ELSEVIER

Calculation of flow and heat transfer near the entrance to film-cooling holes using three-dimensional mesh embedding with solution adaptation

B. L. Lapworth and M. G. Rose
Aerospace Group, Rolls-Royce plc, Derby, UK

R. G. D. Nedar
Volvo Aero Corporation, Combuster Divison, Engineering, Trollhätan, Sweden

A three-dimensional (3-D) mesh embedding algorithm for the Navier–Stokes equations is presented. An upwind control volume discretization is used to maintain accuracy and stability at the embedding interfaces. With minor restrictions on cell shape, fluxes are also conserved across the interfaces. The hanging nodes on the interfaces are treated implicitly to avoid internal boundaries in the flow that would degrade the rate of convergence. The embedded mesh solver is used to model 90 and 30° cylindrical film-cooling holes ingesting flow from a parallel-sided duct at low-, medium- and high-suction ratios. Adaptation of the mesh to the flow solution is used to improve the resolution of high-gradient regions. The influences of the inclination angle and suction ratio on the flow and heat transfer in the vicinity of the hole and on the hole discharge coefficient are modelled and compared with reported experimental measurements.

Keywords: film-cooling holes; mesh embedding; solution adaptation; upwind control volumes

Introduction

Turbine entry temperatures (TET) in modern gas turbine engines are typically 150°K above the melting point of the blade material. Consequently, cooling of the blade is necessary if the turbine is to have a useful working life. Arrays of film-cooling holes that release cool air, bled from the compressor, onto the blade surface are commonly used to control surface temperatures and heat fluxes. As turbine designers strive for higher TET, it is ever more important that film-cooling hole designs achieve their target thermal performances.

Numerical modeling plays an important role in film-cooling design through the use of two-dimensional (2-D) or quasi-three-dimensional (3-D) boundary-layer analysis methods (e.g., Norton et al. 1990). These methods are intended for parametric design studies and model only the laterally averaged behaviour of the film-cooling rows. More detailed analyses of the 3-D flow within and around film-cooling holes have concentrated on jets injecting into a cross-flow. Bergeles et al. (1978) modelled a single jet using a semi-elliptic procedure above and immediately downstream of the hole, coupled with a marching procedure elsewhere. The key aspect of their calculations was the use of a nonisotropic form of the high Reynolds number k-ϵ model, which improved the predictions of film-cooling effectiveness. White (1980) modelled a single jet using separate computations in the pipe and cross-flow duct, which were iteratively linked at the jet exit plane. White also used a nonisotropic k-ϵ model to improve the predictions of the velocity profiles downstream of the hole. White, among others, demonstrated the need for an elliptic procedure within and downstream of the hole to model the strong recirculations at higher injection rates. To reduce computational resources, a number of models have used "locally elliptic" procedures in the near-injection region coupled with marching procedures away from the hole (e.g., Demuren et al. 1985; Dibelius et al. 1990). Alvarez et al. (1993) reported fully elliptic calculations for a single jet in cross-flow using both the isotropic k-ϵ model with wall functions and a second-moment closure model. Alvarez concluded that overall both models gave a similar level of agreement with the measurements.

The measurements of Andreopoulus and Rodi (1984) demonstrated the biasing of the jet exit flow to the downstream side of the hole and, hence, the importance of including the pipe in the calculations. Such calculations generally assume either plug or fully developed flow at the pipe entrance. However, in turbine cooling applications, the pipe is fed from internal cooling passages that produce complex flow patterns at the pipe entry.

Address reprint requests to B. L. Lapworth, Rolls-Royce plc, P.O. Box 31, Derby DE24 8BJ, UK.

Received 20 January 1995; accepted 26 October 1995

Int. J. Heat and Fluid Flow 17: 155–166, 1996

655 Avenue of the Americas, New York, NY 10010

0142-727X/96/$15.00
SSDI 0142-727X(95)00099-2

Numerical modelling of the entrance flows to film-cooling holes has received much less attention than has exit flows. Leylek and Zerkle (1993) modelled both the entrance and exit flows to a jet pipe using a plenum to feed the pipe and demonstrated the development of very strong secondary flows within the pipe. The modelling of film-cooling holes on turbine blades also introduces the need for flexible grid systems. Meshes adopting a single structured style throughout the domain incur large overheads in node numbers, because the grid lines needed to resolve the holes extend all the way to the domain boundaries and provide excessive mesh resolution where it is not needed. Benz et al. (1993) modelled two leading-edge film-cooling holes (slots) on a 2-D turbine blade using a multizone grid. H-meshes within the holes were attached to an O-mesh around the blade using periodic boundary conditions. Choi (1993) modelled a row of film-cooling holes on the suction side of a turbine blade using a 3-D slice of the blade that contained a single hole. Separate overlapping meshes were used for the external flow and the flow within and near the hole. The *Chimera* scheme of Steger et al. (1983) was used to communicate information between the meshes.

The present contribution extends the mesh-embedding algorithm of Lapworth (1993) to provide a flexible grid system for film-cooling holes. The grid system uses a single nonoverlapping mesh with internal embedding interfaces that allow a locally fine resolution in the vicinity of the hole. The goal of the numerical model is the calculation of multiple film-cooling holes on 3-D turbine blades, including the internal cooling passages. Mesh embedding is a key tool for keeping node numbers within acceptable computing limits and for ensuring optimal mesh resolution through solution adaptation. However, the resolution of individual holes is likely to remain relatively sparse, and, hence, the high Reynolds number k-ε model with wall functions has been chosen for the numerical model. The authors acknowledge that this model is unlikely to give the most accurate predictions. Our aim is to develop a flexible 3-D mesh-embedding technique and establish whether it provides solutions of sufficient accuracy to be of use in the engineering design environment. The present study considered only isotropic turbulence, the importance of anisotropic diffusivities is recognized and planned for future study. In this paper, mesh embedding is used to model the entrance flows to a single 100-times engine scale hole ingesting air from a parallel-sided duct, as investigated experimentally by Byerley (1989). The flow field and heat transfer are modelled for two hole inclination angles (30 and 90°) and for low, medium, and high suction ratios. The mesh-embedding technique is presented in the next section. This is followed by the numerical model and a discussion of the treatment of the internal embedding interfaces. The application of the numerical model to film-cooling entrance flows is then described. This is followed by the numerical results, which are compared to the experimental measurements. Finally, conclusions are drawn from the study.

Embedded mesh model

Mesh structure

The flow domain is covered with a single, nonoverlapping, curvilinear H-style mesh that contains locally embedded fine mesh regions. The embedded regions are created by a cell subdivision process that divides cells in any of one, two, or three of the mesh directions (Figure 1 shows a cell that has been divided in two of the three mesh directions). Any number of cells can be selected for subdivision, and there is no constraint that the embedded regions form well-defined blocks. Three criteria are available for deciding which cells to subdivide.

(1) *Geometric embedding*: The mesh cells adjacent to a wall can be successively refined in the direction normal to the wall until a desired near-wall spacing is achieved.
(2) *Block embedding*: User-defined regions of the mesh can be refined in one or more directions.
(3) *Solution adaptation*: Cells where some measure of the solution accuracy is greater than a specified threshold are refined in the appropriate direction. The numerical model can adapt

Notation

a	discrete coefficient matrix
b	discrete source term
c_1, c_2, c_μ	coefficients in k-ε equations
$\underline{\underline{C}}^u$	discrete continuity operator
$\underline{\underline{C}}^p$	discrete pressure gradient operator
C_d	hole discharge coefficient
c_p	specific heat at constant pressure
D	film-cooling hole diameter
$d\underline{S}$	area of cell face
G	generation of turbulent kinetic energy
H	enthalpy
I	linear interpolation operator
k	turbulent kinetic energy
L	mixing length
Nu_H	Nusselt number based on duct height
Pr	Prandtl number
P_0	stagnation pressure
p	static pressure
R	gas constant
Re_H	Reynolds number based on duct height
T	temperature
$\underline{u}$	velocity
$\underline{x}$	Cartesian coordinates

Greek

α	hole inclination angle
δ_{ij}	Kronecker delta
ε	turbulent dissipation rate
μ	viscosity
ρ	density
σ	Prandtl number
$\underline{\underline{\tau}}$	viscous shear stress tensor
φ	general transported variable
ΔV	cell volume
Φ	adaptation parameter

Subscripts

cl	centerline value
i,j	tensor components
i	element in discrete difference stencil
l	laminar property
t	turbulent property
$H_1, \ldots, H_8$	hanging node values, Figure 1a
$N_1, \ldots, N_8$	regular node values, Figure 1a
k, ε	property associated with k-ε model
0	stagnation property
1	upstream reference location at duct inlet
2	film-cooling hole exit location

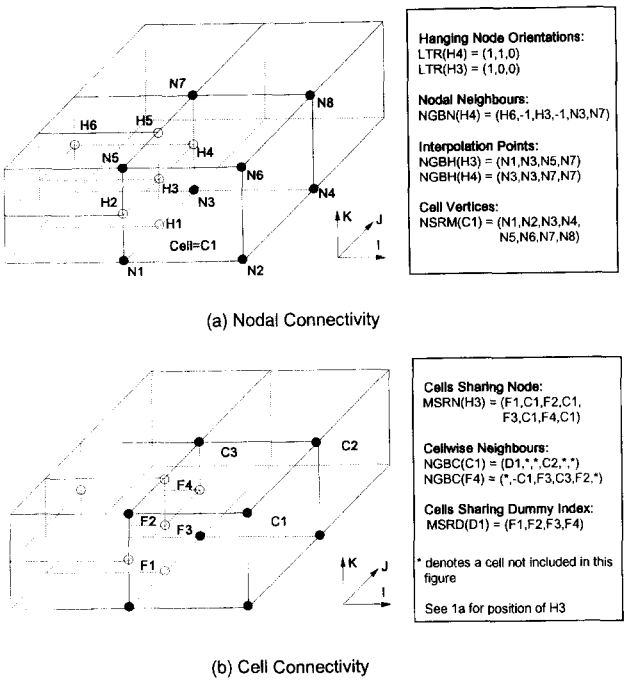


Figure 1 Embedded mesh data structures: (a) nodal connectivity arrays; (b) cell-wise connectivity arrays

on first- and second-order differences (or gradients) of the flow variables. Adaptation may be restricted to cells greater than a given size to avoid excessively fine cells being created; for example, to ensure near wall cells remain within the logarithmic region of the boundary layer.

Mesh embedding produces internal interfaces in the mesh where hanging nodes are used to facilitate the transition between the coarse and fine regions. Values at the hanging nodes are set by linear interpolation, as discussed in the Control volume upwinding . . . section.

Data structure

The cell divisions are applied recursively, beginning with a structured H-mesh that has only a coarse resolution of the flow domain. This gives the embedded mesh a background structure, which means that if all the hanging nodes are eliminated by extending the mesh lines to the domain boundaries, a single structured H-mesh is recovered. This is referred to as the *background fine mesh* and is the mesh that would be needed to perform the computations if the mesh-embedding model were not used. The background fine mesh provides each node in the embedded mesh with a unique i, j, k triplet that identifies its position. The nodes are also given a unique index in a sequential list of all the nodes. It is the latter index that is used to identify the nodes in the discretization scheme — the background i, j, k triplet is only used to compute the mesh connectivity arrays and for diagnostic output.

The background i, j, k triplets by themselves are insufficient to define the mesh connectivity. The necessary additional information is contained in the array $LTR(n)$, which indicates the presence or absence of neighbours to node n in each of the mesh directions: $LTR(n) = (i$-neighbour indicator, j-neighbour indicator, k-neighbour indicator). The neighbour indicators are each set to one of $-1, 0, +1$, with -1 indicating that the neighbour in the $-ve$ mesh direction is absent; $+1$ indicating that the $+ve$ neighbour is absent; and, 0 indicating that both neighbours are present. The mesh structure does not allow a node to have both neighbours absent in the same mesh direction. Typical values of *LTR* at two hanging nodes are shown in Figure 1a. At nonhanging nodes, $LTR = (0, 0, 0)$.

The values of *LTR* and the background (i, j, k) triplet for each node completely define the structure of the embedded mesh — this information together with the node coordinates is all that is passed from the mesh generator to the flow solver. However, to simplify the numerical discretizations, the flow solver generates a set of connectivity arrays. These arrays are:

(1) *NGBN* — list of direct nodal neighbours in each mesh direction;
(2) *NGBH* — list of nodal neighbours for hanging node interpolations;
(3) *NSRM* — list of nodal vertices for each cell;
(4) *NGBC* — list of direct cell-wise neighbours in each mesh direction;
(5) *MSRN* — list of cells touching each node;
(6) *MSRD* — list of cells sharing each dummy cell index; and
(7) *ICFN* — list of nodes in convection–diffusion equation difference stencil.

The first six of these arrays are illustrated in Figures 1a and b. To reduce unstructured array handling, the connectivity arrays have fixed dimensions. For example, the array *NGBN* has storage for six neighbours (two in each direction) at all nodes, independently of how many neighbours are actually present. At a hanging node, *NGBN* is set to -1 at those neighbours that are absent. Typical values of *NGBN* are shown in Figure 1a. At hanging nodes, the most important neighbours are those used to interpolate linearly for the hanging node properties. These are stored in the array *NGBH*. The interpolation uses values at the four vertices of the face for face-centered hanging nodes, and the values at the two ends of the edge for edge-centered nodes. *NGBH* stores four neighbours for all hanging nodes — at edge-centered nodes, each neighbour is stored twice. Hence, the interpolation weights can be set to 0.25 independently of the hanging node position. At edge-centered nodes, the weights automatically increase to 0.5, because each neighbour is counted twice. Typical values of *NGBH* are shown in Figure 1a.

The structure of *NGBH* highlights an important aspect of the embedded-mesh model. Namely, neighbouring cells can only differ by at most one *generation* in the cell division process. Each cell edge or face may contain no more than one hanging node, which must lie strictly at its geometric centre. Because the values at hanging nodes are set by linear interpolation, the properties within a cell on the coarse side of the interface can be obtained solely from the values at the cell's eight vertices using the same linear interpolation rules. The number and location of the hanging nodes is immaterial, and no special interpolation treatments are required. The indices of a cell's eight vertices are stored in the array *NSRM*, as illustrated in Figure 1a. In addition, the *inverse* of *NSRM* is stored in the array *MSRN*, which contains the eight cells that share the given node as a vertex. At nonhanging nodes, the eight cells sharing a node are unique. Hanging nodes are surrounded by fewer than eight cells, with the cells on the coarse side inserted more than once into *MSRN*, as shown in Figure 1b.

To evaluate certain flux contributions, it is necessary to find the cells on either side of a cell face. This information is available from the array *NGBC*, which contains a cell's neighbours in each mesh direction. At an embedding interface, the cell on the coarse side has multiple neighbours on the fine side. However, because *NGBC* can only store one neighbour in each direction, the fine side cells are allocated a dummy index, which is stored in *NGBC*. A further array *MSRD* identifies the fine side cells, which share the dummy index. The dummy indices are easily identified, because they are appended to the list of all cells and, hence, have values greater than the total number of cells. The cells on the fine side of an interface share a unique coarse

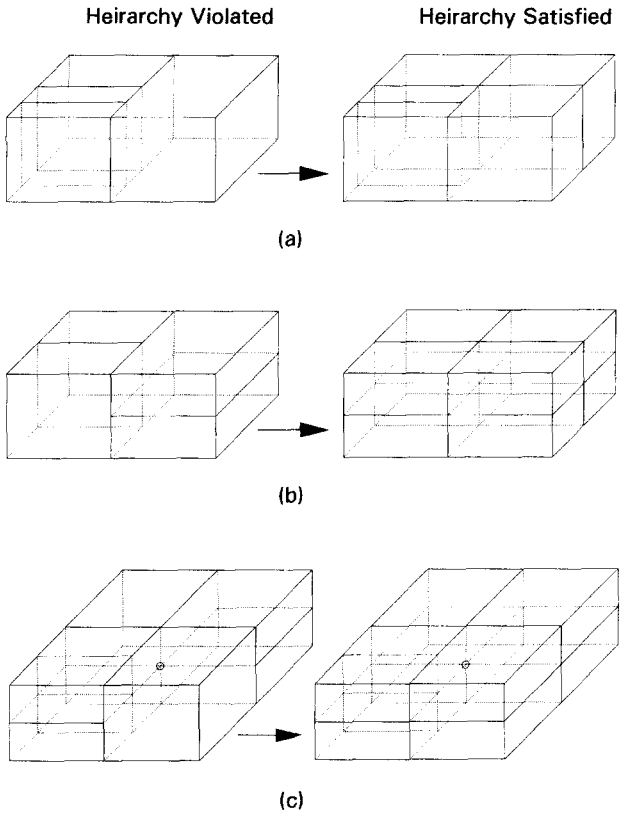


Figure 2 Embedded mesh hierarchy constraints: (a) adjacent cells must differ by no more than one generation; (b) cells cannot be simultaneously on the fine and coarse sides of an interface; (c) nonhanging nodes cannot lie on an interface

side neighbour whose index is multiplied by -1 before it is stored in *NGBC*. At boundary cells that do not have neighbours in a given direction, *NGBC* is set to 0. Hence, *NGBC* contains all the information needed to decide whether a given cell is on the coarse or fine side of an interface or away from any interfaces. Typical values of *NGBC* and *MSRD* are shown in Figure 1b.

The difference stencil for the convection–diffusion operator is stored in the array *ICFN*; this is will be discussed in the Numerical discretization scheme section. The preceding connectivity arrays provide efficient *look-up tables* for the nodal and cellwise indices needed in the discretization scheme. Searching for neighbours is avoided except in the initial calculation of the connectivity arrays. Here, searches are carried out along the mesh lines of the background fine mesh using *LTR* to decide when to terminate a search.

Hierarchical constraints on the cell divisions

The preceding data structure places certain hierarchical constraints on the mesh-embedding algorithm. These are illustrated in Figure 2. The first constraint prevents cells on the fine side of an interface being further divided unless the coarse side cell is also divided (Figure 2a). Otherwise, neighbouring cells would differ by more than one generation. The second constraint prevents neighbouring cells being refined in conjugate directions (Figure 2b). Otherwise, a cell is simultaneously on the fine side of an interface in one orientation and on the coarse side in the conjugate orientation. The third constraint prevents diagonally adjacent cells being refined in the same direction without their neighbours also being refined (Figure 2c). Otherwise, the node highlighted in the figure has a full set of neighbours (*LTR* = (0, 0, 0)) and, hence, is not classed as a hanging node but nevertheless lies on an embedding interface.

Any violations of the hierarchy constraints are removed by first dividing the cell(s) on the coarse side of the interface. Unfortunately, this division may create a further hierarchy violation in the next cell along and so lead to a series of *parasitic* divisions that cascade through the mesh. Experience has shown that the third constraint leads to a particularly high number of parasitic divisions.

Embedded mesh generation

Embedded meshes are generated using the following algorithm.

(1) Define initial mesh, or read previous mesh if performing solution adaptation.
(2) Calculate connectivity arrays.
(3) Select cells for subdivision using the geometric, block, or adaptation criteria.
(4) Select any cells needed to remove hierarchy violations.
(5) Order the selected cells so that coarse cells are always divided before their neighbours.
(6) Subdivide cells, updating *LTR* and the i, j, k triplets as new nodes are added.
(7) Generate new background fine mesh.
(8) Ensure that new nodes on a solid boundary lie on the boundary and blend any changes to a boundary node's position into the surrounding mesh.
(9) Repeat from (2) until all embedding criteria are satisfied.

As cells are divided, either new nodes are created and appended to the list of nodes, or existing nodes are connected to new neighbours. Existing nodes simply have their *LTR* values modified; new nodes must also be assigned an i, j, k index on the background mesh. This may correspond to an index that already exists on the background mesh; otherwise, the index is given a half-integer value so as not to conflict with the indices at any other node. Once the cell divisions have been completed, new lines are inserted into the background mesh at all the half-integer values that have been created. The i, j, k triplets at all nodes are then modified to account for the increased dimensions of the background mesh. The only connectivity array updated until the during the cell divisions is *NGBN*, all other arrays are not updated until the embedding sweep has been completed.

Numerical model

Governing equations

The flow and heat transfer are modelled using the steady Reynolds-averaged conservation equations of mass, momentum, and energy, together with the following perfect gas law:

$$\frac{\partial}{\partial x_j}(\rho u_j) = 0$$

$$\rho u_j \frac{\partial u_i}{\partial x_j} = \frac{\partial \tau_{ij}}{\partial x_j} - \frac{\partial p}{\partial x_j}$$

$$\rho u_j \frac{\partial H}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\frac{\mu_l}{\sigma_l} + \frac{\mu_t}{\sigma_t}\right)\frac{\partial H}{\partial x_j} + \frac{\partial}{\partial x_j}\left[\mu_l\left(1 - \frac{1}{\sigma_l}\right) + \mu_t\left(1 - \frac{1}{\sigma_t}\right)\right]\frac{\partial}{\partial x_j}\left(\frac{u_i u_i}{2c_p}\right)$$

$$p = R\rho T$$

where

$$\tau_{ij} = (\mu_l + \mu_t)\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\delta_{ij}\rho k \qquad (1)$$

The laminar and turbulent Prandtl numbers in the energy equation are taken as 0.7 and 0.9, respectively. The turbulent

viscosity is modelled using the high Reynolds number $k-\varepsilon$ model.

$$\mu_t = c_\mu \rho \frac{k^2}{\varepsilon}$$

$$\rho u_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu_l + \frac{\mu_t}{\sigma_k}\right)\frac{\partial k}{\partial x_j} + G - \rho\varepsilon$$

$$\rho u_j \frac{\partial \varepsilon}{\partial x_j} = \frac{\partial}{\partial x_j}\left(\mu_l + \frac{\mu_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_j} + c_1 G\frac{\varepsilon}{k} - c_2\rho\frac{\varepsilon^2}{k} \qquad (2)$$

where the generation term G is defined as:

$$G = \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\frac{\partial u_i}{\partial x_j}$$

and the k–ε model constants take the standard values:

$c_\mu = 0.09$, $\sigma_k = 1.0$, $\sigma_\varepsilon = 1.3$, $c_1 = 1.44$, $c_2 = 1.92$.

The *log–law* is used in the near-wall regions, whereby the production term in the k equation is evaluated using the logarithmic velocity profile, and ε is set directly from the local equilibrium assumption. This places a constraint on the proximity of the near-wall grid nodes, which should be no closer than $y^+ = 12$. The average y^+ values in each of the present calculations are in the range 25–50.

Control volume assignment

The flow equations are discretized using the upwind control volume algorithm of Moore (1985). In this algorithm, the control volumes allocated to each node are not fixed in advance but are allowed to adjust as the calculation proceeds so that they are aligned as closely as possible with the local flow direction. The alignment is computed by subdividing each computational cell into eight 1/8th subcells and assigning each 1/8th to the cell vertex closest to being in its downstream direction. The division into 1/8th subcells is the method used to perform the finite volume integration and is independent of the mesh-embedding algorithm. Once all the 1/8th cells have been allocated, the control volume for each node is the agglomeration of all the 1/8th cells assigned to it. To illustrate, the case of flow along the grid lines is shown in Figure 3. Here, the downstream vertices in each cell are allocated two 1/8th sub-cells; whereas, none is allocated to the upstream vertices (Figure 3a). Agglomerating the 1/8th cells produces a nodal control volume that has the node at the center of the downstream face (Figure 3b). Over each control

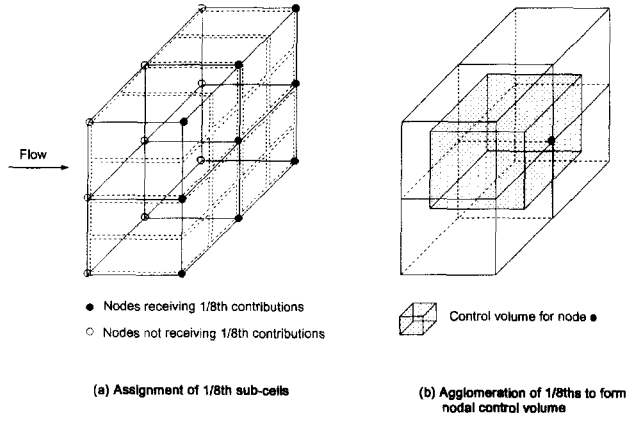


Figure 3 Upwind control volumes for convection-dominated flow along the grid lines

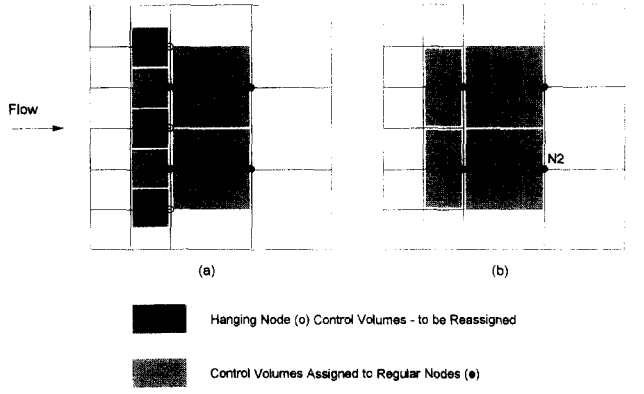


Figure 4 Reassignment of hanging node control volumes for flow from fine to coarse side of an interface: (a) before reassignment; (b) after reassignment

volume the convective fluxes are discretized using second-order accurate central differencing without the need for additional artificial viscosity to stabilize the discretization. (Readers are referred to Moore for fuller details of the control volume upwinding algorithm.)

Control volume upwinding at embedding interfaces

It is important that the interfaces are treated implicitly so that they do not act as internal boundaries and retard the rate of convergence. Stability, accuracy, and flux conservation should also be maintained at the interfaces. These features are achieved by extending the numerical scheme proposed by one of the present authors (Lapworth 1993), which used stacked meshes in the third direction. The rationale of the interface treatment is to ensure that none of the 1/8th subcells is assigned to the hanging nodes. If the hanging nodes do not receive any control volume updates, there is no conflict in setting their values using linear interpolation. To achieve this, any 1/8th subcells that the control volume upwinding algorithm would prefer to assign to the hanging nodes must be reassigned to their neighbours.

Stability is maintained by ensuring the reassignment preserves the upwind bias of the control volumes. Lapworth (1993) has shown that three generic cases arise: flow across an interface from fine side to coarse side; flow across an interface in the opposite sense; and flow along an interface. For brevity, the first of these cases is used to illustrate the interface treatment (see Lapworth for fuller details). The control volumes before and after reassignment are shown in Figure 4. Here, the 1/8th cells are reassigned to their immediate neighbours on the interface to produce pseudo-coarse side control volumes on the fine side of the interface. The interface control volumes are identical to those that would arise if the hanging nodes were not present and the mesh were simply expanding or contracting at this location. This preserves the upwind bias and, hence, stability of the scheme. Accuracy is maintained, because there is no modification to the control volume integrations at the interface — the integrations are simply assigned to a different node.

Fluxes are conserved across an interface if the sum of the fine side fluxes balances the coarse side flux. Considering the interface shown in Figure 1a, the sum of the fine side integrations gives the following:

$$\sum_1^4 \oint_{\substack{\text{fine}\\ \text{face}}} \varphi \, d\underline{S} = \frac{1}{4}(\varphi_{N1} + \varphi_{H1} + \varphi_{H2} + \varphi_{H3})\, d\underline{S}_1 + \frac{1}{4}(\varphi_{N3} + \varphi_{H1} + \varphi_{H3} + \varphi_{H4})\, d\underline{S}_3$$

$$+\frac{1}{4}(\varphi_{N5}+\varphi_{H2}+\varphi_{H3}+\varphi_{H5})\,\mathrm{d}\underline{S}_5$$

$$+\frac{1}{4}(\varphi_{N7}+\varphi_{H5}+\varphi_{H3}+\varphi_{H4})\,\mathrm{d}\underline{S}_7 \qquad (3a)$$

where $d\underline{S}_i$ are the fine side face areas designated according to the nonhanging node they touch. Replacing the hanging node values by their linear interpolation rules leads to:

$$\sum_1^4 \oint_{\substack{\text{fine}\\ \text{face}}} \varphi\,\mathrm{d}\underline{S} = \frac{1}{16}(9d\underline{S}_1+3d\underline{S}_3+3d\underline{S}_5+d\underline{S}_7)\varphi_1$$

$$+\frac{1}{16}(9d\underline{S}_3+3d\underline{S}_1+3d\underline{S}_7+d\underline{S}_5)\varphi_3$$

$$+\frac{1}{16}(9d\underline{S}_5+3d\underline{S}_1+3d\underline{S}_7+d\underline{S}_3)\varphi_5$$

$$+\frac{1}{16}(9d\underline{S}_7+3d\underline{S}_3+3d\underline{S}_5+d\underline{S}_1)\varphi_7 \qquad (3b)$$

Provided the face forms a parallelogram, $d\underline{S}_i = 1/4\,dS_c$ where $d\underline{S}_c$ is the area of the coarse side face, the sum of the fine side fluxes becomes:

$$\sum_1^4 \oint_{\substack{\text{fine}\\ \text{face}}} \varphi\,\mathrm{d}\underline{S} = \frac{1}{4}(\varphi_{N1}+\varphi_{N3}+\varphi_{N5}+\varphi_{N7})\,\mathrm{d}\underline{S}_c = \oint_{\substack{\text{coarse}\\ \text{face}}} \varphi\,\mathrm{d}\underline{S} \qquad (3c)$$

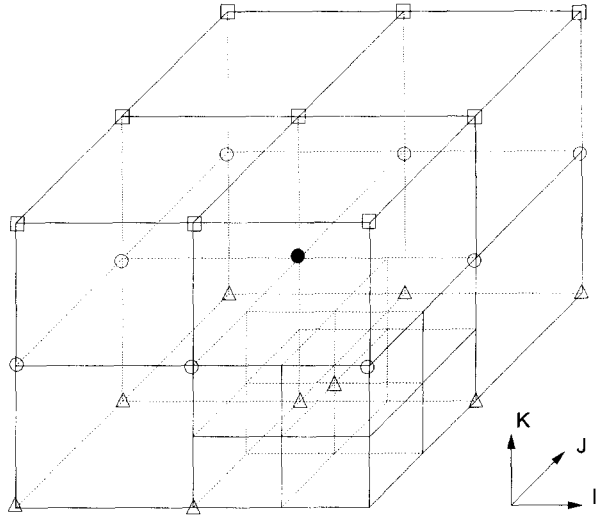
Hence, the convective fluxes are conserved for parallelogram-shaped interfaces. In practice, this is not a severe restriction, because the cells in the initial coarse H-mesh will generally be close to parallelpipeds, and the cell divisions do not change the underlying shape of the parent cell.

The viscous fluxes require derivatives to be evaluated at the cell faces. The nodes used to evaluate these derivatives depend on the configuration of the upwinded control volumes. In particular, the difference stencil for the discrete convection–diffusion operator only uses a node's immediate neighbours (including diagonal neighbours). Hence, considering the control volume for node $N2$ in Figure 4, the viscous derivative on the interface is evaluated using only the nodes $N1$ and $N2$. Otherwise, the difference stencil for node $N2$ would extend outside its immediate neighbours. Node $N1$ can and does use the same evaluation, because $N2$ lies within its difference stencil. Because one-sided differencing is used, the viscous fluxes on an interface are evaluated once and distributed to the control volumes that share the interface. If the differencing is biased toward the coarse side, the fine side fluxes are each set to one-quarter of the coarse side flux. If the bias is toward the fine side, the coarse side flux is the sum of the fine side fluxes. Hence, viscous fluxes are conserved for all shaped interfaces. The biasing of the viscous derivatives is similar to a thin shear layer approximation but without a predetermined bulk flow direction. When the control volume upwinding is along the grid lines, the lateral faces of the control volume lie between the nodes (as shown in Figure 3), and central differencing is used for the viscous derivatives. Similarly, if the flow is along an interface, the interface lies entirely within the interior of a control volume, and fluxes across the interface are not needed. In this case, both the convective and diffusive fluxes are inherently conserved.

Numerical discretization scheme

After integrating over the upwind control volumes, the discrete convection–diffusion equations take the following form:

$$\sum_{i=1}^{27} a_i^{\varphi}\delta\varphi_i = b^{\varphi} \qquad (4)$$



● Centre point of difference stencil

○ Neighbouring points on same k-plane

△ Neighbouring points on lower k-plane

□ Neighbouring points on upper k-plane

Figure 5 Convection–diffusion equation difference stencil in the vicinity of an interface

where φ is the transported property. The coefficients a^{φ} contain the convective and diffusive fluxes, and b^{φ} is the residual of the equation. Away from any interfaces, the i-summation is over the $3\times3\times3$ difference stencil, stored in the connectivity array *ICFN*, surrounding the node to which the control volume is assigned. As shown in Figure 5, the difference stencil at an interface solely relates nonhanging nodes: the hanging node values are implicitly replaced by their linear interpolation rules and do not form part of the stencil. In some circumstances, the interface treatment introduces coefficients that lie outside the *ICFN* stencil. Prior to solving the equations, such coefficients temporarily reside in difference stencils about the hanging nodes. During each internal sweep in the linear solver, the additional coefficients are evaluated and allocated to the appropriate node. Hence, each linear solution solves the complete set of equations, and none of the coefficients is lagged.

The discrete flow equations are iterated toward convergence using the SIMPLER algorithm (Patankar 1980) in which the discrete momentum equations are substituted into the continuity equation in order to form a pressure correction equation used to update the pressure and promote the satisfaction of continuity. The pressure corrections are computed at the cell centres and then interpolated to the nodes to update the pressure. To avoid computing and storing a second large difference stencil, the pressure correction equations are solved using a defect correction procedure:

$$\sum_{i=1}^{7} a_i^p \Delta p_i^n = b^p - \sum_{i=1}^{8} \rho_i \underline{C}_i^u \cdot \delta\underline{u}_i^{n-1}$$

$$\delta p^n = \delta p^{n-1} + \Delta p^n$$

$$\delta\underline{u}^n_{\text{regular}} = \frac{1}{a_c^u}\sum_{i=1}^{8} \underline{C}_i^p \delta p_i^n$$

$$\delta\underline{u}^n_{\text{hanging}} = I_r^h \delta\underline{u}^n_{\text{regular}} \qquad (5)$$

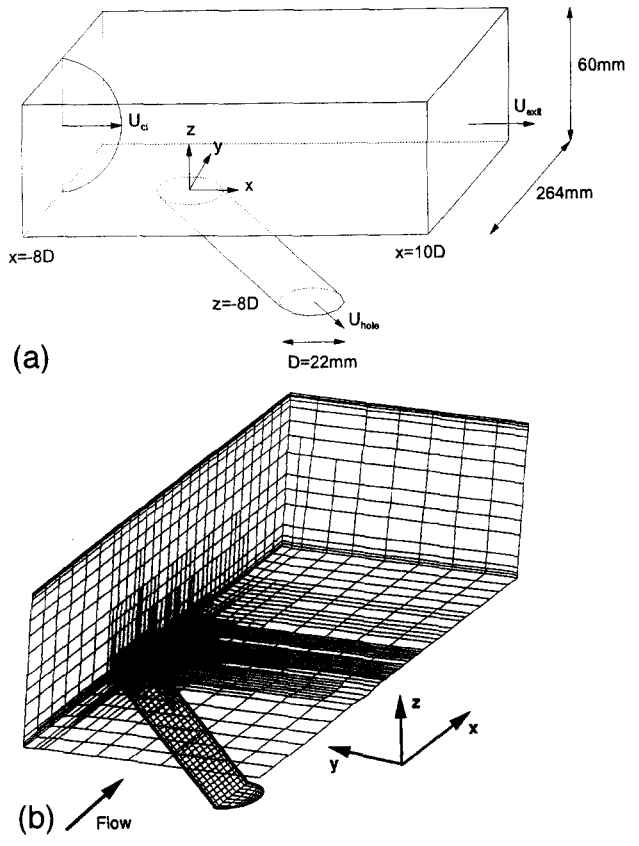


Figure 6 Film-cooling model geometry: (a) schematic of calculation domain; (b) embedded mesh for 30° hole, SR = 4.70, after one solution adaptation

where b^p is the continuity equation residual; $\underline{C}^u$ is the discrete continuity operator with the i-summation over the cell vertices; $\underline{C}^p$ is the discrete pressure gradient operator with the summation over the eight pressure correction points surrounding the node; a_c^u is the centre point coefficient from the momentum equations; and, I_r^h linearly interpolates for the hanging node values. The pressure correction matrix a^p links each cell to its forward and backward neighbours in each of the three coordinate directions (as stored in *NGBC*). For cells on the coarse side of an interface, the multiple fine side neighbours are agglomerated into a single dummy cell whose value is the average of the fine cell values. The advantage of the defect approach is that all cells have eight vertices and all *regular* nodes are surrounded by eight cells whatever the mesh structure. On convergence of the defect system, $\Delta p^n \to 0$ and the pressure and velocity corrections produce a field that satisfies continuity.

Solution of the numerical equations

The discrete equations are solved using the preconditioned *GMRES* algorithm (Saad and Schultz 1986). The preconditioners are tailored to the equations to be solved. The momentum equations are preconditioned using a Gauss–Seidel iteration. The pressure correction equations, which are strongly elliptic in nature, use an incomplete lower–upper factorization. The k-ε equations, which are very strongly coupled through their source terms, are preconditioned using a pointwise coupled Gauss–Seidel iteration. The coupled system is derived by linearising the k-ε equations about the previous solution:

$$\begin{pmatrix} \sum a^k & \rho\Delta V \\ -\rho c_2\left(\frac{\varepsilon}{k}\right)^2\Delta V & \sum a^\varepsilon + 2\rho c_2\frac{\varepsilon}{k}\Delta V \end{pmatrix}\begin{pmatrix}\delta k \\ \delta\varepsilon\end{pmatrix} = \begin{pmatrix} b^k \\ b^\varepsilon \end{pmatrix} \qquad (6)$$

The productive source terms are omitted from the linearization, because they were found to cause poor convergence of *GMRES* and, hence, divergence of the nonlinear iterations. Stability also required the inclusion of the near wall log-law equations in the coupled system.

Application to film-cooling hole entrance flows

The numerical model is used to calculate the film-cooling entrance flows measured by Byerley (1989) using apparatus of approximately 100 times engine scale. The test section consisted of a parallel-sided duct of 60×600 mm cross section with a single hole, of 22-mm diameter, in one side of the duct through which flow was extracted into a plenum. The hole length-to-diameter ratio was 10, compared to typical engine values of 5–10; and the duct-to-hole hydraulic diameter ratio was 5.45, compared to typical engine values of 5–15. Figure 6a shows the modelled section of the apparatus. Numerical results are obtained for hole inclination angles of 30 and 90° and for low, medium, and high suction ratios (SR). Suction ratio is the mean velocity in the hole divided by the peak inlet velocity. The cases modelled are summarized in Table 1.

Boundary conditions

The inlet velocity and temperature profiles are taken from Byerley's (1989) measurements. The profiles are fully developed with peak inlet velocities and temperatures, as shown in Table 1. The inlet profiles of k and ε are set using the assumption of local equilibrium (i.e., turbulence production = turbulence dissipation):

$$\mu_t G = \rho\varepsilon$$

$$\varepsilon = c_\mu^{3/4}\frac{k^{3/2}}{L} \qquad (7)$$

where L is the eddy-viscosity length scale calculated using the mixing length model from Lapworth (1993). G is computed directly from the inlet velocities. Eliminating μ_t using Equation 2 allows k and ε to be evaluated directly.

At the duct and hole exits, the control volume upwinding allows the transported variables to be updated from their difference equations. The suction ratio is controlled by specifying flow

Table 1 Flow conditions and mesh sizes for the numerical test cases

α, °	SR	U_{ci}, m/s	Re_L	T_{ci}, K	T_{wall}, K	Grid size	Rel size *
30	1.85	8.7	25500	350.7	290.5	32472	26%
30	4.70	8.7	25500	350.7	290.5	36555	21%
30	7.68	8.7	25500	350.7	290.5	36555	21%
90	1.20	9.3	25700	363.0	296.0	32941	24%
90	3.47	8.6	27700	332.0	296.0	32068	30%
90	7.48	9.3	26300	364.0	292.0	59456	22%

* Rel size is the ratio of nodes in the embedded mesh to the *background fine mesh*

rates at the exits of the hole and duct. The flow rates are achieved by iteratively updating the exit static pressures by a bulk amount. At the walls the viscous no-slip condition is used. The wall values of k and ε are set to zero, while the near wall values are set consistently with a logarithmic velocity profile. The wall temperature is held fixed at the value shown in Table 1. A symmetry condition is used at the hole centreplane.

Calculation details

The initialization of the flow variables was found to be a crucial element in the stability of the numerical model. The velocities and pressures were initialized separately in the duct and the hole with a linear *ramp* function used to blend the two fields at the hole orifice. In the duct downstream of the hole, the fully developed inlet profiles were retained, but the flow rate was reduced to account for the flow extracted through the hole. Overall continuity balance was an important part of the initialization strategy. The k and ε fields were initialized using the same local equilibrium assumption as used at the duct inlet. Relaxation of the iterative updates was also needed to guarantee convergence. Relaxation factors of 0.6 were used on all the variables for the low and medium SR cases; these were reduced to 0.2 for the high SR cases. The initialization and relaxation strategies allowed convergence to be obtained in all cases without the need to start the calculations for one SR from the solution for a lower SR.

The numerical calculations were declared converged when the rms of the changes in the flow variables between successive iterations were less than 1% of the changes on the first iteration. This typically required between 70 and 100 iterations. In practice, the calculations were run for a fixed number of iterations and then continued using a restart file if the convergence criterion had not been achieved. The adequacy of the convergence criterion was confirmed by performing an extra 50 iterations for the 90° medium SR case. These produced no discernible differences in the numerical solution (the maximum difference in the centreline heat transfer enhancement factors was less then 0.8%). The calculations were performed on a Silicon Graphics Indigo workstation with a R3000 (33 MHz) processor. A typical CPU time was 26.9 hours for 100 iterations on a mesh of 36,555 nodes.

Solution adaptation strategy

After a solution had been obtained on the initial embedded mesh, a solution adaptation was performed and the calculations continued, using the previous solution, until the convergence criterion was again achieved. The adaptation criterion is based on differences in a chosen flow property Φ across each cell, given by (using the notation of Figure 1a):

$$\Delta_i\Phi = \frac{1}{4}(\Phi_{N2} + \Phi_{N4} + \Phi_{N6} + \Phi_{N8} - \Phi_{N1} - \Phi_{N3} - \Phi_{N5} - \Phi_{N7}) \quad (8)$$

where Δ_i represents the difference across the cell in the i-direction, with corresponding definitions for the differences in the j and k directions. The value of $\Delta_i\Phi$ in each cell is then compared with the rms value of $\Delta_i\Phi$ over all the cells.

If $[\Delta_i\Phi/\text{rms}(\Delta_i\Phi) \geq$ tolerance], then the cell is flagged for division in the i-direction. The same test is used to determine whether the cell should also be divided in the j and k directions. The tests in the i, j, and k directions are completely independent. Experience has shown that setting the adaptation tolerance to unity produces acceptable levels of refinement. This value is used in all the present calculations.

In the present study, the adaptation parameter is based on differences in the flow velocity across a cell:

$$\Delta_i\Phi = \left[(\Delta_i u)^2 + (\Delta_i v)^2 + (\Delta_i w)^2\right]^{1/2} \quad (9)$$

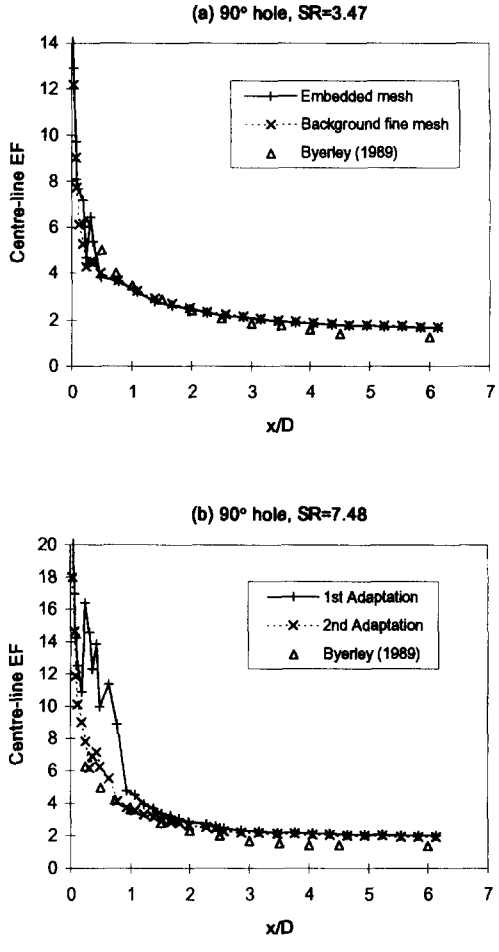


Figure 7 Mesh sensitivity of the centreline heat transfer enhancement factors: (a) comparison of embedded and background fine mesh solutions; (b) comparison of one and two levels of solution adaptation

This parameter is chosen, because it detects large gradients in flow velocity in the vicinity of the film cooling hole. It is also a good sensor for locating and resolving separated and secondary flow regions where changes in velocity magnitude may be less significant than the changes in flow direction.

All the calculations were obtained using a single-solution adaptation, except the 90° high SR case, which used two adaptations. A typical adapted mesh is illustrated in Figure 6b. The mesh sizes for each of the cases modelled are summarized in Table 1. This table also shows the mesh sizes relative to the background fine mesh. Savings in node numbers of between 70 and 80% show the significant benefits that result from mesh embedding. It is noted that Figure 6b shows regions of apparently unnecessary resolution. These are a result of the parasitic subdivisions required by the hierarchy constraints (see the Hierarchical constraints section). The large extent of these subdivisions is an undesirable consequence of the rigid hierarchical structure of the present model. Further work is needed to relax the present constraints and, hence, curtail the number of parasitic subdivisions.

The grid dependencies in the numerical model were assessed by comparing the embedded mesh results for the 90° medium SR case with the results obtained using the background fine mesh. Figure 7a compares the centreline heat transfer enhancement factors from the two calculations. The embedded mesh results show good agreement with the fine mesh results apart from near the stagnation point downstream of the hole. Here, the embedded

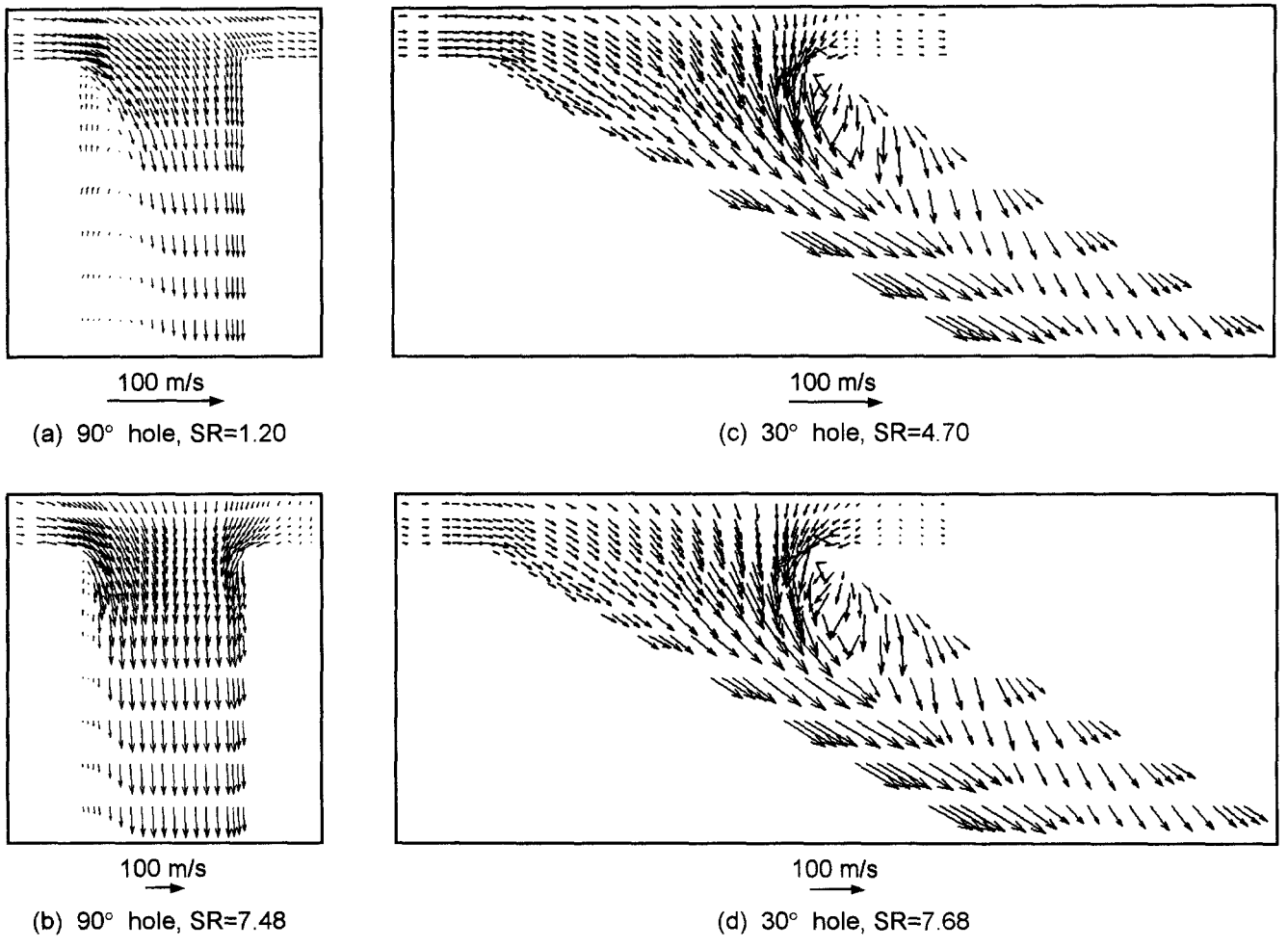


Figure 8 Velocity vectors in duct centreplane: (a) 90° hole, SR = 1.20; (b) 90° hole, SR = 7.48; (c) 30° hole, SR = 4.70; (d) 30° hole, SR = 7.68; vectors interpolated onto a coarser mesh for clarity

mesh results show a small kink in enhancement factor that is probably the result of interfaces, and hence, larger cell sizes, in the vicinity of the stagnation point. However, the kink does not adversely affect the enhancement factors away from the stagnation point, and the mesh dependencies are felt to be acceptable for the present demonstrations. In the 90° high SR case, kinks near the stagnation point are much more pronounced after one solution adaptation than in any other case. To improve the solution, a further adaptation was performed. This dramatically improved the enhancement factors, as shown in Figure 7b. Insufficient computer memory prevented a third adaptation. However, because the enhancement factors were of a similar quality as those shown in Figure 7a, further significant changes in the solution were not anticipated.

Computational results

Flow field

Figure 8 shows the computed velocity vectors on the symmetry plane of the 90° and 30° holes for a range of suction ratios. For the 90° hole, the vectors show how the separation bubble at the upstream entrance to the hole dramatically reduces in size as the SR is increased. The vectors also illustrate the effect SR has on the stagnation point of the streamline separating the flow ingested into the hole and the flow continuing along the duct. At SR = 1.20, the stagnation point is very close to the downstream lip of the hole; at SR = 7.48, the stagnation point is at a position 0.5D downstream of the hole. At SR = 3.47 (not shown in Figure 8), the position of the stagnation point is 0.34D downstream of the hole. These positions show good overall agreement with Byerley's (1989) measurements where the stagnation point remained at the downstream lip of the hole for SRs up to 4.2 and then moved downstream until it reached the 0.5D position at SR = 5.6, where it remained for the higher SRs. For the 30° hole, the stagnation point shows the same trend with increasing SR as the 90° hole.

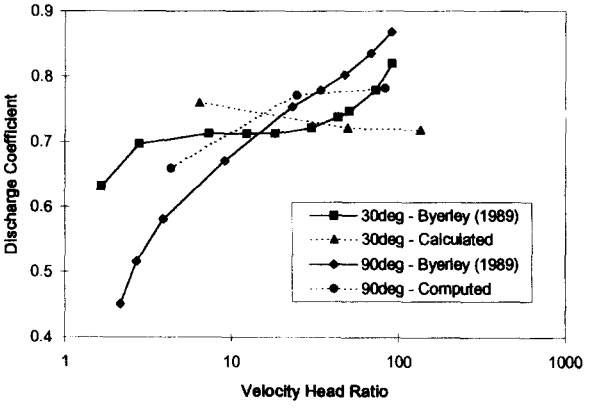
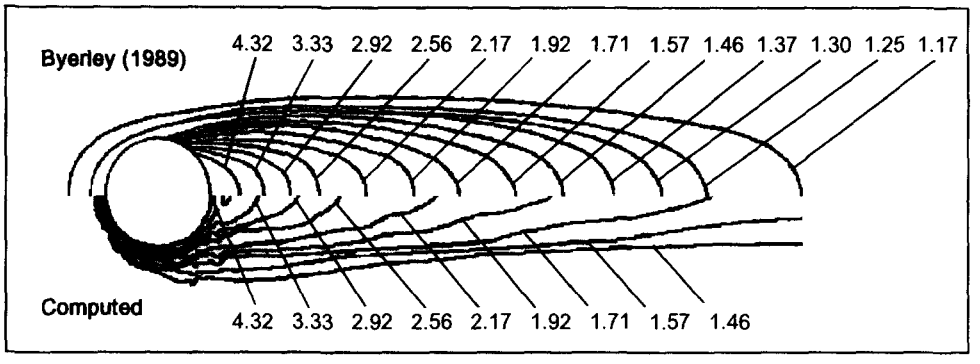


Figure 9 Discharge coefficients for the 30 and 90° holes

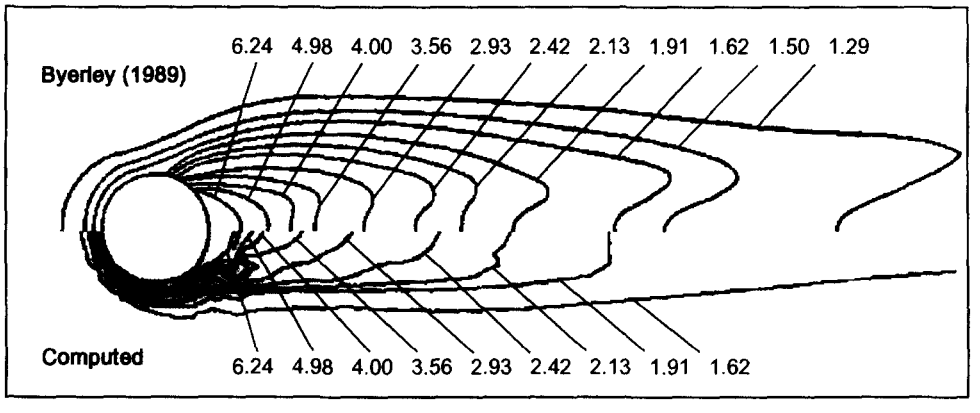
However, at the medium and high SRs (shown in Figure 8), the *reverse* flow entering the hole produces a large separation from the downstream edge compared to the very small separation at the upstream edge. The size of the downstream separation does not change significantly between the medium and high SRs.

Discharge coefficients

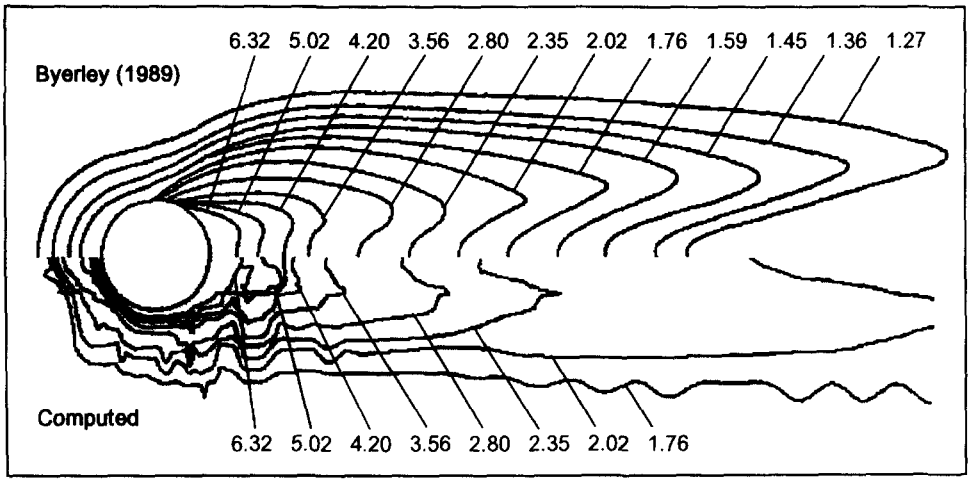
Hole discharge coefficients are an important engineering parameter for the proper sizing and location of film-cooling holes. The discharge coefficient C_d is the ratio of actual mass flow through the hole to the ideal flow that can pass through the hole. The



(a) SR = 1.20



(b) SR = 3.47



(c) SR = 7.48

Figure 10 Contours of heat transfer enhancement factor on the duct endwall for the 90° hole

ideal flow is evaluated using an ideal velocity at the hole exit, which is computed using Bernoulli's equation:

$$u_2^{\text{ideal}} = \left[2\left(P_{02}^{\text{ideal}} - p_2\right)/\rho\right]^{1/2} \tag{10}$$

Where the subscript 2 denotes the hole exit station. For ideal flow, there are no losses in total pressure between the duct inlet station (subscript 1) and the hole exit, hence

$$P_{02}^{\text{ideal}} = P_{01} \tag{11}$$

To compare with Byerley's results, the C_d values are plotted against the *velocity head ratio* defined as:

Velocity head ratio:

$$(P_{01} - p_2)/(P_{01} - p_1) \tag{12}$$

Figure 9 compares the computed C_d values for both the 30 and 90° holes. The computations clearly exhibit the crossover in C_d values observed by Byerley (1989). The numerical results also confirm Byerley's explanation for the crossover in C_d: at the low SR in the 90° hole, there is a large separated region at the upstream entrance that presents a significant blockage and leads to a low C_d value. As SR increases, the separated region reduces in size, and although the flow also separates from the downstream entrance to the hole (as a result of the stagnation point moving downstream), the overall blockage reduces. For the 30° hole at the low SR, the hole is angled favorably into the flow, and there is less blockage than in the 90° low SR case. As the SR increases, the *reverse* flow entering the hole has to turn through 150° and the blockage is greater than for the 90° hole. These effects are clearly shown in the velocity vectors in Figure 8. However, at low SRs, the computed C_d values are of the order of 0.05 larger than the measurements — indicating that the blockage has been underestimated. At high SRs, the computed C_d values are of the order of 0.1 lower than the measurements — indicating that the blockage has been overestimated. For the 30° hole, this leads to the opposite trend to the experimental data, although measurements by Rohde et al. (1969) of a 45° hole show a reduction in C_d between low and medium SRs is possible. The insufficient reduction in the blockage between the medium and high SRs is shown in Figures 8c and 8d and suggests some inadequacies in modelling the separation from the downstream lip of the hole.

Heat transfer

The heat transfer results are presented in terms of an enhancement factor EF, which is defined as the local heat transfer coefficient divided by the base level heat transfer coefficient for fully developed turbulent flow in a channel without film-cooling holes. The base level heat transfer is obtained from the correlation provided by Kays and Crawford (1980):

$$\text{Nu}_H = 0.0146\ \text{Pr}^{0.5}\ \text{Re}_H^{0.8} \tag{13}$$

The Nusselt and Reynolds numbers in this correlation are based on the height of the duct.

Contours of EF on the duct surface for the 90° hole are compared with the measurements in Figure 10. Byerley (1989) observed that the EF contours downstream of the hole change from having a single-lobed nature at low SRs to a double-lobed nature at higher SRs. This is because the downstream enhancement is in part due to the effects of boundary-layer thinning as a result of the downwash from the vortex pair that develop in the downstream duct. At low SRs, the vortices are close to the centre plane, and the downwash occurs in a single area that straddles the centreline. At higher SRs, the distance separating the vortex pair increases and the downwash occurs in two separate areas either side of the centreline leading to higher EFs just away from the

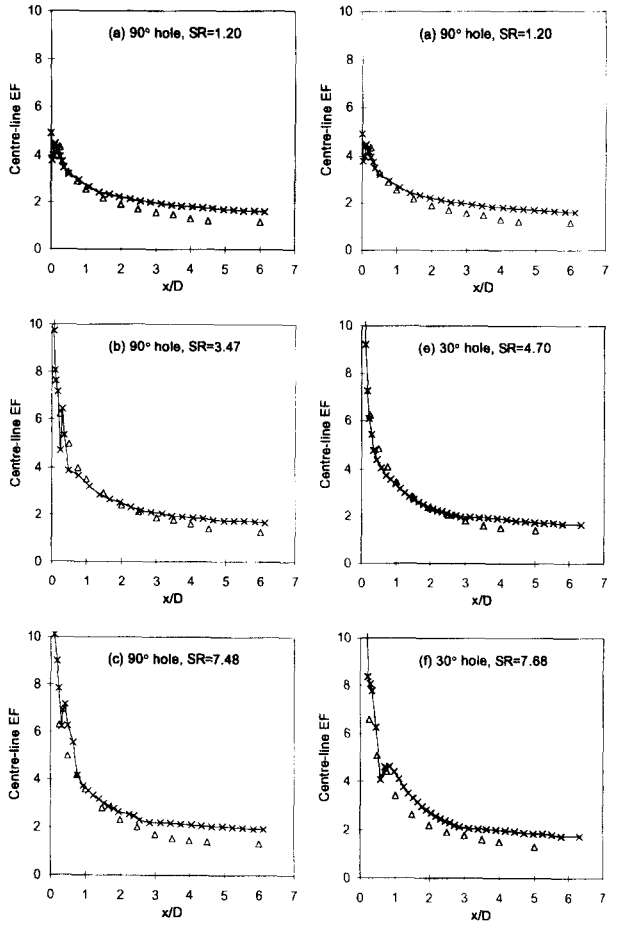


Figure 11 Centreline heat transfer enhancement factors +, computed; Δ, Byerley (1989)

centreline compared to the EF on the centreline. The computed EF contours for the low and high SRs clearly show the transition from a single- to a double-lobed characteristic. The SR = 3.47 EF contours show a small indication of a double-lobed characteristic, although not as pronounced as the measurements. However, this case is close to the SR = 3 value that Byerley (1989) suggested as the crossover point between the single and double-lobed characteristics.

Profiles of EF along the duct centreline for the 90 and 30° holes are compared with the measurements in Figure 11. Overall the agreement with the measurements is acceptable, although the rate of decay of EF downstream of the hole is underestimated. This is probably because of an underprediction of the rate of growth of the new boundary layer downstream of the stagnation point. A difficulty exacerbated by the fact that y^+ values in the downstream region of the duct were often below the cut-off value of 12 for the application of wall functions. The poor modelling of the downstream boundary-layer growth is not unexpected with the high Reynolds number k-ε model, and better accuracy is anticipated with an improved near-wall model. However, the numerical results are sufficiently credible to be of use in the engineering environment.

Conclusions

A three-dimensional (3-D) mesh embedding algorithm with solution adaptation has been described. The algorithm uses upwind

control volumes to maintain accuracy and stability at the internal interfaces. Flux conservation is also maintained, provided the interfaces are parallelograms. A pressure correction solver is used and implicitness of the discrete equations is maintained at the interfaces by replacing hanging node values with their linear interpolation rules. The numerical model allows uni-, bi-, or tri-direction embedding and extends the previous work of Lapworth (1993), which generated 3-D models by stacking topologically similar meshes in the third direction.

The model has been applied to a large-scale apparatus of the entrance flows to a cylindrical film-cooling hole. The model has captured many of the salient features of the flow and heat transfer, such as the downstream movement of the stagnation point and the transition to a double-lobed nature of the EF contours as the SR increases. Solution adaptation has been shown to give significant improvements in solution quality in the vicinity of the hole entrance and to be an important tool for controlling spurious numerical oscillations. Overall the numerical model is felt to provide solutions of sufficient accuracy to be of use in the engineering environment. Detailed comparisons have shown some limitations-caused by the use of a high Reynolds number k-ε model. Further limitations may also be also be ascribed to the use of H-style meshes, which produce distorted grid distributions within the hole. Although the embedded mesh model could equally well be applied to cylindrical polar meshes, which would be better suited to film-cooling holes, the aim of this work is to develop a model that can be easily incorporated into calculations for the flow over turbine blades. In the single nonoverlapping mesh structure of the present model, H-meshes allow easier generalization to multiple holes and to turbine blading applications. The results of this study have shown that a relatively coarse representation of the cooling holes can produce solutions of engineering usefulness. The reductions in node numbers of between 70 and 80%, compared to the *background fine meshes*, produced by the mesh embedding demonstrate the flexibility of the numerical model and are essential for the analysis of multiple film-cooling holes.

Acknowledgments

The authors gratefully acknowledge the permission of Rolls-Royce plc to publish this paper. R. Nedar thanks Volvo Aero Corporation for the opportunity to collaborate with Rolls-Royce on the computations reported herein.

References

Alvarez, J., Jones, W. P. and Seoud, R. 1993. Predictions of momentum and scalar fields in a jet in cross-flow using first and second order turbulence closures. Proc. *AGARD Meeting on Computational and Experimental Assessment of Jets in Cross Flow*, (AGARD CP-534, Paper 24)

Andreopoulus, J. and Rodi, W. 1984. Experimental investigation of jets in a crossflow. *J. Fluid Mech.*, **138**, 93

Benz, E., Wittig, S., Beeck, A. and Fottner, L. 1993. Analysis of cooling jets near the leading edge of turbine blades. *Proc. AGARD Meeting on Computational and Experimental Assessment of Jets in Cross Flow*, (AGARD CP-534, Paper 37)

Bergeles, G., Gosman, A. D. and Launder, B. E. 1978. The turbulent jet in a cross stream at low injection rates: A three-dimensional numerical treatment. *Num. Heat Transfer*, **1**, 217–242.

Byerley, A. R. 1989. Heat transfer near the entrance to a film cooling hole in a gas turbine blade. D. Phil. thesis, Department of Engineering Science, University of Oxford, Oxford, UK

Choi, D. 1993. A Navier–Stokes analysis of film cooling in a turbine blade. *Proc. 31st Aerospace Sciences Meeting and Exhibit*, (AIAA Paper No. AIAA-93-0158), Reno, NV, USA

Demuren, A. O., Rodi, W. and Schonung, B. 1985. Systematic study of film cooling with a three-dimensional calculation procedure. *Proc. ASME Gas Turbine Symposium and Exposition*, (ASME Paper No. 85-GT-2), Beijing, China

Dibelius, G. H., Pitt, R. and Wen, B. 1990. Numerical prediction of film cooling effectiveness and the associated aerodynamic losses with a three-dimensional calculation procedure. *Proc. ASME Gas Turbine and Aeroengine Congress and Exposition*, (ASME Paper No. 90-GT-226), Brussels, Belgium

Lapworth, B. L. 1993. Three-dimensional mesh, embedding for the Navier–Stokes equations using upwind control volumes. *Int. J. Num. Meth. Fluids*, **17**, 195–220

Leylek, J. H. and Zerkle, R. D. 1993. Discrete-jet film cooling: A comparison of computational results with experiments. *Proc. ASME Gas Turbine and Aeroengine Congress and Exposition*, (ASME Paper 93-GT-207), Cincinnati, OH, USA

Moore, J. G. 1985. Calculation of 3-D flow without numerical mixing. AGARD Lecture Series No. 140 on 3-D Computational Techniques Applied to Internal Flows in Propulsion Systems

Norton, R. J. G., Epstein, A. H., Forest, A. E., White, A. J., Henshaw, D. G., Schultz, D. L. and Oldfield, M. L. G. 1990. Turbine cooling system design — Vol. 1, Technical report. Wright Research and Development Center Report No. WRDC-TR-89-2109, Vol. 1

Patankar, S. V. 1980. *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill, New York

Rohde, J. E., Richards, H. T. and Metger, G. W. 1969. Discharge coefficients for thick plate orifices with approach flow perpendicular and inclined to the orifice axis, NASA TND-5467

Saad, Y. and Schultz, M. H. 1986. GMRES: A generalised minimal residual algorithm for solving non-symmetric linear systems. *SIAM J. Sci. Stat. Comput.*, **7**, 856–869

Steger, J. L., Dougherty, F. C. and Benek, J. A. 1983. A chimera grid scheme. *Proc. ASME Minisymposium on Advances in Grid Genearation*, Houston, TX, USA

White, A. J. 1980. The prediction of the flow and heat transfer in the vicinity of a jet in crossflow. *Proc. ASME Winter Annual Meeting*, (ASME 80-WA/HT-26) Chicago, IL, USA