\text{vol}_{\text{cell}}} \sum_{\text{cell}} \text{fluxes} + \frac{N\Delta t}{\text{vol}_{\text{block}}} \sum_{\text{block}} \text{fluxes} \tag{6}$$

where N is the number of cells per face of the block. For a three cell-sided block N can in theory = 3 but in practice it is reduced to about half of that value as a safety factor. The changes ΔF in every cell within the block are then distributed to the eight corners of the cell with double weighting at the solid boundaries. The extra computer effort involved in summing the flux sums is almost negligible.

Exactly the same procedure can be applied to higher levels of multigrid, the flux sums for the next lower level of multigrid, which have already been obtained, can be summed in the same way as Eq. (5) to form the changes for the higher level. For the second level, a block size of $9 \times 9 \times 9$ is usually optimal, combined with the first level this makes the variables theoretically change at 13 times the rate of a single grid, again this is reduced by a safely factor. Figure 6 illustrates a two-dimensional 9×9 block composed from $9 \times 3 \times 3$ blocks.

A final extremely coarse level of multigrid is then added. This forms blocks which cover the whole span and whole pitch with one block upstream of the leading edge, one downstream of the trailing edge and two in the blade passage, i.e., only four blocks per blade row. This "super block" gives very rapid transmission of information from inlet to outlet of the calculation and is very beneficial for multistage calculations.

Mixing Plane Model. Development of a good mixing plane model is one of the most difficult problems in CFD. The mixing plane is an artificial concept designed to permit steady calculations of blade rows which are in relative motion due their rotation. It is generally assumed that the mixing plane should allow the flow from an upstream blade row to mix out as if it were doing so in a long duct with constant area, and it should allow the flow to enter the downstream row as if it had originated from a pitchwise uniform flow far upstream. Hence, the mixing plane must transmit the mixed out fluxes of mass, momentum, and energy from one blade row to the next, while causing the minimum distortion to the pitchwise nonuniform flows leaving the upstream row and entering the downstream row.

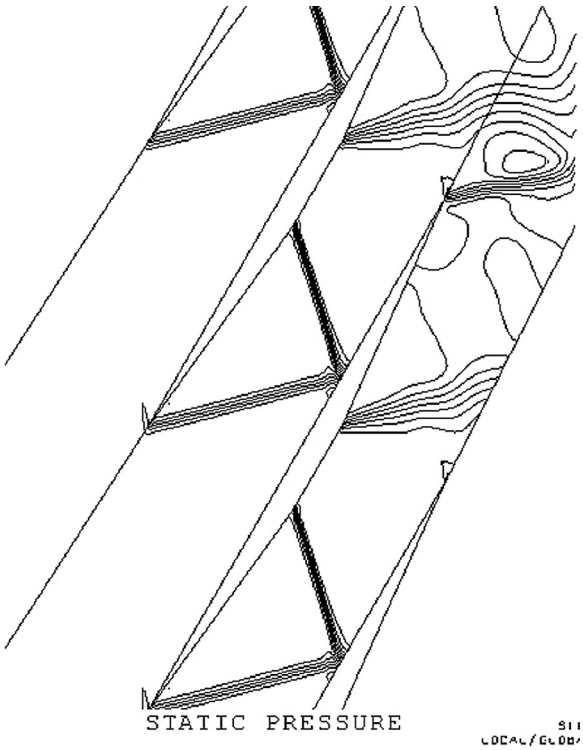

Fig. 5 Solution for the staggered wedge test case

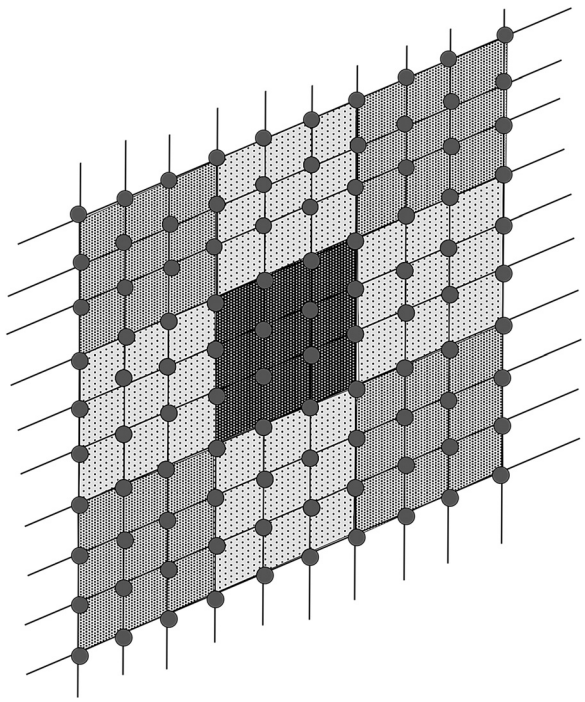

Fig. 6 A 9×9 multigrid block composed of $9 \times 3 \times 3$ blocks

The first mixing plane model was applied to early versions of the code in the late 1970s [5]. This consisted of simply pitchwise averaging the flow at the mixing plane, and this was found to give erroneous results when the mixing plane was close to the blade row leading or trailing edges. Gradual improvements were introduced resulting in the "flux extrapolation" model [6]. An improved version of this is used in the present code. The approach is quite different to, and much simpler than, methods based on the method of characteristics such as that described by Holmes [11]. The method is illustrated schematically in Fig. 7.

At every spanwise location there are two coincident pitchwise gridlines numbered JMIX and JMIX + 1, the flow on both these lines is pitchwise uniform. Information is transmitted between them using one-dimensional time marching to update the pitchwise uniform variables. This ensures that when the solution is converged the flow becomes identical on the two coincident lines, and this is the mixed out flow.

The treatment on the upstream face is the same flux extrapolation method described in Ref. [6]. It is summarized by the following equation:

$$\mathrm{FLUX}_{\mathrm{jmix}} = \mathrm{FLUX}_{\mathrm{avg,jmix}} + \mathrm{F}_{\mathrm{ext}}(\mathrm{FLUX}_{\mathrm{jmix-1}} - \mathrm{FLUX}_{\mathrm{avg,jmix-1}}) \tag{7}$$

Because the average value of the extrapolated flux is zero, this does not change the total flux across the mixing plane. However, it means that the cells adjacent to the mixing plane see only a gradual decay of any pitchwise nonuniformity rather than a sudden removal of the nonuniformity, which would cause an upstream disturbance to the flow. The value of F_{ext} is typically in the range 0.8–0.9.

The flow immediately downstream of the mixing plane should have pitchwise uniform relative stagnation enthalpy and entropy but the flow direction and static pressure must not be pitchwise uniform. Flux extrapolation was found to be of little benefit and slightly destabilizing on the downstream side of the mixing plane and so is not used. Instead the mixed out entropy and relative stagnation temperature from the mixing plane, together with the calculated static pressure is used to calculate an isentropic velocity and density at JMIX + 2. The flow direction at JMIX + 2 is extrapolated from downstream and combined with the isentropic velocity to give the velocity components, changes in these are heavily relaxed. This ensures that the final solution has pitchwise uniform entropy and relative stagnation enthalpy entering the downstream row. This model works very well for subsonic flow, but for supersonic relative flow at the mixing plane extrapolation of the flow angle on the downstream side can give erroneous results. This is because changes in flow direction must be related to changes in Mach number via the Prandtl–Meyer relationship. Whenever the relative downstream flow is supersonic, this relationship is used to obtain the variation of flow angle away from the midpitch value. This allows pressure waves to intersect the mixing plane without reflection as illustrated in Fig. 8.

A further important test of a mixing plane model is its ability to mix out a potential flow without any mixing loss and without any distortion of the upstream and downstream flow. Demonstration of this requirement has been largely neglected by the previous methods. Figure 9 shows a good test case for this. The two disks are separated by 20% of their axial chord, the left-hand disk is stationary, and the right-hand one is rotating at such a speed that it is at zero incidence. This produces a highly nonuniform potential flow at the mixing plane, the flow should be symmetrical and should have no mixing loss. Figure 9 shows that symmetry is not quite achieved at the mixing plane but is very good elsewhere. The mixing loss was not quite zero, producing a stagnation pressure loss coefficient of 0.2%.

Viscous Modeling. Three turbulence models are available in the code. The original model was a thin shear layer approximation to the N–S equations using a mixing length model to obtain turbulent viscosity. This is retained in the present version as it has proved remarkably reliable and robust over many years. An updated mixing length model with full N–S terms is also available as is a version of the Spalart–Allmaras model. In all three models, the viscous terms in the N–S equations are turned into source terms, which are only updated every 5–10 time-steps. This gives a significant saving in run times.

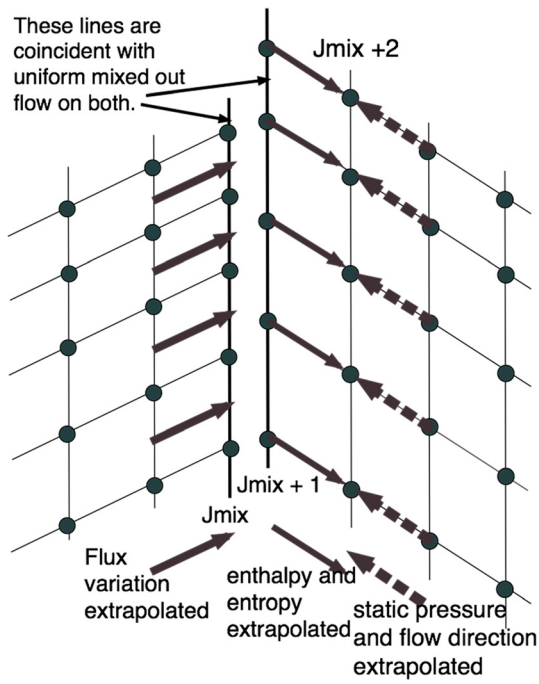

Fig. 7 Mixing plane treatment

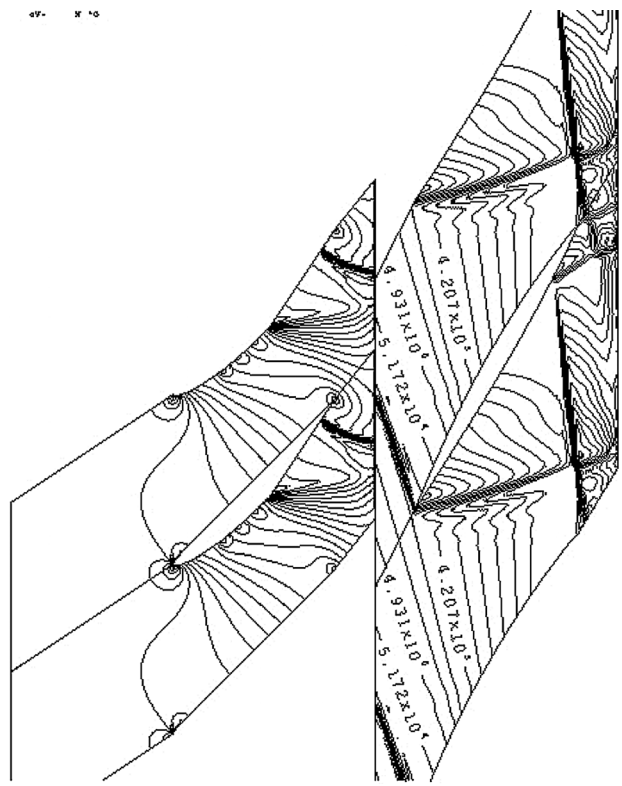

Fig. 8 Mixing plane with shock and expansion waves

Surface shear stresses are evaluated using wall functions. The velocity V_2 at the second grid point from the wall is turned into a Reynolds number $\mathrm{Re}_2 = V_2 y_2/\nu$. This is then used to predict the wall shear stress, τ_w, using a curve fit to results obtained from the standard log–law for turbulent boundary layers

$$\frac{2\,\tau_w}{\rho_2\,{V_2}^2} = -0.00178493 + \frac{0.029072}{\ln(\mathrm{Re}_2)} + \frac{0.270313}{(\ln(\mathrm{Re}_2))^2} \tag{8}$$

This predicts the shear stress from the log–law within 1% over the range Yplus = 10–1000. This enables a large saving in grid size and run time compared to methods which require values of Yplus of order unity. The wall stresses are gradually blended to the laminar values if the value of Yplus is less than 25. The author's experience is that obtaining the correct wall shear stress is the most important part of any turbulence model. The effects of surface roughness are predicted using the model described by White [12], this is done by modifying the wall functions.

Boundary layer transition can be imposed at a specified location on all surfaces or it can be predicted by the simple method described by Baldwin and Lomax [13] which says that the flow remains laminar if the ratio of the maximum calculated turbulent viscosity to the laminar viscosity is less than an input value, typically 14.

Negative Feedback. This is a very simple but effective means of increasing the robustness of a code. It could be applied to most explicit CFD codes. The idea is to limit the rates of change of the flow properties at grid points, where the calculated rates of change are much higher than average.

After summing the fluxes and multiplying by the local time step for all cells, the resulting rates of change, Δ_{calc}, are obtained. The absolute magnitudes of the rates of change are then averaged to find, Δ_{avg}, for the whole flow field. The rates of change of all cells are then changed by

$$\Delta_{\mathrm{used}} = \frac{\Delta_{\mathrm{calc}}}{(1+D)}\,, \quad \text{where } D = \frac{\mathrm{ABS}(\Delta_{\mathrm{calc}})}{\mathrm{DAMP}\,\Delta_{\mathrm{avg}}} \tag{9}$$

DAMP is an input variable which controls the amount of negative feedback. Cells where the calculated change is much less than (DAMP $\times$ Δ_{avg}) will scarcely be changed but cells where the calculated change is comparable to or greater than (DAMP $\times$ Δ_{avg}) will have their changes reduced. This acts as a powerful stabilizing influence on cells which might be unstable due to locally very high Mach numbers. Such local instabilities often occur during an initial transient when starting from a crude initial guess and would otherwise require the whole calculation to be run with a lower CFL number and/or higher smoothing. Typical values of DAMP are in the range 10–25.

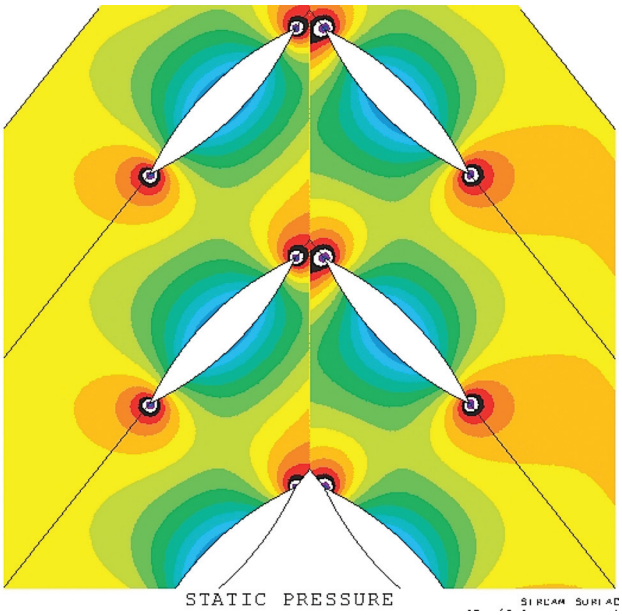

Fig. 9 Static pressure for potential flow across a mixing plane

The Quasi-3D Blade-to-Blade Model. The code can be run with only a single cell between two stream surfaces. With cell corner storage, it is necessary to have two spanwise grid points, one on each stream surface, as illustrated in Fig. 10. The coordinates of only one stream surface are input, and the perpendicular spacing of the stream surfaces, i.e., the stream surface thickness, is input as data. It is trivial to specify no flow through the stream surfaces but this is not sufficient to ensure that the velocity vectors follow the mean surface. It is usual to apply a body force acting perpendicular to the flow to ensure this. However, with two grid points, it is possible to apply the force via a pressure difference between the two surfaces. The midpoint of the two surfaces is regarded as a boundary between two half cells. Any flow crossing this boundary will cause an increase in pressure in one half cell and an equal decrease in the other half cell. This is in addition to the change in pressure calculated by the normal solution procedure, which is the same on both surfaces.

The change in pressure is calculated in a time-marching fashion using

$$\Delta P = c^2\,\Delta\rho, \quad \text{where } \Delta\rho = \frac{(\text{mass flux})\,\Delta t}{\text{cell volume}} \tag{10}$$

The mass flux is the flow crossing the midsurface, and c is a rough estimate of the local speed of sound. This gradually builds up a pressure difference that drives the flow crossing the midsurface to zero. Since the change in pressure is equal and opposite on the two stream surfaces, this does not affect the streamwise component of the pressure force applied on the flow by the surfaces.

Viscous forces, obtained from the usual wall functions are retained on the blade surfaces but not on the stream surfaces so that the model gives a prediction of the blade profile loss. Figure 1 gives an example of the use of the method. Run times for a single blade row are the order of 15 s on a single processor.

The Throughflow Model. Several time-marching throughflow models have been developed in the past, starting with that of Spurr [14], which was based on an early version of the present code. However, they have never proved as popular as streamline curvature methods, largely because of the much longer run times. However, they have a number of advantages as will be shown here.

The present method is based on the same idea as the blade-to-blade model described above and is coded to use a standard Multall data set, including the full blade geometry, although the latter can be much coarser than that used for a 3D solution. As with the Q3D method, a mean stream surface is defined at the midpoint of

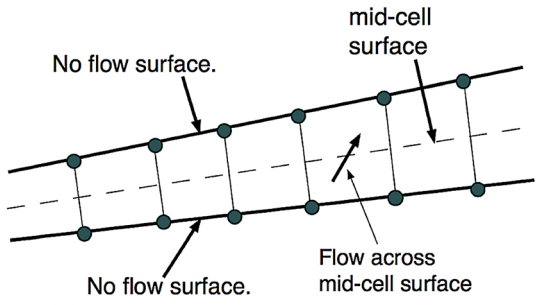

Fig. 10 The cells used for a Q3D calculation

the blade-to-blade gap as illustrated in Fig. 11. Any flow crossing this surface causes an increase in pressure on one blade surface and an equal decrease on the adjacent surface. This builds up an approximate blade loading which is updated every time-step until the flow follows the mean surface. The blade loading automatically acts perpendicular to the mean surface and so does not generate loss. The loading distribution is only a crude approximation to the true 3D loading, but its overall magnitude will be compatible with the flow turning imposed by the blade and the momentum change of the fluid. This, coupled with the use of the full 3D blade geometry makes the prediction of the effects of blade lean and sweep more realistic than that from a conventional throughflow method.

As with any throughflow method, it is necessary to allow for any flow deviation empirically, and this is done by inputting either the deviation angle or the flow exit angle as data. This causes a complication, because the blade force no longer acts perpendicular to the flow, and so, to prevent spurious loss generation, it has to be resolved perpendicular to the imposed flow.

Most throughflow programs also input the loss coefficients empirically, and this is usually done by imposing a streamwise body force; however, in the present code, it is more convenient to maintain the shear stresses on the blade and endwall surfaces and so allow the loss to be generated automatically. Clearly, the wall functions are not valid when there are only two grid points across the pitch, but one of the wall function models available in the code works by specifying the value of Yplus at the grid point on the wall. When this is done, with a uniform velocity across the pitch, it generates a skin friction coefficient of $C_f = 2/(\text{Yplus})^2$. Inputting a value of Yplus = 20 gives a very typical value of $C_f = 0.005$, which in turn produces realistic losses. The loss can be tuned by adjusting the value of Yplus in the input data.

The flow along every stream tube is effectively one-dimensional with the local flow area being that measured perpendicular to the mean stream surface. This means that choking occurs as in 1D nozzle, the flow will choke at the point of minimum area and only normal shock waves can be predicted. Such programs cannot predict the oblique shock waves that are common in turbomachines. This limitation is common to all time-marching throughflow methods which specify the flow direction. If the tangential velocity rather than the flow direction is specified, as in the inverse mode, then choking will only occur when the meridional Mach number exceeds unity. These limitations are discussed in detail by Sturmayr and Hirsch [15]. It should be noted that conventional streamline curvature throughflow methods cannot predict any type of shock wave.

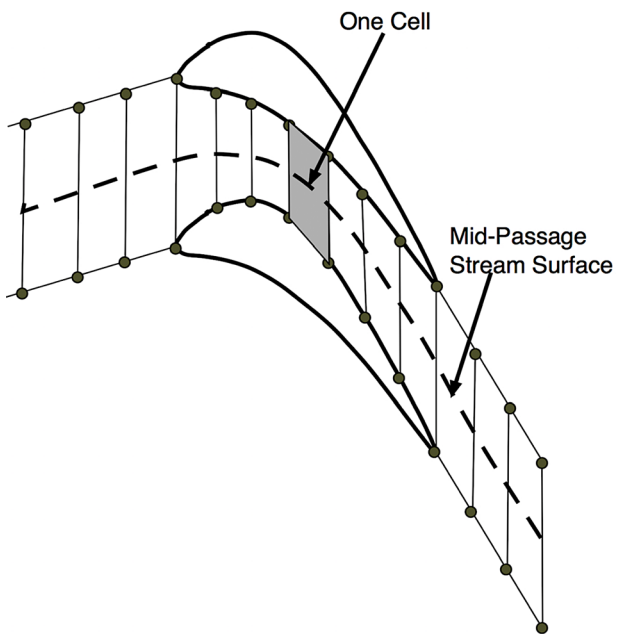

Fig. 11 Cells used for a throughflow calculation

Figure 12(*a*) compares streamlines from a full 3D, fine grid, calculation, and from a coarse grid throughflow calculation on an LP steam turbine with very high Mach numbers. The streamlines are very similar, particularly the important streamline curvature in the stator–rotor gaps. However, a better test is Fig. 12(*b*), which shows a good agreement between the two calculations for the Mach number distributions through the last stage. Figure 12(*c*) shows the midspan surface pressure distribution predicted by the throughflow calculation. Although the distributions are smoother than those from the 3D calculation (not shown), the variations are not unrealistic.

Compared to a conventional streamline curvature throughflow method the present method has the following advantages:

- It gives a crude estimate of blade loading.
- It predicts the 3D effects of blade stacking more accurately.
- The stability is not limited by the grid aspect ratio as it is for streamline curvature methods.
- It predicts viscous losses on the blade and endwall surfaces via a very simple skin friction model.
- It predicts tip leakage flows and losses.
- It predicts the growth of endwall boundary layers, but not the associated secondary flows.
- It works with a specified exit pressure rather than a specified mass flow. This is essential for choked blade rows.
- It predicts supersonic deviation from choked blade rows.
- It predicts normal shock waves but not oblique shocks.
- It is very easy to change from a throughflow to a full 3D calculation.

And the following disadvantages:

- Run times are about 15 s per stage rather than a fraction of a second with streamline curvature.
- It requires a 3D blade shape, although this can initially be a simple guess.
- It is more difficult to use empirical correlations for loss, although they could be included.
- It does not give realistic blade surface pressure distributions for transonic compressors, but nor does any throughflow method.
- Users are not so familiar with the method.

Trailing Edge Modeling. Most CFD methods suffer from a problem at thick trailing edges, as commonly found on turbine blades. If a fine mesh is used around the trailing edge, the flow on the pressure surface usually turns some distance around the trailing edge curve before separating. This is because the boundary layer on the pressure surface is extremely thin and immediately upstream of the trailing edge, it is far from separation, it is less of a problem on the suction surface, because the boundary layer there is much thicker and probably near separation. This leads to a low pressure on the pressure surface at the start of the trailing edge curvature and negative loading on the rear of the blade as illustrated in Fig. 13. Such negative loading is never found in practice where, in subsonic flow, the time-averaged pressures on both surface meet smoothly at the trailing edge. In reality, the flow is unsteady with vortex shedding, and this cannot be captured by a steady calculation, hence, we need to model the trailing edge flow so as to make the loading fall to zero. This is done by the option to use a cusp as illustrated in Fig. 14. The cusp is generated automatically within Multall if one is requested.

A relatively coarse grid is used near the trailing edge, the blade surfaces are extrapolated to the trailing edge and a wedge, typically three cells in length, is added immediately downstream. The wedge may be aligned with either blade surface or, more usually, with the center line. The surfaces of the wedge are porous, and the

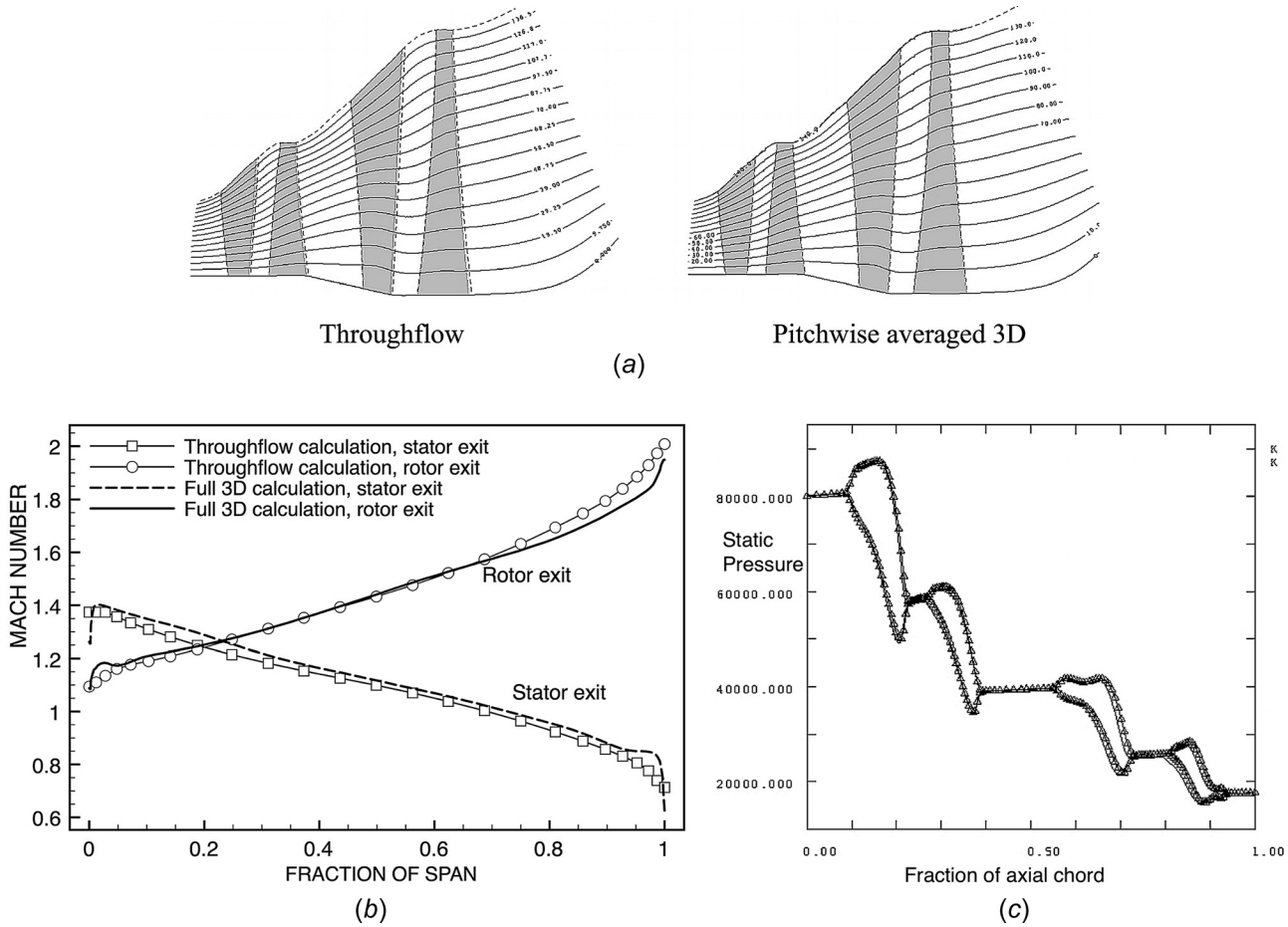

Fig. 12 (*a*) Comparison of streamlines from throughflow and full 3D calculations, (*b*) comparison of throughflow and 3D solutions in the last stage of the LP steam turbine, and (*c*) midspan surface pressure distribution calculated by the throughflow method

mass flow through both faces is made equal to the average over both faces so there is no source of mass. The velocities, pressures, and enthalpies are also averaged so there is no source of tangential momentum or energy. However, the different projected areas of the faces in the axial direction means that there is an axial force, equal to the average pressure × trailing edge thickness, acting on the flow. This corresponds to the base pressure acting on a real trailing edge. There is no guarantee that this effective base pressure is the same as the real base pressure, which is determined by unsteady effects, but it is much better than the low pressure obtained with a fine trailing edge grid. Figure 13 shows that this can influence the pressure distribution over a significant part of the blade.

Low Speed and Incompressible Flow. Time marching methods suffer from problems at low Mach numbers for two reasons. First, the time-step is determined by the speed of sound but the time for the flow, hence enthalpy and entropy, to convect through a blade row is determined by the flow velocity, hence at low Mach numbers, many more time-steps are required to convect out errors in the initial guess. Second, changes in density become so small that they are subject to rounding errors in the computer. When converted to pressure changes, these cause significant random fluctuations in the flow and prevent convergence. The "scree" scheme works better than most at low Mach numbers, and although convergence is inevitably slower it can usually be achieved with average Mach numbers above about 0.15.

However, a simple means to overcome these limitations has been developed, and this is based on the concept of artificial compressibility. The idea is to solve the continuity equation for an artificial density ρ_s whose changes are significant, and to use this to calculate the pressure using

$$(P - P_{\text{ref}}) = S^2(\rho_s - \rho_{s,\text{ref}}) \tag{11}$$

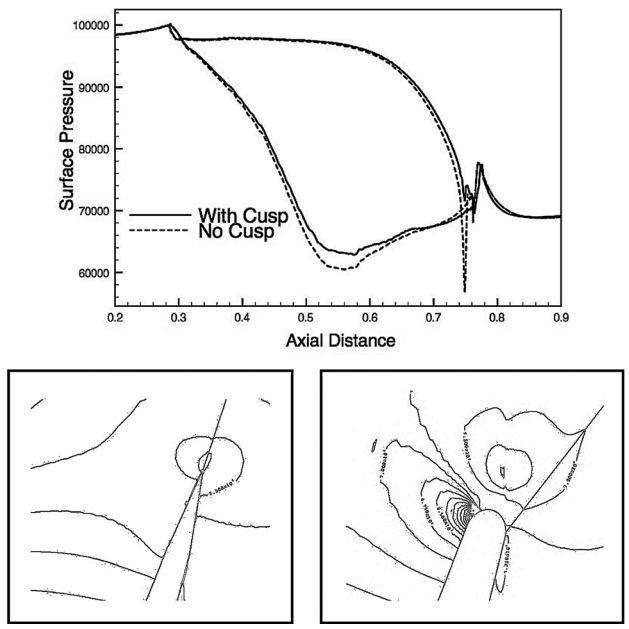

Fig. 13 Computed trailing edge pressure distributions with and without a cusp

P_{ref} is usually the inlet stagnation pressure, and $\rho_{s,ref}$ is the inlet stagnation density. S is an artificial speed of sound whose value is set in the data and is typically about twice the maximum relative velocity expected in the flow, much less than the true speed of sound. The time-steps are based on this artificial speed and so can be larger than the conventional steps by a factor c/S. The changes in the artificial density are a factor $(c/S)^2$ greater than those of the true density and so are not so susceptible to rounding errors. The true density, ρ, undergoes only small changes at low Mach numbers and is only used to obtain the velocities, V, from the mass fluxes, ρV. Hence, it can be calculated from the pressure and temperature, using the gas law, with relaxation of the changes

$$\rho_{new} = (1 - RF)\rho_{old} + RF(P/RT) \tag{12}$$

The relaxation factor, RF, is not critical but a value about 0.01 is usually chosen. For fully incompressible flow, the density is simply help constant.

Figure 15 shows a solution for a water pump, designed by Meangen, with a maximum velocity about 15 m/s and density = 1000 kg/m^3.

Design Exercise

To illustrate the use of the system for design, the steps used to design a large 3 stage axial compressor are shown here. The specification was

- mass flow; 50 kg s,
- ambient air inlet conditions,
- stagnation pressure ratio; 2.0,
- rotational speed; 5000 rpm, and
- outer diameter; 1 m.

The compressor is purely hypothetical and is not based on any real machine.

Using a guessed efficiency with the same loading coefficient for each stage, the loading coefficient required is 0.36, a typical value for moderately loaded compressors. This value was chosen for input to Meangen, together with a flow coefficient of 0.6 and a constant meridional velocity. The Mach numbers into the first rotor would be near sonic so to reduce this 15 deg of inlet swirl was specified. This would have to be provided by inlet guide vanes, which were not included in the calculation. The interstage swirl was gradually reduced so that there was no swirl in the exit flow. Hence, the stages do not have repeating velocity triangles and need to be designed using the second option in Meangen. Some default values in Meangen, e.g., air properties, blade thicknesses, blade row gaps, deviation angles, etc., were set to suit the machine.

Initially, a relatively coarse grid of $37 \times 100 \times 37$ points per blade row with no tip or seal leakages was set. With this grid solution times were about 15 min per run. The design sequence proceeded as listed below, "M" means that changes were made only in Meangen, "S" that detailed blade profile changes were made in Stagen.

- M—Produce an initial layout by running Meangen with default parameters and screen input. The solution has slightly low mass flow and pressure ratio.
- M—Increase the guess of deviation angles to increase the pressure ratio. Increase the blade thickness and blade numbers. Several runs to obtain flow and pressure ratio close to the specified values.
- M—The blades are mid loaded. Move the point of maximum thickness and point of maximum camber forward to obtain more fore-loaded blades. Reduce the trailing edge thickness. Several runs to obtain the required mass flow and pressure ratio.
- M—Move the point of maximum camber slightly further forward. The blades now have good surface pressure distributions but there are high incidences on some sections. Adjust the average incidence angles for each row. Several runs.
- S—Change to a finer $55 \times 140 \times 55$ point grid with tip gaps on all rotors, five cells in each tip gap. Adjust the incidence angles along the span. Rotor 3 is most highly loaded, so increase its blade numbers. Several runs, about 30 min per run when starting from the previous solution.
- S—Add hub shroud leakages on all stators. Adjust incidences.
- S—Bow all stators, with pressure surfaces leaned toward the endwalls, to reduce endwall loadings. Re-adjust the incidences. The performance is now very close to specification with a predicted isentropic efficiency of 91.5%.
- S—To improve the stall margin add forward sweep at all rotor tips by increasing the chord by 15% with a fixed trailing edge location.
- S—Further refine the blade incidences. Several runs.
- S—Run a characteristic from choke to stall.

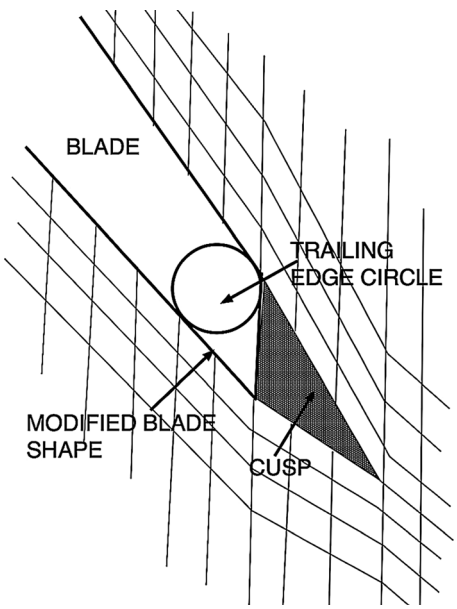

Fig. 14 Grid for a cusp model at the trailing edge

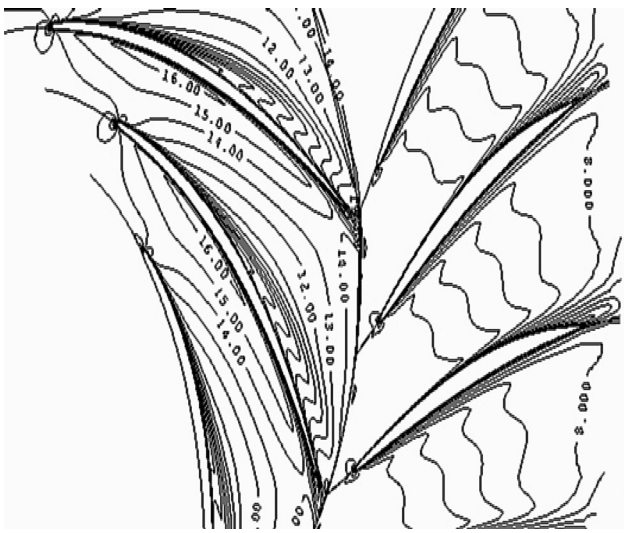

Fig. 15 Contours of relative velocity through a water pump

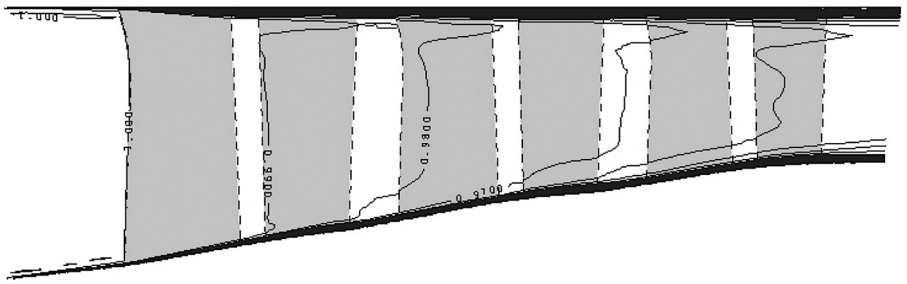

Fig. 16 Compressor layout with contours of pitchwise averaged entropy

Fig. 17 Mach number contours at midspan

- S—Run a very fine grid solution, $83 \times 299 \times 83$ points per row, at the design point to check for any grid sensitivity. The mass flow and pressure ratio each changed by about 0.1% and the efficiency was 0.4% lower.

In total about 30 3D runs were performed to finalize the design. This used about 15 h of CPU time using a single processor on a Linux desktop computer. Allowing for plotting time and thinking time the process could have been completed in about 3 man-days. This could be reduced if the program were parallelized but the author finds that running time can double up with thinking time, and so it is not desirable for it to become too short.

Figure 16 shows a meridional view of the final design with entropy contours, and Fig. 17 shows the midspan Mach number contours, Fig. 18 shows the surface pressure distribution, and Fig. 19 shows the computed characteristic. At the highest pressure ratio, the calculation failed to converge due to large separations near the stator tips where they are impacted by the rotor tip leakage flow. This is usually taken to be the predicted stall point of the machine.

Discussion

Space has only permitted a brief description of the design system, especially of Multall. Further details can be found in the user manual. The main attractions of the system are felt to be the availability of the source code and its relative speed and simplicity.

Learning to use any new CFD system can be a frustrating experience until the user becomes familiar with the numerous options. Nothing can replace actually running the program and studying the results if one wants learn and understand a code. By using the default options in Meangen, it should usually be possible to generate an initial data set and obtain a 3D solution from Multall at the first attempt. Detailed refinements such as tailored blade shapes, tip and shroud clearances, different inter row gaps, inlet flow profiles, etc., can be added gradually to build up the final design. Numerous sample data sets are provided to help with this.

In addition to its use as a design system, some of the methods used in Multall, such as the "scree" scheme, multigrid method, mixing plane model, incompressible flow model, negative feedback, and cusp model may be of use to other CFD developers. They are most easily understood by studying the coding and other developers are welcome to copy them.

Multall is limited to calculating the main blade path and to steady flow only. A multiblock, steady or unsteady code, Tblock, which uses the same algorithm and much of the same modeling, has also been developed by the author, this can handle much more complex geometries and is quite widely used, e.g., see Ref. [16]. A further development of Tblock by Brandvik and Pullan, called Turbostream [17], runs on GPU's and is able to perform very

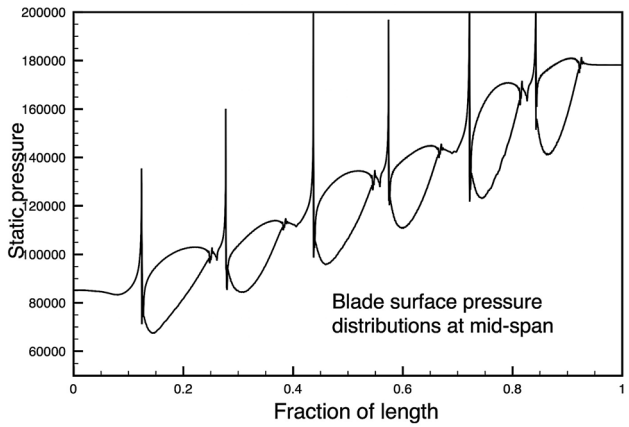

Fig. 18 Surface pressure distribution at midspan

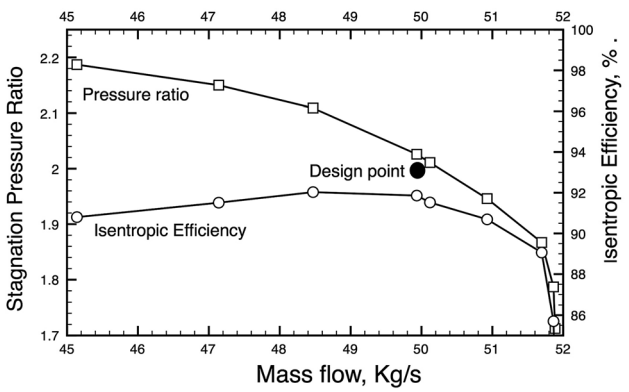

Fig. 19 Computed characteristic of the compressor

large-scale unsteady calculations. These codes are not freely available.

Acknowledgment

Many people have contributed to the development of these codes, this includes users of the codes who have contributed their experience and helped to find many "bugs." The author would like to thank them all. Thanks are especially due to Dr Liping Xu and Dr Simon Gallimore, for their continued assistance.

Nomenclature

c = speed of sound
C_f = skin friction coefficient
F = conserved variable, ρ, ρV, or ρE
m = meridional distance
P = static pressure
r = radius
R = gas constant
Re = Reynolds number
S = artificial speed of sound
t = time
T = static temperature
U = blade speed
V = velocity component
vol = cell volume
y = distance from a wall
α = absolute flow angle
β = relative flow angle
θ = circumferential angle
λ = stage reaction
ν = kinematic viscosity
ρ = fluid density
τ = wall shear stress
ϕ = flow coefficient
ψ = stage loading coefficient

Subscripts

0 = at inlet to a stage
1 = at inlet to a blade row
2 = at exit from a blade row
4 = at stage exit

Appendix: Downloading the System

The source programs, user manuals and sample data sets can be downloaded as a folder named "multall-open" from DROPBOX.[1]

You do not need a DROPBOX account to access this, simply click "download anyhow" if asked to set up an account.

The link can also be found on the website.[2]

Any future developments to the programs or changes to the link will be announced on this web site.

References

[1] Denton, J. D., 1994, "Designing in Three Dimensions," *Turbomachinery Design Using CFD* (AGARD Lecture Series No. 195), AGARD Advisory Group for Aerospace Research & Development, Neuilly sur Seine, France.
[2] Denton, J. D., and Xu, L., 1999, "The Exploitation of Three-Dimensional Flow in Turbomachinery Design," Proc. Inst. Mech. Eng., J. Mech. Eng. Sci., **213**(C2), pp. 125–137.
[3] Turner, M. G., Merchant, A., and Bruna, D., 2006, "A Turbomachinery Design Tool for Teaching Design Concepts for Axial Flow Fans, Compressors and Turbines," ASME Paper No. GT2006-90105.
[4] Denton, J. D., 1975, "A Time Marching Method for Two and Three Dimensional Blade-to-Blade Flows," UK Aero Research Council, Cranfield, UK, Report.
[5] Denton, J. D., and Singh, U. K., 1979, "Time Marching Methods for Turbomachinery Flow Calculation," *Transonic Flows in Turbomachinery* (VKI Lecture Series), von Karman Institute for Fluid Dynamics, Sint-Genesius-Rode, Belgium.
[6] Denton, J. D., 1990, "The Calculation of Three Dimensional Viscous Flow Through Multistage Turbomachines," ASME Paper No. 90-GT-019.
[7] Bryanston-Cross, P. J., and Denton, J. D., 1984, "Comparison of Measured and Predicted Transonic Flow Around an Aerofoil," AIAA J., **22**(8), pp. 1025–1026.
[8] Fottner, L., ed., 1990, *Test Cases for Computation of Internal Flows in Aero Engine Components* (AGARD Advisory Report No. AR-275), AGARD Advisory Group for Aerospace Research & Development, Neuilly sur Seine, France.
[9] Jameson, A., 1983, "Solution of the Euler Equations for a Two Dimensional Transonic Flow by a Multigrid Method," Appl. Math. Comput., **13**(3–4), pp. 327–356.
[10] Peterson, A., 2006, "Cavitation Prediction," Ph.D. thesis, Cambridge University, Cambridge, UK.
[11] Holmes, D. G., 2008, "Mixing Planes Revisited. A Steady State Mixing Plane Approach Designed to Combine High Levels of Conservation and Robustness," ASME Paper No. GT2008-51296.
[12] White, F. M., 1991, *Viscous Fluid Flow*, 2nd ed., McGraw-Hill, New York.
[13] Baldwin, B. S., and Lomax, H., 1978, "Thin Layer Approximation and Algebraic Model for Separated Turbulentflows," AIAA Paper No. 78-257.
[14] Spurr, A., 1980, "The Prediction of 3D Transonic Flow in Turbomachinery Using a Combined Throughflow and Blade-to-Blade Time Marching Method," Int. J. Heat Fluid Flow, **2**(4), pp. 189–199.
[15] Sturmayr, A., and Hirsch, C., 1999, "Shock Representation by Euler Throughflow Models and Comparison With Pitch Averaged Navier–Stokes Equations," American Institute of Aeronautics and Astronautics, Reston, VA, ISABE Paper No. 99-7281.
[16] Jacobi, S., and Rosic, B., 2015, "Development and Aerothermal Investigation of Integrated Combustor Vane Concept," ASME Paper No. GT2015-43217.
[17] Brandvik, T., and Pullan, G., 2011, "An Accelerated Navier–Stokes Solver for Flows in Turbomachines," ASME J. Turbomach., **133**(2), p. 021025.

[1] https://www.dropbox.com/sh/8i0jyxzjb57q4j4/AABD9GQ1MUFwUm5hMWFylucva?dl=0
[2] https://sites.google.com/view/multall-turbomachinery-design