Three-Dimensional Elliptic Grid Generation Technique With Application to Turbomachinery Cascades

S.C. Chen
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio

and

J.R. Schwab
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio

August 1988
THREE-DIMENSIONAL ELLIPTIC GRID GENERATION TECHNIQUE
WITH APPLICATION TO TURBOMACHINERY CASCADES

S. C. Chen
Sverdrup Technology, Inc.
NASA Lewis Research Center Group
Cleveland, Ohio 44135

and

J. R. Schwab
National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

SUMMARY

This report describes a numerical method for generating three-dimensional grids for turbomachinery computational fluid dynamics codes. The basic method is general and involves the solution of a quasi-linear elliptic partial differential equation via pointwise relaxation with a local relaxation factor. It allows specification of the grid point distribution on the boundary surfaces, the grid spacing off the boundary surfaces, and the grid orthogonality at the boundary surfaces. It includes adaptive mechanisms to improve smoothness, orthogonality, and flow resolution in the grid interior. A geometry preprocessor constructs the grid point distributions on the boundary surfaces for general turbomachinery cascades. Representative results are shown for a C-grid and an H-grid for a turbine rotor. Two appendices serve as user's manuals for the basic solver and the geometry preprocessor.

INTRODUCTION

Three-dimensional computational fluid dynamics codes require computational grids with suitable resolution, smoothness, and orthogonality. High grid resolution allows complex flow physics to be modeled near shocks and in shear layers. Smoothness of the metric data prevents the flow solution from being dominated by truncation error in the metric coefficients. Grid orthogonality at the boundaries simplifies and improves the accuracy of any boundary condition involving normal gradients.

The above qualities are especially difficult to maintain for realistic turbomachinery geometries. Modern three-dimensional designs comprise tapered, twisted, leaned, and bowed blade shapes within contoured endwalls to tailor secondary flows. Centrifugal compressors and radial turbines involve simultaneous flow turning in the meridional and blade-to-blade planes. The periodicity condition within blade rows poses an additional unique problem.

Current grid generation technology is fairly well developed for general two-dimensional turbomachine cascade geometries. Conformal mapping, algebraic interpolation, and partial differential equation methods are all used successfully. The general three-dimensional geometry represented by a realistic turbomachine cascade has unique requirements for a body-fitted grid that have not been met with current technology.

This report describes a numerical method for generating three-dimensional grids for turbomachinery computational fluid dynamics codes. The basic method is general and involves the solution of a quasi-linear elliptic partial differential equation via pointwise successive over-relaxation with a local relaxation factor. The governing equation contains forcing functions that depend upon the boundary point distribution and the boundary surface gradient. The method allows specification of the grid point distribution on the boundary surfaces, the grid spacing off the boundary surfaces, and the grid orthogonality at the boundary surfaces. It includes adaptive mechanisms to improve smoothness, orthogonality, and flow resolution in the grid interior.

A geometry preprocessor constructs the grid point distributions on the boundary surfaces for general turbomachinery cascades. It utilizes a two-dimensional version of the basic solver and algebraic interpolation to form the boundary distributions for the three-dimensional basic solver.
This report includes a description of the coordinate system, a discussion of the mathematical formulation of the method, and some representative results. Two appendices serve as the user's manuals for the geometry preprocessor and the basic solver.

COORDINATE SYSTEM

The partial differential equations for computational fluid dynamics codes are usually described with reference to a generalized coordinate system to simplify the implementation and make it independent of any specific geometry. The function of a grid generation system is to generate an ordered distribution of points in physical space to align with some body-conforming generalized coordinate system in computational space. Physical space is described with reference to a cartesian coordinate system for the grid generation technique described in this report. Each boundary surface segment in physical space must coincide with a boundary surface segment in computational space.

Three types of generalized body-conforming coordinate systems are commonly used for turbomachinery cascades: C-grid, H-grid, and O-grid. The grid generation technique described in this report can produce C-grids and H-grids.

Figure 1(a) shows a C-grid about a generic blade shape in the cartesian \((x_1, x_2, x_3)\) coordinate system in physical space. The inlet surface is \(A_1-A_2-A_2'-A_1'\) and the outlet surface is \(B_1-B_2-B_2'-B_1'\). The hub endwall surface is \(A_1'-B_1'-B_2'-A_2'\) and the shroud endwall surface is \(A_1-B_1-B_2-A_2\). The periodic surfaces are \(A_1-B_1'-A_1'\) and \(A_2-B_2-B_2'-A_2'\). The blade surface is the wrapped \(D'-C'-D\) surface. Surface \(C-D-D'-C'\) represents a branch cut from the wrapped blade surface to the outlet surface. Figure 1(b) shows the C-grid in the generalized \((\xi_1, \xi_2, \xi_3)\) coordinate system in computational space.

Figure 2(a) shows an H-grid between two generic blade shapes in the cartesian \((x_1, x_2, x_3)\) coordinate system in physical space. The inlet surface is \(A_1-A_2-A_2'-A_1'\) and the outlet surface is \(B_1-B_2-B_2'-B_1'\). The hub endwall is \(A_1'-B_1'-B_2'-A_2'\) and the shroud endwall surface is \(A_1-B_1-B_2-A_2\). The periodic surfaces are \(A_1-C_1-C_2-C_2'-A_1'\), \(A_2-C_2-C_2'-A_2'\), \(D_1-B_1-B_2-D_1'\), and \(D_2-B_2-B_2'-D_2'\). The blade surfaces are \(C_1-D_1-D_1'-C_1'\) and \(C_2-D_2-D_2'-C_2'\). Figure 2(b) shows the H-grid in the generalized \((\xi_1, \xi_2, \xi_3)\) coordinate system in computational space.

MATHEMATICAL FORMULATION

The quasi-linear elliptic governing equation is taken from reference 1 as

\[
\sum_{i=1}^{3} \sum_{j=1}^{3} g^{ij} \frac{\partial^2 \tilde{x}}{\partial \xi_i \partial \xi_j} + \sum_{k=1}^{3} g^{kk} P_k \frac{\partial \tilde{x}}{\partial \xi_k} = 0. \tag{1}
\]

The metric tensor components \(g^{ij}\) and \(g^{kk}\) in equation (1) are defined as

\[
g^{ij} = \tilde{a}^i \cdot \tilde{a}^j \tag{2}
\]

where

\[
\tilde{a}^i = a_j \times a_k / \sqrt{g} \quad i,j,k \text{ cyclic}
\]

\[
\sqrt{g} = \tilde{a}_1 \cdot (\tilde{a}_2 \times \tilde{a}_3)
\]

\[
\tilde{a}_i = \frac{\partial \tilde{x}}{\partial \xi_i}.
\]

The \(\tilde{x}\) in equation (1) represents the position vector in physical space with cartesian \((x_1, x_2, x_3)\) components. The \(P_k\) are the forcing functions specified by the user. They represent one-dimensional stretching in each coordinate direction. Values of the forcing functions on the boundaries are determined by specification of the boundary point distribution and the boundary surface gradient. Values of the forcing functions in the interior are determined by interpolation of the values on the boundaries.
Equation (1) can be rewritten using matrix notation as

\[ A + B P = 0 \]  

where

\[ A = \begin{pmatrix} g_{11} \frac{\partial^2 \phi}{\partial \xi_1^2} & g_{12} \frac{\partial^2 \phi}{\partial \xi_1 \partial \xi_2} & g_{13} \frac{\partial^2 \phi}{\partial \xi_1 \partial \xi_3} \\ g_{21} \frac{\partial^2 \phi}{\partial \xi_2 \partial \xi_1} & g_{22} \frac{\partial^2 \phi}{\partial \xi_2^2} & g_{23} \frac{\partial^2 \phi}{\partial \xi_2 \partial \xi_3} \\ g_{31} \frac{\partial^2 \phi}{\partial \xi_3 \partial \xi_1} & g_{32} \frac{\partial^2 \phi}{\partial \xi_3 \partial \xi_2} & g_{33} \frac{\partial^2 \phi}{\partial \xi_3^2} \end{pmatrix} \]

\[ B = \begin{pmatrix} g_{11} \frac{\partial^2 \phi}{\partial \xi_1^2} & 0 & 0 \\ 0 & g_{22} \frac{\partial^2 \phi}{\partial \xi_2^2} & 0 \\ 0 & 0 & g_{33} \frac{\partial^2 \phi}{\partial \xi_3^2} \end{pmatrix} \]

\[ P = \begin{pmatrix} P_1 \\ P_2 \\ P_3 \end{pmatrix} \]

Equation (3) can be solved for \( P_k \) on the boundaries as

\[ P_0 = -B_0^{-1} A_0 \]

where the subscript "0" indicates values on the boundary. Tangential derivatives for terms on the right hand side of equation (4) are determined by applying standard difference formulas to the prescribed boundary point distribution on the surface. Normal derivatives are determined by specifying the first normal derivative equal to the desired spacing off the boundary and using the approximation

\[ \frac{\partial^2 \bar{z}}{\partial n^2} = \frac{2(\bar{z}_1 - \bar{z}_0)}{(\Delta n)^2} - \frac{2}{\Delta n} \frac{\partial \bar{z}_0}{\partial n} \]  

where \( n \) indicates the normal direction, the subscript "0" indicates values on the boundary, and the subscript "1" indicates values one point away from the boundary.

Once the boundary values are known, the interior values of \( P_k \) can be determined using

\[ P_k(\xi_1, \xi_2, \xi_3) = \sum_{l=1}^{3} P_{0k,l,1} (1 + \alpha_{k,l,1}) \gamma_{k,l,1} + \sum_{l=1}^{3} P_{0k,l,2} (1 + \alpha_{k,l,2}) \gamma_{k,l,2} \]  

where

\[ \alpha_{k,l,1} = C_{\alpha_{k,l,1}} (\xi_i - \xi_{im}) / (\xi_{im} - \xi_{im-1}) \]

\[ \alpha_{k,l,2} = C_{\alpha_{k,l,2}} (\xi_{im} - \xi_i) / (\xi_{im} - \xi_{im-1}) \]

\[ \beta_{k,l,1} = [(\xi_{im} - \xi_{im-1}) / (\xi_i - \xi_{im-1})]^{C_{\beta_{k,l,1}}'} \]

\[ \beta_{k,l,2} = [(\xi_{im} - \xi_{im-1}) / (\xi_i - \xi_{im-1})]^{C_{\beta_{k,l,2}}'} \]

\[ \gamma_{k,l,1} = \beta_{k,l,1} / \left( \sum_{m=1}^{3} \beta_{k,l,1} + \sum_{m=1}^{3} \beta_{k,l,2} \right) \]

\[ \gamma_{k,l,2} = \beta_{k,l,2} / \left( \sum_{m=1}^{3} \beta_{k,l,1} + \sum_{m=1}^{3} \beta_{k,l,2} \right) \]

The value of \( P_{0k,l,1} \) represents the \( k \)-th component of the \( P_0 \) vector on the minimum boundary surface in the \( l \)-th direction. The value of \( P_{0k,l,2} \) represents the \( k \)-th component of the \( P_0 \) vector on the maximum boundary surface in the \( l \)-th direction. The functions \( \alpha \), \( \beta \), and \( \gamma \) have subscript notation similar to that of \( P_0 \). The \( \alpha \) function represents linear extrapolation from a controlled boundary using a constant factor \( C_{\alpha} \).
The $\gamma$ function represents the combined effect of the $\beta$ functions, which represent power-law factorization with constant exponent $C_\beta$ to control the depth of influence away from a controlled boundary.

The values of the forcing functions can be modified for improved smoothness by using

$$P'_k = \theta \, P_k \tag{7}$$

where

$$\theta = 1 - \tanh(C_{\theta_1}(1 - \sigma C_{\theta_2}))$$

$$\sigma = J/((||l_1|| \cdot ||l_2|| \cdot ||l_3||)).$$

The constants $C_{\theta_1}$ and $C_{\theta_2}$ define the rate and order of the adaptation. The variable $\sigma$ is a measure of the shear of the grid, with $J$ representing the Jacobian and $l_1, l_2, l_3$ representing the grid cell lengths in each direction.

A measure of the local orthogonality of the grid can be defined as

$$\phi = \sigma C_\phi / J. \tag{8}$$

The constant exponent $C_\phi$ defines the order of the adaptation. A one-dimensional variational form of equation (8) can then be written as

$$\phi'_k = \frac{\delta_k(J)}{J} - C_\phi \frac{\delta_k(\sigma)}{\sigma} \tag{9}.$$ 

If a flow variable gradient $E$ is computed by the flow solver such that $E \geq 0$, a measure of the local flow resolution can be defined as

$$\psi = J(1 + E). \tag{10}$$

A one-dimensional variational form of equation (10) can then be written as

$$\psi'_k = \frac{\delta_k(J)}{J} - \frac{\delta_k(E)}{1 + E}. \tag{11}.$$ 

The values of the forcing functions can be modified for improved local orthogonality and flow resolution using

$$P'_k = (1 + \lambda_k)P_k \quad F_k \cdot P_k > 0$$

$$P'_k = (1 - \lambda_k)P_k \quad F_k \cdot P_k < 0 \tag{12}$$

where

$$\lambda_k = C_{\lambda_1} \tan(C_{\lambda_2} |F_k|^{C_{\lambda_3}})$$

$$F_k = (W_\phi \phi'_k + W_\psi \psi'_k)/(W_\phi + W_\psi)$$

with the constants $C_{\lambda_1}, C_{\lambda_2}$, and $C_{\lambda_3}$ determining the range, rate, and order of the adaptation. The variable $F$ represents a weighted combination of the skewness and flow error variations, with $W_\phi$ and $W_\psi$ as the respective weighting constants.

These adaptive adjustments to the forcing functions can be employed in an accumulative or a non-accumulative manner. Under the accumulative method, the adjustments are lagged, but the forcing functions always satisfy the physical constraints. Under the non-accumulative method, the adjustments are immediate, but they are limited by the range and rate constants.

The following sequence is iterated until convergence is attained:

1. Solve the governing equation with current forcing functions using pointwise relaxation with a local relaxation factor for stability.
2. Evaluate the second normal derivatives on the boundary surfaces.
3. Evaluate the forcing functions on the boundary surfaces.
4. Evaluate the forcing functions in the interior by interpolation.
5. Adjust the forcing functions for adaptive smoothness, orthogonality, and flow resolution.

RESULTS

Figure 3 shows a three-dimensional C-grid for a turbine rotor. The grid comprises 101 points in the $\xi_1$ direction around the blade, 16 points in the $\xi_2$ direction away from the blade, and 15 points in the $\xi_3$ direction from hub to tip. The grid was truncated in the $\xi_1$ direction and thinned in the $\xi_2$ direction to improve the appearance of the figure. The spacing at the solid surfaces was specified as 0.1 relative to the uniform unit spacing. Default values were used for all other variables. This grid required approximately 1200 seconds of Cray X-MP computing time for 600 iterations.

Figure 4 shows a three-dimensional H-grid for the same turbine rotor as Figure 3. The grid comprises 61 points in the streamwise $\xi_1$ direction, 31 points in the pitchwise $\xi_2$ direction, and 15 points in the radial $\xi_3$ direction. The grid was thinned in the $\xi_2$ direction to improve the appearance of the figure. The spacing at the solid surfaces was specified as 0.1 relative to the uniform unit spacing. Default values were used for all other variables. This grid required approximately 1400 seconds of Cray X-MP computing time for 600 iterations.

CONCLUDING REMARKS

A general numerical method for generating three-dimensional grids was developed and implemented along with a geometry preprocessor for turbomachinery cascades. Results were shown for a C-grid and an H-grid for a turbine rotor. The method includes an adaptive mechanism for improved flow resolution when coupled with a flow solver, but this was not demonstrated. Since the basic method is completely general, additional preprocessors for other physical geometries could be developed to extend the application of this grid generation method.
APPENDIX A

USER'S MANUAL FOR BASIC SOLVER

The basic solver is coded in FORTRAN IV as program GRID3D with 34 subprograms. The nonstandard PARAMETER statement is used to facilitate dimensioning of arrays, while the nonstandard NAMELIST feature is used for quasi-free-format input. Figure 5 shows a flow chart of the program.

Input to the basic solver comprises three namelists read on unit 5 and the initial grid binary file read on unit 10. The type (REAL or INTEGER) of each variable follows standard FORTRAN convention. The I, J, and K indices correspond to the $\xi_1$, $\xi_2$, and $\xi_3$ directions, respectively.

Namelist SYSTEM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| ISYS      | 1: Solve governing equation as a Laplacian system without forcing functions (default). This option produces a grid that tends toward uniform spacing away from curved boundaries, but produces smaller spacing near convex boundaries and larger spacing near concave boundaries.  
            2: Solve governing equation as a Poisson system with forcing functions. This option allows specification of spacing and orthogonality at the boundaries. |
| INTERP    | 1: Run three-dimensional interpolation as initial guess (default). This option is used when the geometry preprocessor is employed to generate the initial grid binary file without interior values.  
            0: Bypass three-dimensional interpolation. This option is used when a complete initial grid binary file is imported from another grid generator. |
| ICHECK    | 1: Check boundary structures for grid folding and write report on unit 20.  
            0: Bypass boundary structure check (default). |
| ICONT     | 0: Terminate operation if boundary structure grid folding is detected.  
            1: Continue operation (default). |
| ITMAX     | Maximum number of overall iterations (default = 100). |
| LREFI     | 1: Establish a reference length scale for spacing off the $I = 1$ and $I = \text{IMAX}$ surfaces based upon the unit length in the J-direction for each K-layer at the $I = 1$ surface. The unit length is defined as ($\text{total length} / (\text{number of points}-1)$).  
            2: Establish a reference length scale for spacing off the $I = 1$ and $I = \text{IMAX}$ surfaces based upon the local unit length in the J-direction. The unit length is defined as ($\text{total length} / (\text{number of points}-1)$).  
            3: Establish a reference length scale for spacing off the $I = 1$ and $I = \text{IMAX}$ surfaces based upon the value of FACI1 from the initial grid binary file or the three-dimensional interpolation (default). |
| LREFJ     | Same as above for spacing off the $J = 1$ and $J = \text{JMAX}$ surfaces. |
| LREFK     | Same as above for spacing off the $K = 1$ and $K = \text{KMAX}$ surfaces. |
| RLXSOR    | Relaxation factor for the SOR solver. Suitable values range from 0.0 to 2.0 (default = 1.0). |
| RLXBDE    | Relaxation factor for extrapolated second derivatives at the boundary surfaces (default = 0.15). |
| RLXADP    | Relaxation factor for grid adaptation (default = 0.15). |
| EPSFLD    | Folding grid point criterion (default = 0.0). |
| EPSBAD    | Sheared grid point criterion (default = $\text{sine 10}^\circ$). |
| EPSCNV    | Convergence criterion (default = 0.0001). |

Namelist PBASIC

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
</table>
| IICONT    | 1: Control spacing and orthogonality off the $I = 1$ surface.  
            0: Do not control spacing and orthogonality off the $I = 1$ surface (default). |
The following parameters control the depth of grid clustering away from the controlled surface. The desired spacing of the \( I = I_{\text{MAX}} \) surface.

Same as above for \( J = J_{\text{MAX}} \) surface.

Same as above for \( K = K_{\text{MAX}} \) surface.

Desired spacing of the \( I = 1 \) surface. If \( \text{DELTII} = 0.0 \), the value will be set equal to the value of \( \text{FACI1} \) from the initial grid binary file (default).

Same as above for the desired spacing of the \( I = I_{\text{MAX}} \) surface.

Same as above for the desired spacing of the \( J = 1 \) surface.

Same as above for the desired spacing of the \( J = J_{\text{MAX}} \) surface.

Same as above for the desired spacing of the \( K = 1 \) surface.

Same as above for the desired spacing of the \( K = K_{\text{MAX}} \) surface.

The following parameters control the depth of grid orthogonality away from the controlled boundary. Larger positive values produce more smoothing near discontinuous areas.

The following parameters control the depth of grid clustering away from the controlled surface. The default value of 0.0 produces a quasi-linear rate of spacing increase away from the boundary. Positive values produce a nonlinear rate with continuously increasing spacing in the interior. Negative values produce a nonlinear rate with near-uniform spacing in the interior.

\[ \text{APII} \]
\[ C_{\alpha_{1,1}} \] for \( \xi_{1} \)-direction extrapolation of \( P_1 \) from \( I = 1 \) surface.

\[ \text{APIM} \]
\[ C_{\alpha_{1,2}} \] for \( \xi_{1} \)-direction extrapolation of \( P_1 \) from \( I = I_{\text{MAX}} \) surface.

\[ \text{APJ1} \]
\[ C_{\alpha_{2,1}} \] for \( \xi_{2} \)-direction extrapolation of \( P_1 \) from \( J = 1 \) surface.

\[ \text{APJM} \]
\[ C_{\alpha_{2,2}} \] for \( \xi_{2} \)-direction extrapolation of \( P_1 \) from \( J = J_{\text{MAX}} \) surface.

\[ \text{APKI} \]
\[ C_{\alpha_{3,1}} \] for \( \xi_{3} \)-direction extrapolation of \( P_1 \) from \( K = 1 \) surface.

\[ \text{APKM} \]
\[ C_{\alpha_{3,2}} \] for \( \xi_{3} \)-direction extrapolation of \( P_1 \) from \( K = K_{\text{MAX}} \) surface.

\[ \text{AQI1} \]
\[ C_{\alpha_{1,1}} \] for \( \xi_{1} \)-direction extrapolation of \( P_2 \) from \( I = 1 \) surface.

\[ \text{AQIM} \]
\[ C_{\alpha_{1,2}} \] for \( \xi_{1} \)-direction extrapolation of \( P_2 \) from \( I = I_{\text{MAX}} \) surface.

\[ \text{AQJ1} \]
\[ C_{\alpha_{2,1}} \] for \( \xi_{2} \)-direction extrapolation of \( P_2 \) from \( J = 1 \) surface.

\[ \text{AQJM} \]
\[ C_{\alpha_{2,2}} \] for \( \xi_{2} \)-direction extrapolation of \( P_2 \) from \( J = J_{\text{MAX}} \) surface.

\[ \text{AQL1} \]
\[ C_{\alpha_{3,1}} \] for \( \xi_{3} \)-direction extrapolation of \( P_2 \) from \( K = 1 \) surface.

\[ \text{AQKM} \]
\[ C_{\alpha_{3,2}} \] for \( \xi_{3} \)-direction extrapolation of \( P_2 \) from \( K = K_{\text{MAX}} \) surface.

\[ \text{ARI1} \]
\[ C_{\alpha_{1,1}} \] for \( \xi_{1} \)-direction extrapolation of \( P_3 \) from \( I = 1 \) surface.

\[ \text{ARIM} \]
\[ C_{\alpha_{1,2}} \] for \( \xi_{1} \)-direction extrapolation of \( P_3 \) from \( I = I_{\text{MAX}} \) surface.

\[ \text{ARJ1} \]
\[ C_{\alpha_{2,1}} \] for \( \xi_{2} \)-direction extrapolation of \( P_3 \) from \( J = 1 \) surface.

\[ \text{ARJM} \]
\[ C_{\alpha_{2,2}} \] for \( \xi_{2} \)-direction extrapolation of \( P_3 \) from \( J = J_{\text{MAX}} \) surface.

\[ \text{ARK1} \]
\[ C_{\alpha_{3,1}} \] for \( \xi_{3} \)-direction extrapolation of \( P_3 \) from \( K = 1 \) surface.

\[ \text{ARKM} \]
\[ C_{\alpha_{3,2}} \] for \( \xi_{3} \)-direction extrapolation of \( P_3 \) from \( K = K_{\text{MAX}} \) surface.

The following parameters control the depth of grid orthogonality away from the controlled boundary. Larger positive values produce increased depth (default = 3.0).

\[ \text{BI1} \]
\[ C_{\beta_{1,1}} \] for \( \xi_{1} \)-direction power-law factorization of \( P_1, P_2, P_3 \) from \( I = 1 \) surface.

\[ \text{BIM} \]
\[ C_{\beta_{1,2}} \] for \( \xi_{1} \)-direction power-law factorization of \( P_1, P_2, P_3 \) from \( I = I_{\text{MAX}} \) surface.

\[ \text{BJ1} \]
\[ C_{\beta_{2,1}} \] for \( \xi_{2} \)-direction power-law factorization of \( P_1, P_2, P_3 \) from \( J = 1 \) surface.

\[ \text{BJM} \]
\[ C_{\beta_{2,2}} \] for \( \xi_{2} \)-direction power-law factorization of \( P_1, P_2, P_3 \) from \( J = J_{\text{MAX}} \) surface.

\[ \text{BK1} \]
\[ C_{\beta_{3,1}} \] for \( \xi_{3} \)-direction power-law factorization of \( P_1, P_2, P_3 \) from \( K = 1 \) surface.

\[ \text{BKM} \]
\[ C_{\beta_{3,2}} \] for \( \xi_{3} \)-direction power-law factorization of \( P_1, P_2, P_3 \) from \( K = K_{\text{MAX}} \) surface.

**Namelist PADAPT**

**SMRATE**

Rate of the penalty function for smoothness adaptation (default = 3.0).

Larger values produce smoother grids tending toward the Laplacian solution.

**SMORDR**

Power of the penalty function for smoothness adaptation (default = 1.0).

Larger values produce more smoothing near discontinuous areas.

**SMBDWT**

Parameter providing boundary protection for smoothness adaptation (default = 0.0).
Larger values allow deeper penetration of the specified boundary conditions into the interior at the expense of smoothness adaptation. The maximum value of unity produces a linear penetration.

**OFRNGE**

Range of the penalty function for combined orthogonality and resolution adaptation, when IACCUM = 0 (default = 1.0). Larger values produce more combined adaptation.

**OFORDR**

Lag of the combined orthogonality and resolution adaptation when IACCUM = 1. A zero value produces no lag, while a unity value produces full lag.

**OFRATE**

Rate of the penalty function for combined orthogonality and resolution adaptation. Suggested values are OFRATE = 5.0 when IACCUM = 0 and OFRATE = 50.0 when IACCUM = 1. Larger values produce more combined adaptation.

**OFBDWT**

Power of the penalty function for combined orthogonality and resolution adaptation (default = 0.0). Larger values allow deeper penetration of the specified boundary conditions into the interior at the expense of combined adaptation. The maximum value of unity produces a linear penetration.

**OTORDR**

Power of the skewness function for orthogonality adaptation (default = 2.0). Larger values produce more adaptation near highly sheared areas.

**WTOT**

Weight factor for relative effect of orthogonality adaptation (default = 1.0).

**WTFW**

Weight factor for relative effect of resolution adaptation (default = 0.0).

**IACCUM**

- **= 1** Perform accumulative adaptation. The adjustments to the forcing functions are lagged, but the forcing functions always satisfy the physical constraints (default).
- **= 0** Perform nonaccumulative adaptation. The adjustments to the forcing functions are immediate and are limited only by the range and rate constants.

The initial grid binary file is read from FORTRAN unit 10 with the following code:

```
READ (10) IMAX,JMAX,KMAX
DO 100 K=1,KMAX
   READ (10) ((X(I,J,K),I=1,IMAX),J=1,JMAX),
   * ((Y(I,J,K),I=1,IMAX),J=1,JMAX),
   * ((Z(I,J,K),I=1,IMAX),J=1,JMAX)
100 CONTINUE
READ (10) IGTYPE
READ (10) ISING1,ISING2,ISING3,ISING4
READ (10) JSING1,JSING2,JSING3,JSING4
READ (10) FAC1,FAC1M
READ (10) FAC3,FAC3N
READ (10) FAC5,FAC5M
```

The variables in the binary file are defined as follows:

- **IMAX** Number of grid points in the $\xi_1$ direction.
- **JMAX** Number of grid points in the $\xi_2$ direction.
- **KMAX** Number of grid points in the $\xi_3$ direction.
- **X** Physical Cartesian coordinate in the $z_1$ direction.
- **Y** Physical Cartesian coordinate in the $z_2$ direction.
- **Z** Physical Cartesian coordinate in the $z_3$ direction.
- **IGTYPE**
  - **= 1** C-grid.
  - **= 2** H-grid.
- **ISING1** I-index of first line singularity on the $J = 1$ surface.
- **ISING2** I-index of second line singularity on the $J = 1$ surface.
- **ISING3** I-index of first line singularity on the $J = JMAX$ surface.
- **ISING4** I-index of second line singularity on the $J = JMAX$ surface.
| JSING1       | J-index of first line singularity on the $I = 1$ surface. |
| JSING2       | J-index of second line singularity on the $I = 1$ surface. |
| JSING3       | J-index of first line singularity on the $J = J_{\text{MAX}}$ surface. |
| JSING4       | J-index of second line singularity on the $J = J_{\text{MAX}}$ surface. |

The following values are only used if DELTI1, DELTIM, DELTJ1, DELTJM, DELTK1, and DELTKM in namelist PBASIC are defaulted.

| FACI1       | Desired spacing off the $I = 1$ surface. |
| FACIM       | Desired spacing off the $I = I_{\text{MAX}}$ surface. |
| FACJ1       | Desired spacing off the $J = 1$ surface. |
| FACJM       | Desired spacing off the $J = J_{\text{MAX}}$ surface. |
| FACK1       | Desired spacing off the $K = 1$ surface. |
| FACKM       | Desired spacing off the $K = K_{\text{MAX}}$ surface. |

Output from the basic solver consists of five files. The system message file is written on unit 6. The locations of any boundary and interior folding points are written on units 20 and 30, respectively. The final grid file is written on unit 40 with the same format as the initial grid file. A plot file for graphics post-processing is written on unit 50.

Folding points are identified where the normalized Jacobian is negative. Sheared points are identified where the normalized Jacobian is less than EPSBAD. An error index is computed every ten iterations and is defined as the relative movement of a point normalized by the diagonal length of the grid. Convergence is attained when the absolute maximum error index among all points is less than EPSCNV with no folding points.
APPENDIX B

USER'S MANUAL FOR GEOMETRY PREPROCESSOR

The preprocessor is coded in FORTRAN IV as program BLADE with 39 subprograms. A two-dimensional version of the basic solver is included as subroutine GRID2D to define nodal point distributions on the hub and shroud boundary surfaces. The nonstandard PARAMETER statement is used to facilitate dimensioning of arrays, while the nonstandard NAMELIST feature is used for quasi-free-format input.

Input to the preprocessor comprises three namelists read on unit 5 for GRID2D, one namelist read on unit 7 for BLADE, and formatted blade geometry data on unit 8. The type (REAL or INTEGER) of each variable follows standard FORTRAN convention. The I, J, and K indices correspond to the $\xi_1$, $\xi_2$, and $\xi_3$ directions, respectively.

Namelist SYSTEM (Unit 5)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISYS</td>
<td>= 1 Solve governing equation as a Laplacian system without forcing functions (default). This option produces a grid that tends toward uniform spacing away from curved boundaries, but produces smaller spacing near convex boundaries and larger spacing near concave boundaries.</td>
</tr>
<tr>
<td>ICHECK</td>
<td>= 1 Check boundary structures for grid folding and write report on unit 20.</td>
</tr>
<tr>
<td>ICNT</td>
<td>= 0 Bypass boundary structure check (default).</td>
</tr>
<tr>
<td>ITMAX</td>
<td>= 1 Maximum number of overall iterations (default = 100).</td>
</tr>
<tr>
<td>RLXSOR</td>
<td>Relaxation factor for the SOR solver. Suitable values range from 0.0 to 2.0 (default = 1.0).</td>
</tr>
<tr>
<td>RLXADBDE</td>
<td>Relaxation factor for extrapolated second derivatives at the boundary surfaces (default = 0.15).</td>
</tr>
<tr>
<td>EPSFLD</td>
<td>Folding grid point criterion (default = 0.0).</td>
</tr>
<tr>
<td>EPSBAD</td>
<td>Sheared grid point criterion (default = sine 10°).</td>
</tr>
<tr>
<td>EPSCNV</td>
<td>Convergence criterion (default = 0.0001).</td>
</tr>
</tbody>
</table>

Namelist PBASIC (Unit 5)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICONT</td>
<td>= 1 Control spacing and orthogonality off the I = 1 surface.</td>
</tr>
<tr>
<td>IMCONT</td>
<td>= 0 Do not control spacing and orthogonality off the I = 1 surface (default).</td>
</tr>
<tr>
<td>JCONT</td>
<td>Same as above for J = 1 surface.</td>
</tr>
<tr>
<td>JMCNT</td>
<td>Same as above for J = JMAX surface.</td>
</tr>
<tr>
<td>DELT1</td>
<td>Desired spacing off the I = 1 surface. If DELT1 = 0.0, the value will be set equal to the value of FAC1 from the initial grid binary file (default).</td>
</tr>
<tr>
<td>DELTMI</td>
<td>Same as above for the desired spacing off the I = IMAX surface.</td>
</tr>
<tr>
<td>DELTJ1</td>
<td>Same as above for the desired spacing off the J = 1 surface.</td>
</tr>
<tr>
<td>DELTJM</td>
<td>Same as above for the desired spacing off the J = JMAX surface.</td>
</tr>
</tbody>
</table>

The following parameters control the depth of grid clustering away from the controlled surface. The default value of 0.0 produces a quasi-linear rate of spacing increase away from the boundary. Positive values produce a nonlinear rate with continuously increasing spacing in the interior. Negative values produce a nonlinear rate with near-uniform spacing in the interior.
API1  \( C_{\alpha_{1,1}} \) for \( \xi_1 \)-direction extrapolation of \( P_1 \) from \( I = 1 \) surface.
APIM  \( C_{\alpha_{1,2}} \) for \( \xi_1 \)-direction extrapolation of \( P_1 \) from \( I = \text{IMAX} \) surface.
APJ1  \( C_{\alpha_{1,3}} \) for \( \xi_1 \)-direction extrapolation of \( P_1 \) from \( J = 1 \) surface.
APJM  \( C_{\alpha_{1,4}} \) for \( \xi_1 \)-direction extrapolation of \( P_1 \) from \( J = \text{JMAX} \) surface.
AQI1  \( C_{\alpha_{2,1}} \) for \( \xi_1 \)-direction extrapolation of \( P_2 \) from \( I = 1 \) surface.
AQIM  \( C_{\alpha_{2,2}} \) for \( \xi_1 \)-direction extrapolation of \( P_2 \) from \( I = \text{IMAX} \) surface.
AQJ1  \( C_{\alpha_{2,3}} \) for \( \xi_1 \)-direction extrapolation of \( P_2 \) from \( J = 1 \) surface.
AQJM  \( C_{\alpha_{2,4}} \) for \( \xi_1 \)-direction extrapolation of \( P_2 \) from \( J = \text{JMAX} \) surface.

The following parameters control the depth of grid orthogonality away from the controlled boundary. Larger positive values produce increased depth (default = 3.0).

BI1  \( C_{\beta_{1,1}} \) for \( \xi_1 \)-direction power-law factorization of \( P_1 \) and \( P_2 \) from \( I = 1 \) surface
BIM  \( C_{\beta_{1,2}} \) for \( \xi_1 \)-direction power-law factorization of \( P_1 \) and \( P_2 \) from \( I = \text{IMAX} \) surface
BJ1  \( C_{\beta_{2,1}} \) for \( \xi_2 \)-direction power-law factorization of \( P_1 \) and \( P_2 \) from \( J = 1 \) surface
BJM  \( C_{\beta_{2,2}} \) for \( \xi_2 \)-direction power-law factorization of \( P_1 \) and \( P_2 \) from \( J = \text{JMAX} \) surface

Namelist PADAPT (Unit 5)

SMRATE  Rate of the penalty function for smoothness adaptation (default = 3.0). Larger values produce smoother grids tending toward the Laplacian solution.
SMORDR  Power of the penalty function for smoothness adaptation (default = 1.0). Larger values produce more smoothing near discontinuous areas.
SMBDWT  Parameter providing boundary protection for smoothness adaptation (default = 0.0). Larger values allow deeper penetration of the specified boundary conditions into the interior at the expense of smoothness adaptation. The maximum value of unity produces a linear penetration.
OFRNGE  Range of the penalty function for combined orthogonality and resolution adaptation, when IACCUM = 0 (default = 1.0). Larger values produce more combined adaptation. Lag of the combined orthogonality and resolution adaptation when IACCUM = 1. A zero value produces no lag, while a unity value produces full lag.
OFRATE  Rate of the penalty function for combined orthogonality and resolution adaptation. Suggested values are OFRATE = 5.0 when IACCUM = 0 and OFRATE = 50.0 when IACCUM = 1. Larger values produce more combined adaptation.
OFORDR  Power of the penalty function for combined orthogonality and resolution (default = 1.0). Larger values produce more adaptation near the specified boundary conditions into the interior at the expense of combined adaptation. The maximum value of unity produces a linear penetration.
OFBDWT  Parameter providing boundary protection for combined orthogonality and resolution adaptation (default = 0.0). Larger values allow deeper penetration of the specified boundary conditions into the interior at the expense of combined adaptation. The maximum value of unity produces a linear penetration.
OTORDR  Power of the skewness function for orthogonality adaptation (default = 2.0). Larger values produce more adaptation near highly sheared areas.
WTOT  Weight factor for relative effect of orthogonality adaptation (default = 1.0).
WTFW  Weight factor for relative effect of resolution adaptation (default = 0.0).
IACCUM  = 1 Perform accumulative adaptation. The adjustments to the forcing functions are lagged, but the forcing functions always satisfy the physical constraints (default).
= 0 Perform nonaccumulative adaptation. The adjustments to the forcing functions are immediate and are limited only by the range and rate constants.

Namelist BLDATA (Unit 7)

IGTYPE  = 1 Generate C-grid.
= 2 Generate H-grid.
DELTAT  Periodic pitch angle in radians.
Kcut

Kmax

Fack1

Fack2

Pwk

Nbl

Facb1

Facb2

Pwbd

Nlead

Flcut1

Flcut2

Ntail

Ftcut1

Ftcut2

Pwcut

Nio

Facio1

Facio2

The following parameters apply only to the C-grid:

Isl

Isu

Fpda

Fpdb

Pwin

Fpdc

Fpd

Pwpdr

The formatted blade geometry data is read from FORTRAN unit 8 with the following code:

DO 680 K=1,Kcut
READ (8,1010) Npath
READ (8,1020) (Zpataci),i=1,Npath)
READ (8,1020) (rpath(i),i=1,Npath)
READ (8,1010) Ibldp
READ (8,1020) (Bldpzc(i),i=1,IBLDP)
READ (8,1020) (Bldptc(i),i=1,IBLDP)
The variables in the formatted file are defined as follows:

- **NPATH**: Total number of data points defining the surface cut.
- **ZPATH**: Axial coordinates of data points on the surface cut.
- **RPATH**: Radial coordinates of data points on the surface cut.
- **IBLDP**: Total number of data points defining the pressure or lower surface of blade.
- **BLDPZC**: Axial coordinates of data points on pressure surface.
- **BLDPTC**: Circumferential coordinates of data points on pressure surface (in radians).
- **BLDPRC**: Radial coordinates of data points on pressure surface.
- **IBLDS**: Total number of data points defining the suction or upper surface of blade.
- **BLDSZC**: Axial coordinates of data points on suction surface.
- **BLDSTC**: Circumferential coordinates of data points on suction surface (in radians).
- **BLDSRC**: Radial coordinates of data points on suction surface.
- **SLOPL**: Tangent of inflow angle or leading edge mean camber angle. If SLOPL > 100.0, SLOPL will be reset to match the computed leading edge mean camber angle.
- **SLOPT**: Tangent of outflow angle or trailing edge mean camber angle. If SLOPT > 100.0, SLOPT will be reset to match the computed trailing edge mean camber angle.

Output from the preprocessor consists of eight files. The system message file is written on unit 6. The initial grid file is written on unit 10 for use by the basic solver. The locations of any boundary folding points on the hub and shroud surfaces are written on unit 20. The location of any interior folding points on the hub and shroud surfaces are written on units 40 and 45. Plot files for graphics post-processing are written on units 50 and 55 for the hub and shroud surfaces.

Folding points are identified where the normalized Jacobian is negative. Sheared points are identified where the normalized Jacobian is less than EPSBAD. An error index is computed every ten iterations and is defined as the relative movement of a point normalized by the diagonal length of the grid. Convergence is attained when the absolute maximum error index among all points is less than EPSCNV with no folding points.

The initial grid binary file is written on FORTRAN unit 10 with the following code:

```fortran
WRITE (10) IMAX, JMAX, KMAX
DO 100 K=1,KMAX
  WRITE (10) ((X(I,J,K), I=1,IMAX), J=1,JMAX),
             ((Y(I,J,K), I=1,IMAX), J=1,JMAX),
             ((Z(I,J,K), I=1,IMAX), J=1,JMAX)
100 CONTINUE
```
WRITE (10) IGTYPE
WRITE (10) ISING1, ISING2, ISING3, ISING4
WRITE (10) JSING1, JSING2, JSING3, JSING4
WRITE (10) FACI1, FACIM
WRITE (10) FACJ1, FACJM
WRITE (10) FACK1, FACKM

The variables in the binary file are defined as follows:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IMAX</td>
<td>Number of grid points in the $\xi_1$ direction.</td>
</tr>
<tr>
<td>JMAX</td>
<td>Number of grid points in the $\xi_2$ direction.</td>
</tr>
<tr>
<td>KMAX</td>
<td>Number of grid points in the $\xi_3$ direction.</td>
</tr>
<tr>
<td>X</td>
<td>Physical Cartesian coordinate in the $x_1$ direction.</td>
</tr>
<tr>
<td>Y</td>
<td>Physical Cartesian coordinate in the $x_2$ direction.</td>
</tr>
<tr>
<td>Z</td>
<td>Physical Cartesian coordinate in the $x_3$ direction.</td>
</tr>
<tr>
<td>IGTYPE = 1</td>
<td>C-grid.</td>
</tr>
<tr>
<td></td>
<td>H-grid.</td>
</tr>
<tr>
<td>ISING1</td>
<td>I-index of first line singularity on the $J = 1$ surface.</td>
</tr>
<tr>
<td>ISING2</td>
<td>I-index of second line singularity on the $J = 1$ surface.</td>
</tr>
<tr>
<td>ISING3</td>
<td>I-index of first line singularity on the $J = \text{JMAX}$ surface.</td>
</tr>
<tr>
<td>ISING4</td>
<td>I-index of second line singularity on the $J = \text{JMAX}$ surface.</td>
</tr>
<tr>
<td>JSING1</td>
<td>J-index of first line singularity on the $I = 1$ surface.</td>
</tr>
<tr>
<td>JSING2</td>
<td>J-index of second line singularity on the $I = 1$ surface.</td>
</tr>
<tr>
<td>JSING3</td>
<td>J-index of first line singularity on the $J = \text{JMAX}$ surface.</td>
</tr>
<tr>
<td>JSING4</td>
<td>J-index of second line singularity on the $J = \text{JMAX}$ surface.</td>
</tr>
<tr>
<td>FACI1</td>
<td>Desired spacing off the $I = 1$ surface.</td>
</tr>
<tr>
<td>FACIM</td>
<td>Desired spacing off the $I = \text{IMAX}$ surface.</td>
</tr>
<tr>
<td>FACJ1</td>
<td>Desired spacing off the $J = 1$ surface.</td>
</tr>
<tr>
<td>FACJM</td>
<td>Desired spacing off the $J = \text{JMAX}$ surface.</td>
</tr>
<tr>
<td>FACK1</td>
<td>Desired spacing off the $K = 1$ surface.</td>
</tr>
<tr>
<td>FACKM</td>
<td>Desired spacing off the $K = \text{KMAX}$ surface.</td>
</tr>
</tbody>
</table>
REFERENCES


(A) C-GRID IN PHYSICAL SPACE.

(B) C-GRID IN COMPUTATIONAL SPACE.

FIGURE 1. C-GRID ABOUT GENERIC BLADE.
Figure 2. - H-grid between generic blades.

Figure 3. - Three-dimensional C-grid for turbine rotor.

Figure 4. - Three-dimensional H-grid for turbine rotor.
FIGURE 5. - FLOW CHART FOR PROGRAM GRID 3D.

1. Check for folding points (Subroutine BCHECK)
   - No
   - Yes
     - Folding points?
       - No
       - Stop
       - Yes
         - Continue?
           - No
           - Stop
           - Yes

2. Linear transfinite interpolation (Subroutine TRANSF)
   - No
   - Yes
     - Check boundary point folding?
       - No
       - Yes

3. Establish reference length scales (Subroutine BOUND)
4. Calculate normal first derivatives (Subroutine CNSTRT)
5. Establish extrapolation scheme (Subroutine EXTRAP)
6. Establish factorization scheme (Subroutine FCTILZ)

FIGURE 5. - CONTINUED.
Establish boundary protection scheme (Subroutine INTERR)

Calculate invariants for each surface (Subroutines PREF1, PREF2, ... PREF6)

Calculate variants for each surface (Subroutines POR1, POR2, ... POR6)

Update interior points using SOR (Subroutine PSOR)

FIGURE 5. - CONTINUED.

Calculate error (Subroutine ERROR)

Check folding and shearing in interior (Subroutine GLOW)

Converged?

Yes

Stop

No

Poisson system?

Yes

Write output (Subroutines OUT40 and OUT50)

No

FIGURE 5. - CONCLUDED.
This report describes a numerical method for generating three-dimensional grids for turbomachinery computational fluid dynamics codes. The basic method is general and involves the solution of a quasi-linear elliptic partial differential equation via pointwise relaxation with a local relaxation factor. It allows specification of the grid point distribution on the boundary surfaces, the grid spacing off the boundary surfaces, and the grid orthogonality at the boundary surfaces. It includes adaptive mechanisms to improve smoothness, orthogonality, and flow resolution in the grid interior. A geometry preprocessor constructs the grid point distributions on the boundary surfaces for general turbomachinery cascades. Representative results are shown for a C-grid and an H-grid for a turbine rotor. Two appendices serve as user’s manuals for the basic solver and the geometry preprocessor.