AN INVESTIGATION INTO GRID PATCHING TECHNIQUES*

C. R. Forsey, M. G. Edwards, and M. P. Carr

Aircraft Research Association
Manton Lane
Bedford, England

ABSTRACT

In the past decade significant advances have been made using flow field methods in the calculation of external transonic flows over aerodynamic configurations. It is now possible to calculate inviscid transonic flow over three-dimensional configurations by solving the potential equation. However, with the exception of the Transonic Small Disturbance methods which have the advantage of a simple cartesian grid, the configurations over which it is possible to calculate such flows are relatively simple (eg wing plus fuselage). The major reason for this is the difficulty of producing compatibility between grid generation and flow equation solutions. The main programs in use, eg Jameson in US and Forsey in UK, use essentially analytic transformations for prescribed configurations and, as such, are not easy to extend. Whilst there is work in progress to extend this type of system to a limited extent, our longer term effort is directed towards a more general approach. This approach should not be restricted to producing grid systems in isolation but rather a consideration of the overall problem of flow field solution.

This paper describes one approach to this problem.

*This work has been carried out with the support of Procurement Executive, Ministry of Defence.

GRID GENERATION

GENERAL APPROACH

1. Grid generation, or equivalent, vital to solution of general flow field problems.

2. It is not obvious which technique to use.

3. Various methods being explored

   a) Non-aligned grid — Catherall R.A.E

   b) Aligned grid with global solution for grid with control function — Roberts British Aerospace

   c) Aligned grid with local grids patched together — Forsey A.R.A

FIGURE 1 GENERAL APPROACH TO GRID GENERATION

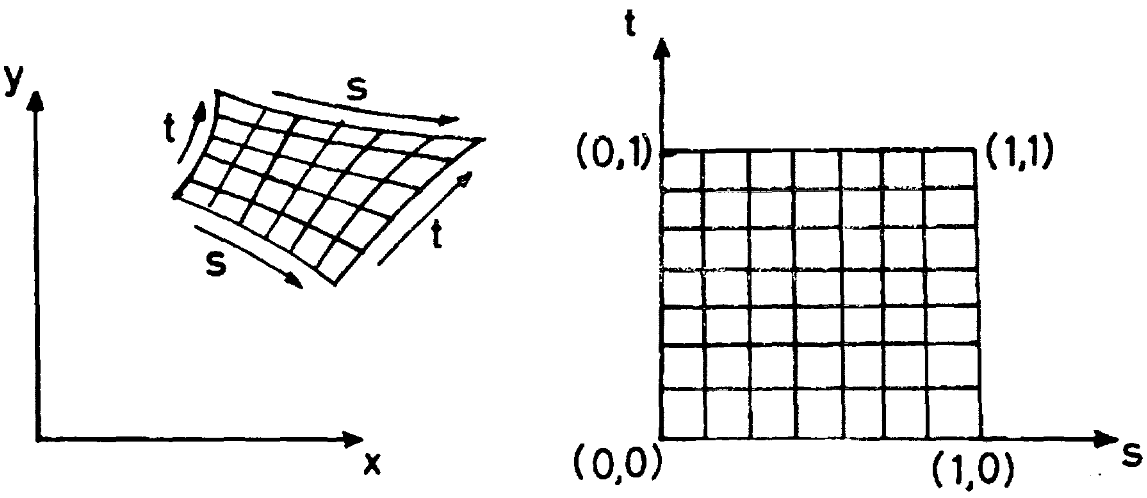
1. GENERAL APPROACH TO GRID GENERATION

Accepting the requirement to solve a set of discretised partial differential equations on the nodes of a suitable grid, then some method of grid generation is vital. Since there are a number of ways in which one can tackle this problem and, at the moment, no one technique appears sufficiently superior to others, it would appear judicious to attempt more than one approach. Therefore, a program of work is being undertaken in various UK establishments to investigate suitable techniques and the method described in this paper is part of this overall project.

Probably the first question one poses when considering the requirement of a computing grid is whether or not to align the grid with the surface. Catherall at RAE is investigating the non-aligned grid concept. The grid, being cartesian, can be generated in a straight forward manner with the major problems being the complicated application of boundary conditions and the general 'housekeeping' for complex configurations. However, extra components can be added fairly easily and it should be versatile.

If an aligned grid is considered mandatory then the application of boundary conditions becomes much simpler and grid generation becomes a major problem. Roberts at British Aerospace is attempting to produce a method of grid generation for general three-dimensional configurations by producing a global solution of a set of partial differential equations. The introduction of mapping singularities is used to control the distribution of grid points using discretisation based on triquintic splines.

The work described here investigates a method some way between these two techniques. The requirement for an aligned grid system is accepted but with the flow field divided into segments, each segment with its own rather straight forward grid system. The surface boundary conditions are easy to apply but the main problem is one of solving the flow equations through the boundaries where the segments are patched together. This approach will now be described in more detail.



$$f(s,t)=(1-s)f(0,t)+sf(1,t)+(1-t)f(s,0)+tf(s,1)$$
$$-(1-s)(1-t)f(0,0)-(1-s)tf(0,1)-s(1-t)f(1,0)-stf(1,1)$$

where $f(s,t)=x(s,t)$ or $y(s,t)$

and $f(s,0)$, $f(s,1)$, $f(0,t)$, $f(1,t)$

define the four sides as parametric functions of s and t .

FIGURE 2 BASIC ISOPARAMETRIC MAPPING

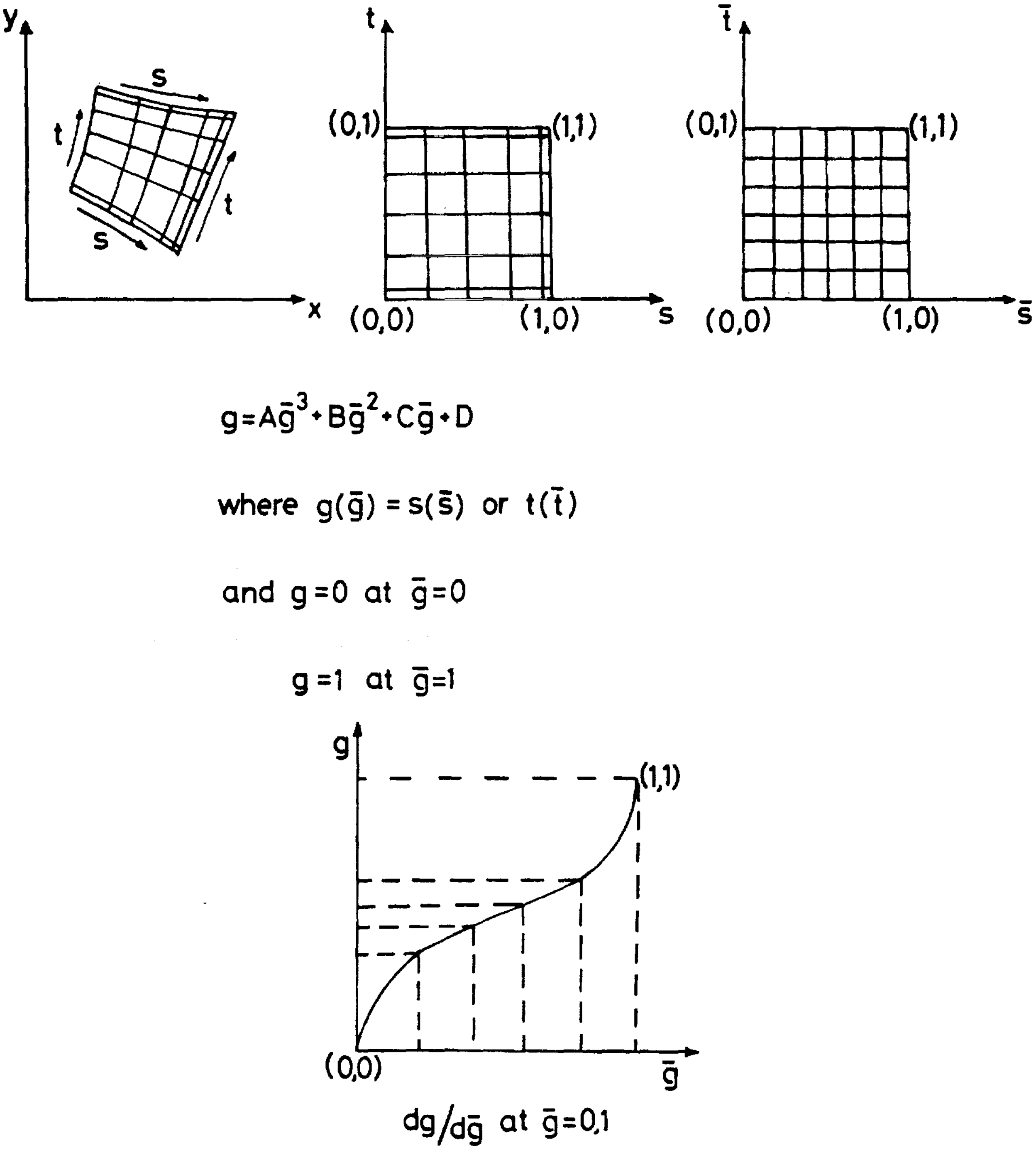
2. BASIC ISOPARAMETRIC MAPPING

The patching technique consists of splitting up regions of interest into a series of quadrilateral segments which are patched together across common boundaries. For the grids in each segment any convenient method of grid generation may be used. However, as it is necessary to match the grids across common boundaries and to maintain some control over the grid spacing in these regions, which is most conveniently done using interactive graphics, a grid generation technique combining simplicity with minimal computer requirements is needed.

One such method, which has been used extensively in finite element work, is the isoparametric or blending function method. This method consists of defining the x and y coordinates of points within a quadrilateral as parametric functions (f) of two parameters (s,t) where s = 0,1 and t = 0,1 define the sides of the quadrilateral in parametric space. If the values of f are defined along s = 0,1 and t = 0,1 (ie the point distributions along the sides are prescribed) then the blending function f defines internal points as a smooth blending between these boundary values. Taking equal intervals in s and t then defines the grid lines within the quadrilateral.

The values of f along s = 0,1 and t = 0,1 are defined by cubic spline curve fits of f vs.s or t where s and t are taken as the arc lengths along the appropriate sides.

The blending function used in the present patching method is the lowest order blending function which is a bilinear blending. However, higher order blendings (eg cubics) could be used with very little increase in computing time and the extra degrees of freedom then used to define the shape of some of the internal grid lines or the slope of the grid lines at the boundaries.



$$g = A\bar{g}^3 + B\bar{g}^2 + C\bar{g} + D$$

where $g(\bar{g}) = s(\bar{s})$ or $t(\bar{t})$

and $g = 0$ at $\bar{g} = 0$

$g = 1$ at $\bar{g} = 1$

$dg/d\bar{g}$ at $\bar{g} = 0{,}1$

specified by user

FIGURE 3 STRETCHED ISOPARAMETRIC MAPPING

3. STRETCHED ISOPARAMETRIC MAPPING

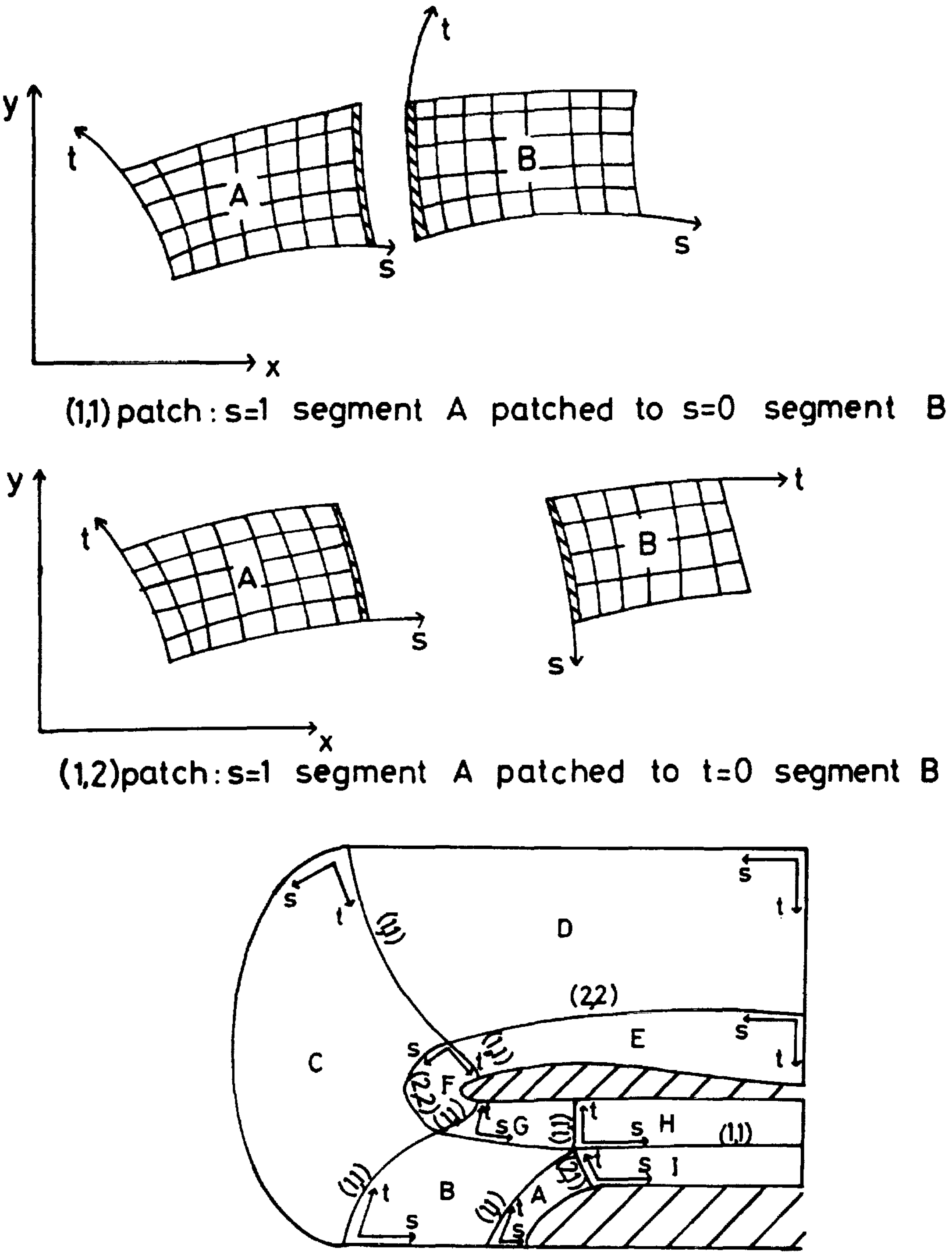
The basic isoparametric mapping produces grids which vary smoothly between opposing sides of the quadrilateral. However, for equally spaced intervals in parametric (s,t) space the corresponding grid lines in physical (x,y) space, although curvilinear, are still equally spaced. It is convenient to have the facility for packing grid lines near specific quadrilateral boundaries or near the middle of the quadrilaterals.

Hence, preliminary stretching transformations are applied to the s and t coordinates. One simple stretching which gives considerable user control is to make s and t cubic functions of some other parameters $\bar{s}$ and $\bar{t}$. Taking equal intervals in $\bar{s}$ and $\bar{t}$ then results in unequally spaced intervals in s and t. By appropriate choice of the derivatives $ds/d\bar{s}$ and $dt/d\bar{t}$ at each end of the cubic it is possible to pack points towards either end (one value of $ds/d\bar{s} < 1$, the other value > 1), towards the middle (both values of $ds/d\bar{s} > 1$), or towards both ends (both values of $ds/d\bar{s} < 1$).

For all cases except the last a single cubic appears adequate. In the last case, however, attempting to pack points towards both ends usually results in a grid which has very fine spacing near both ends but which then suddenly jumps to much wider spacing near the middle. This seems to be due to an inability to control the slope of the cubic near the middle where the slope remains much the same regardless of the slopes imposed at each end. One solution which we are currently using for this case is to replace the single cubic by a cubic spline curve through 4 points. The slopes are still specified at the end pair of points and the middle pair of points are chosen to control the slope of the curve near the middle.

In practice, the stretching parameters (ie $ds/d\bar{s}$ and $dt/d\bar{t}$ at each end plus the two middle points for the cubic spline stretching) are chosen interactively by the user with the aid of interactive graphics.

In order to increase flexibility still further different values of the stretching parameters can be specified on opposing sides of the quadrilateral and a linear variation between these values is used for all internal grid lines between these two sides.



(1,1) patch : s=1 segment A patched to s=0 segment B

(1,2) patch : s=1 segment A patched to t=0 segment B

Intake patching schematic

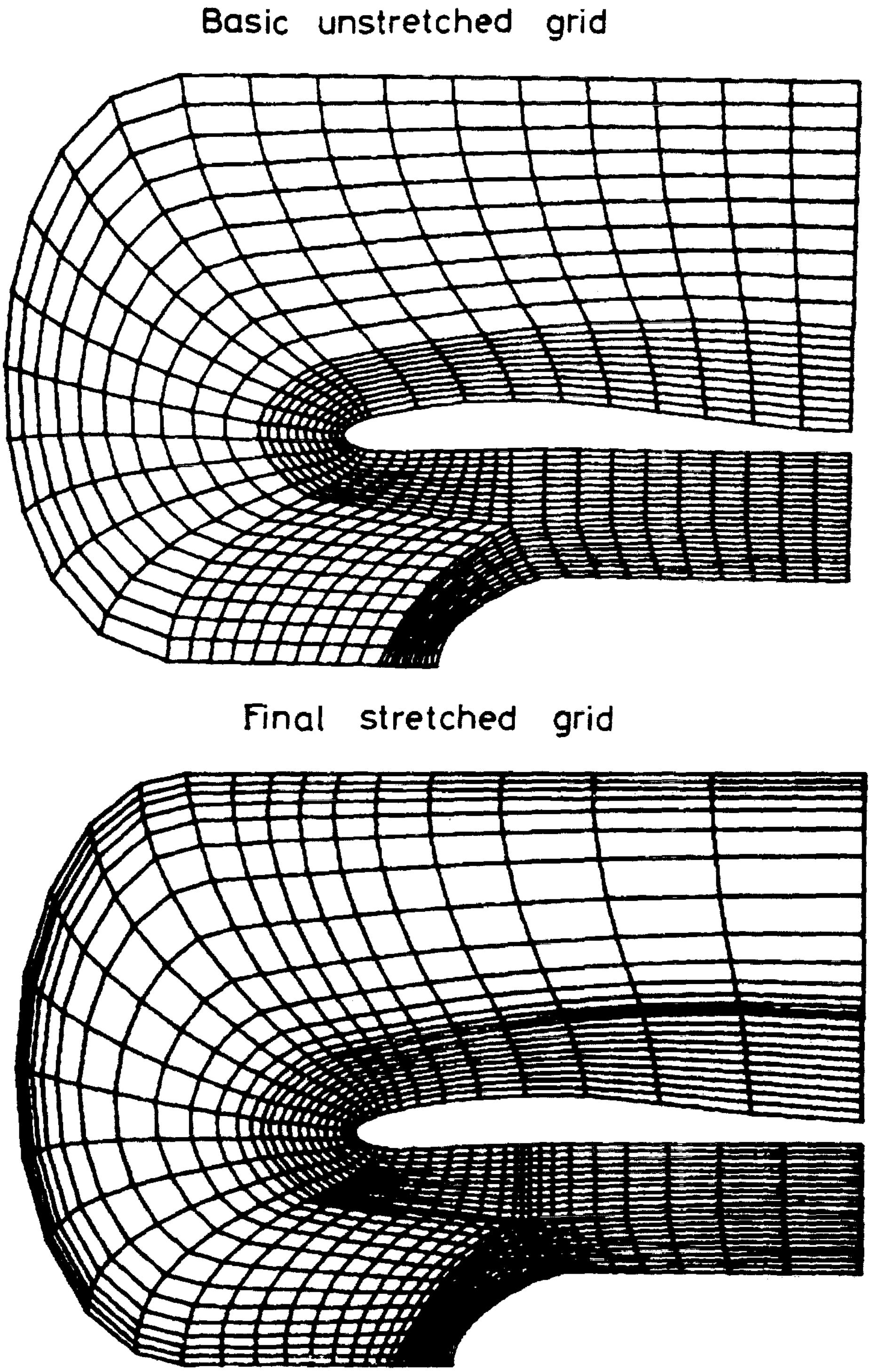
FIGURE 4 PATCHING SCHEMATIC

4. PATCHING SCHEMATIC

Individual segments, each with their own local isoparametric grids are joined (or 'patched') along common boundaries until the whole region of interest has been covered. Grid lines in two adjoining segments must meet on their common boundary which implies that the same number of points and the same stretching functions are used on both sides of this boundary. However, the grid lines may change direction through this boundary, i.e. they need not be smooth. Instead, special boundary conditions are applied on patched boundaries to ensure flow continuity. Sides of segments corresponding to solid surfaces or free stream conditions also have appropriate boundary conditions applied.

Since maximum flexibility is required when choosing the way the region of interest is split up into segments, it is necessary to allow any side of one segment to patch to any side of an adjoining segment. Two typical patches, designated (1,1) patches and (1,2) patches are illustrated and there are several others. In principle, different types of patch should not significantly increase the difficulty of applying the appropriate patch boundary conditions. However, in practice they considerably increase the general program housekeeping needed and in the current program not all types of patches have been allowed for as yet.

At present the way the region is divided into segments is controlled by user input although eventually it is hoped to (at least partially) automate this process. Initially, all solid surfaces (eg aerofoil surfaces etc) are defined accurately and then the user defines the segment boundaries, some of which are parts of solid surfaces and some separate hand drawn curves. A schematic showing a typical setting up procedure for an inlet with central bullet is shown. The orientations of each segment and the type of each patch are indicated on the schematic.



Basic unstretched grid

Final stretched grid

FIGURE 5 EXAMPLE OF GRID-INTAKE

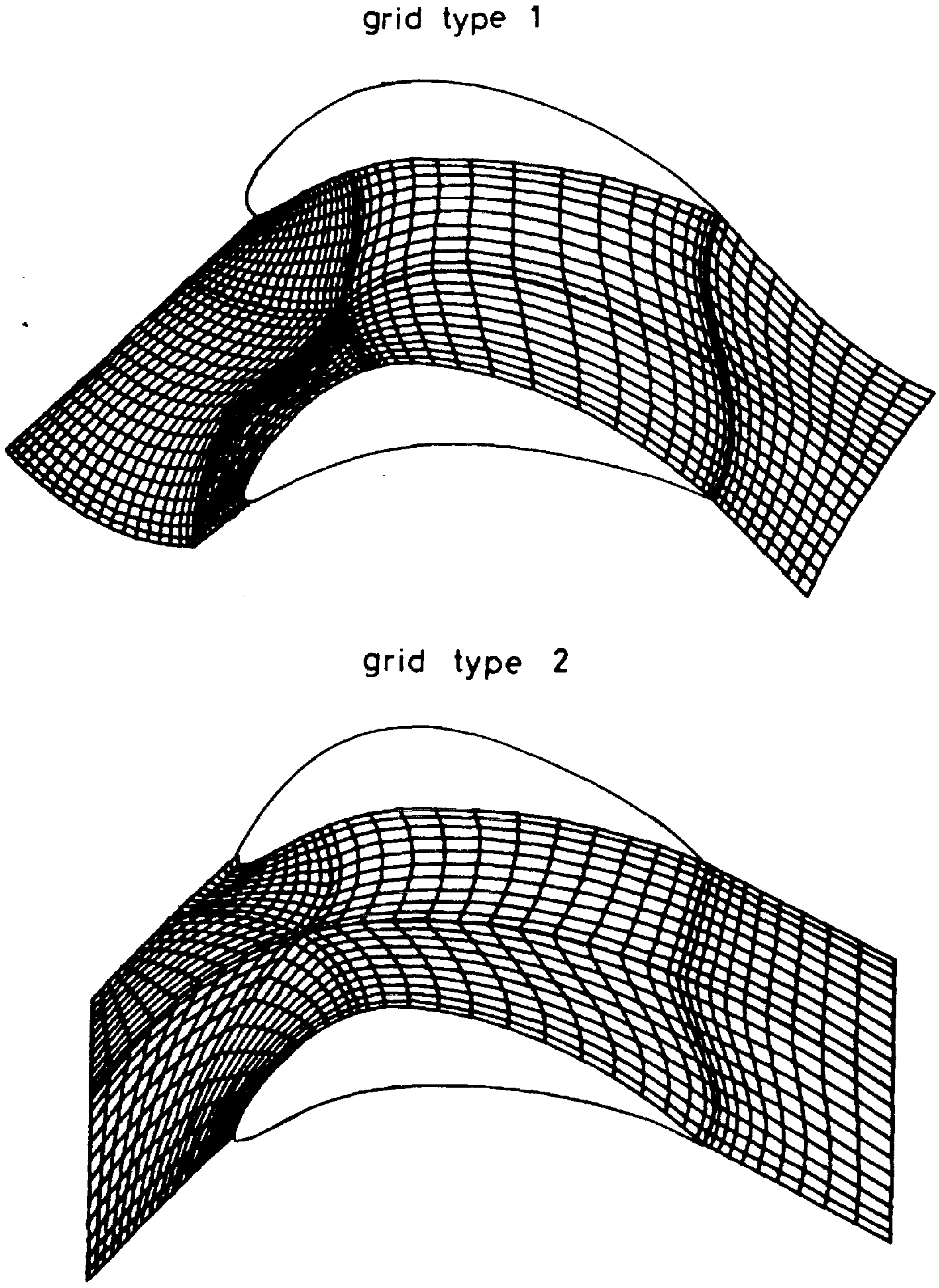
5. EXAMPLE OF GRID - INTAKE

When the region has been divided into segments to the user's satisfaction, the next step is to define the number of points in the s and t directions and the corresponding unstretched grids in each segment. This gives a general idea of what the final overall grid will be like. An example is given at the top of Fig 5 for the intake with central bullet.

Various stretchings are then applied to the s and t coordinates in each segment to remove sudden changes in the width of grid intervals (particularly across patched boundaries) and to pack grid lines in regions where the flow is expected to vary rapidly. The stretching parameters are modified, and the resulting grids displayed, interactively using a graphics terminal. The final overall grid is shown at the bottom of Fig 5. Approximately two days work was required to produce this grid from scratch and only a small amount of computer time was required on a modest Prime 400 computer linked to a Tektronix 4051 graphics terminal.

A few comments regarding the choice of segments for this example may be useful. Because of the nature of the blending functions used, the easiest way to ensure that grid lines are approximately normal to solid surfaces is to choose the shape of the patch boundaries which join such surfaces to be nearly normal to the surfaces concerned as has been done in segments A and F. Furthermore, in order to accurately model the cowl surface boundary condition and the channel flow between the cowl and the bullet, a fine inner grid is patched to a sparse outer grid but stretchings are used to ensure that a sudden change in grid spacing does not occur at the patch boundaries.

It will be noticed that at one point five patch boundaries (ie grid lines) meet rather than the usual four. We feel that this should give no particular problems especially if the point is in a region where there are no flow singularities. (There is some evidence that putting such a point at a stagnation point of the flow can lead to difficulties).



grid type 1

grid type 2

FIGURE 6 EXAMPLE OF GRID - CASCADE

6. EXAMPLE OF GRID - CASCADE

The region of interest may be divided into segments, under user control, in any number of ways and Fig 6 illustrates two different grids for a typical cascade flow problem. The generating segments are marked on each grid. In the two cases different numbers of segments are used with individual segments having quite different shapes. Although there is no restriction, in principle, on the number of segments which may be used, there is some evidence that, at the current stage of development, the convergence rate for the solution of the flow equation decreases with increasing numbers of segments.

The top grid shows the first attempt at a cascade grid. In this attempt the main criterion in choosing the segments was to produce a fine grid spacing around the leading edges of the two aerofoils, a region where many previous cascade grids have been deficient. Again, notice that at one point five patch boundaries (ie grid lines) meet rather than the more usual four.

However, after producing this grid it was realised that it would be impossible to use periodic boundary conditions across the two lines upstream of the leading edges of the two aerofoils without some form of interpolation. This was because the upper and lower lines are formed by different combinations of segment boundaries and hence have different numbers of points and different point spacings. The same argument applies to the two lines downstream of the trailing edge of the two aerofoils. Since neither pair of lines is intended to represent actual streamlines, periodic boundary conditions are the only correct boundary conditions which may be applied across these lines. Hence, the lower grid was produced to try and overcome this restriction without significantly compromising the other advantages of the first grid.

This perhaps illustrates the interactions between grid generation methods and flow calculation methods as the two cannot really be studied independently.

GENERAL TENSOR FORM OF THE FULL POTENTIAL EQUATION

$$(c^2-q^2)\,v^i v^j \frac{\partial^2 \phi}{\partial r^i \partial r^j} + c^2(q^2 g^{ij} - v^i v^j)\frac{\partial^2 \phi}{\partial r^i \partial r^j} + \frac{c^2 q^2}{\sqrt{g}}\frac{\partial \phi}{\partial r^j}\frac{\partial}{\partial r^i}(\sqrt{g}\,g^{ij})$$

$$-\frac{q^2}{2} v^i v_j v_k \frac{\partial g^{jk}}{\partial r^i} - q^2 v^i v^j \frac{\partial^2 x^1}{\partial r^i \partial r^j} = 0$$

r^i = coordinates in transform space ($r^1 = r$, $r^2 = s$, $r^3 = t$)

ϕ = perturbation velocity potential ($\Phi = \phi + x$)

$c^2 = \frac{1}{M_\infty^2} + \frac{(\gamma-1)}{2}(1-q^2)$ = square of local speed of sound

g = determinant (g_{ij})

g^{ij} = cofactor (g_{ij}) / determinant (g_{ij})

$g_{ij} = \frac{\partial x^k}{\partial r^i} \cdot \frac{\partial x^k}{\partial r^j}$ = metric tensor

x^k = cartesian coordinates in physical space ($x^1 = x$, $x^2 = y$, $x^3 = z$)

$v^i = g^{ij} v_j$ = contravariant velocity

$v^i = \frac{\partial \Phi}{\partial r^i}$ = covariant velocity

$q^2 = v^i v_i$ = total velocity

BOUNDARY CONDITIONS

1. Free stream boundary condition
   a) v_i = free stream velocity in i direction
      or
   b) $\phi = 0$
2. Solid surface boundary condition
      $v^i = 0$
3. Patch boundary condition

FIGURE 7 FLOW EQUATION

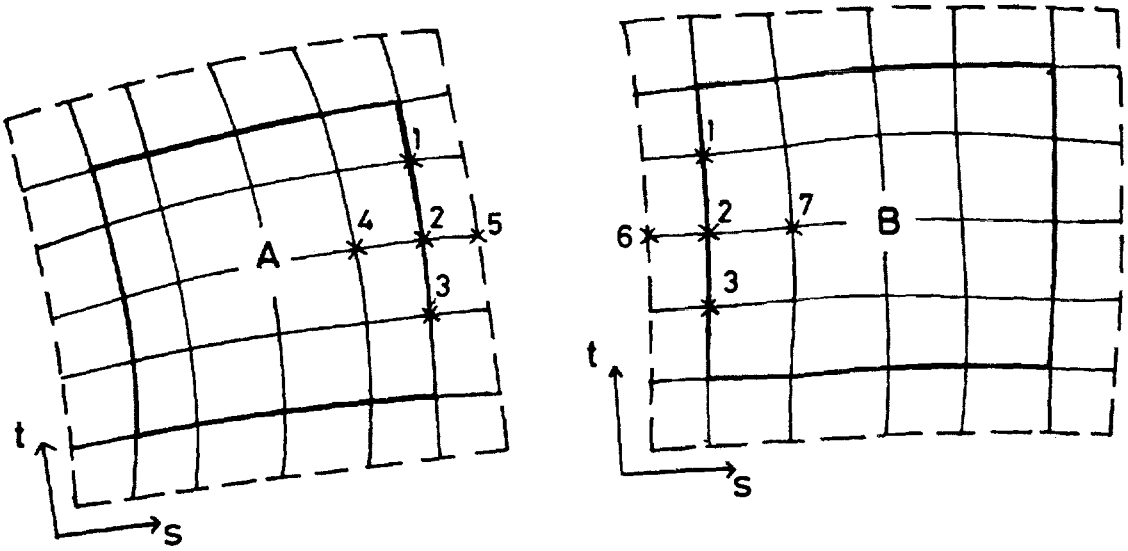
7. FLOW EQUATION

At all interior grid points of each segment the flow is calculated by solving a finite difference approximation to the compressible potential equation. Appropriate boundary conditions, described later, are applied on the four sides of each segment. We find it particularly convenient to work with the general tensor form of the potential equation and its boundary conditions. This is because this form eliminates dependence on the precise nature of the local grid transformations used and because the same equations encompass both two dimensions and three dimensions. Hence, methods developed in tensor form are equally applicable to two and three dimensions.

In Fig 7 the tensor form of the potential equation is shown in terms of a perturbation potential ϕ. It is written in so called rotated form (ie with the principle part split up into streamwise and streamnormal components). The underlined term is the streamwise component of the principle part and it is this term which is backward differenced in supersonic regions. The metric tensor g_{ij} represents a transformation between physical space with coordinates x^i = w,y,z and some arbitrary space with coordinates r^i = r,s,t.

This potential equation may be solved by any convenient numerical method. At present we solve it in nonconservative form using a line overrelaxation method. However, it is planned to implement an approximate factorisation scheme in order to improve the convergence rate in the near future.

Three main types of boundary conditions can be applied on the sides of each segment. The first two types: solid surface conditions (ie zero normal flow through the surface) and free stream conditions (ie zero perturbation velocity or zero perturbation potential) are the same as used with non-patched grids. These are applied in a standard way using dummy rows of grid points outside of the relevant boundaries and no further description will be given. The third type: patch boundary conditions are the heart of the patching method and will be described in detail.



(a) 2D flow equation in non-rotated form with

$r^1 = s,\ r^2 = t$

$q^2(c^2g^{11}-v^1v^1)\phi_{ss}+q^2(c^2g^{22}-v^2v^2)\phi_{tt} =$ cross derivatives - low order terms

(b) Continuity of normal velocity across common boundary

$$\left[\frac{v}{\sqrt{g^{11}}}\right]_A = \left[\frac{v}{\sqrt{g^{11}}}\right]_B .$$

i.e.

$$\left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s+x_s)+\frac{g^{12}}{\sqrt{g^{11}}}(\phi_t+x_t)\right]_A = \left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s+x_s)+\frac{g^{12}}{\sqrt{g^{11}}}(\phi_t+x_t)\right]_B$$

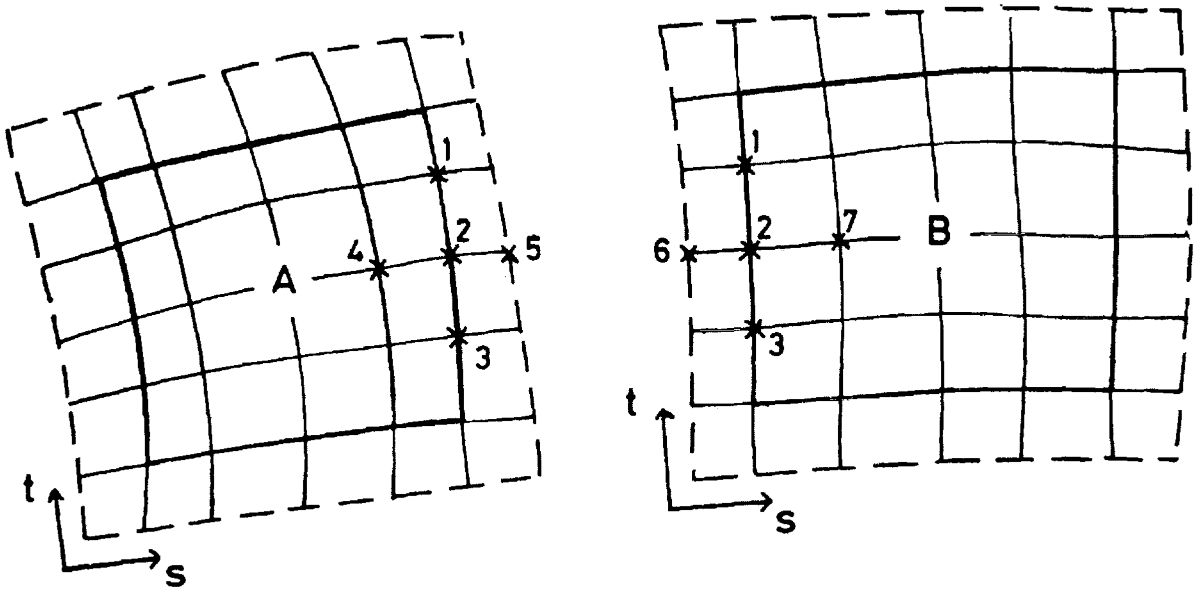
FIGURE 8 SUBSONIC PATCH BOUNDARY CONDITIONS -DIFFERENTIAL FORM

8. SUBSONIC PATCH BOUNDARY CONDITIONS - DIFFERENTIAL FORM

Fundamental to the satisfactory use of patched grids is the treatment of patch boundary conditions (ie the conditions applied along the common boundary of adjacent segments). Since such boundaries do not represent real flow boundaries but simply boundaries between different local grids, the flow equation is still satisfied on these boundaries and in addition the flow velocity along and across such boundaries is continuous. These conditions are sufficient to patch the flow calculations in adjacent segments together to produce the overall flow solution. The technique is somewhat easier to apply when the flow is subsonic at the boundary points and this case will be described first.

Fig 8 shows two adjacent segments A and B where for clarity the two segments are drawn as though separated although they are actually joined along the common boundary. Also shown surrounding each segment is a row of dummy points. These points do not actually exist but are convenient for the development of the patch boundary conditions.

Taking a typical point on the common boundary the usual five point finite difference star is shown for each segment. Points 1,2,3 are common to both segments being on the common boundary. Points 4,7 are internal to segments A and B respectively while points 5,6 represent dummy points. The flow equation is solved at all internal grid points of segments A and B using ϕ on the common boundary from the previous iteration. Hence, updated values of ϕ_4 and ϕ_7 are available and it is required to calculate updated values of ϕ on the common boundary, ie ϕ_1, ϕ_2, ϕ_3.



(1) Finite difference approximation to flow equation at point 2 in segment A

$$a_1(\phi_5-2\phi_2+\phi_4)+a_2(\phi_1-2\phi_2+\phi_3)=a_6$$

(2) Finite difference approximation to flow equation at point 2 in segment B

$$b_1(\phi_7-2\phi_2+\phi_6)+b_2(\phi_1-2\phi_2+\phi_3)=b_3$$

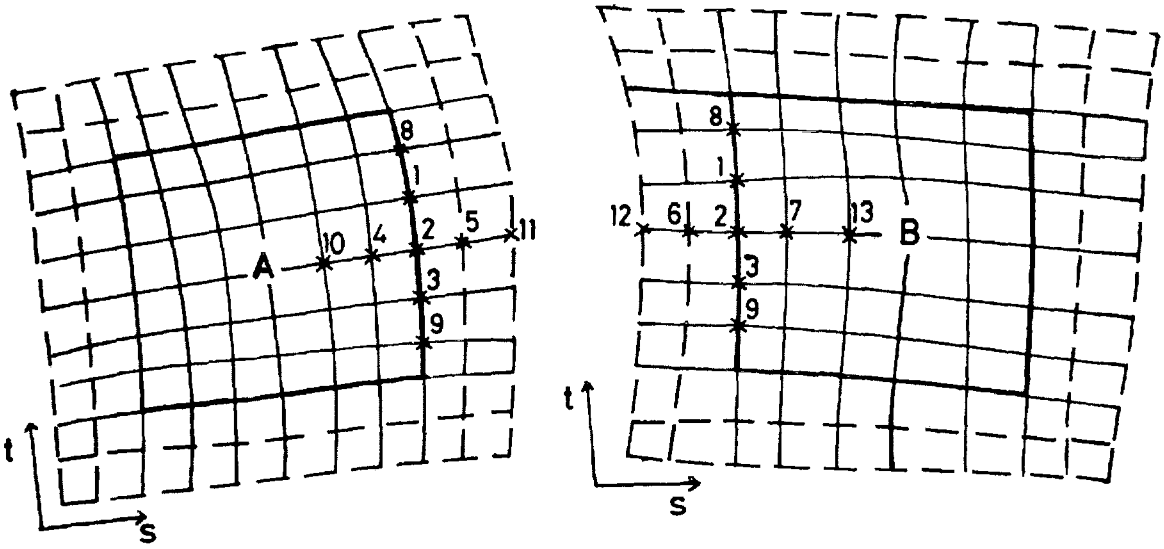
(3) Continuity of normal velocity at point 2

$$c_1(\phi_5-\phi_4)+c_2(\phi_1-\phi_3)+c_3=d_1(\phi_7-\phi_6)+d_2(\phi_1-\phi_3)+d_3$$

FIGURE 9 SUBSONIC PATCH BOUNDARY CONDITIONS DIFFERENCE FORM

9. SUBSONIC PATCH BOUNDARY CONDITIONS - DIFFERENCE FORM

This is done by writing finite difference approximations to the flow equation at point 2 separately for segments A and B using second order central differences which gives two equations and five unknowns (ϕ_1, ϕ_2, ϕ_3, ϕ_5, ϕ_6). A third equation with the same unknowns can be obtained using a finite difference approximation to the condition that the velocity normal to the common boundary is continuous across the boundary. Again second order central differences are used to approximate the velocities. (Continuity of velocity along the boundary is implicit in deriving the above equations). By combining these three equations the dummy values ϕ_5, ϕ_6 can be eliminated leaving one equation with three unknowns ϕ_1, ϕ_2, ϕ_3. Applying the same technique at all points along the common boundary produces a tridiagonal system of equations which may be solved for ϕ_1, ϕ_2, ϕ_3 etc using the standard algorithm.



(a) 2D flow equation in rotated form with $r = s$, $r^2 = t$

$$(c^2-q^2)v^1v^1\phi_{ss}+(c^2-q^2)v^2v^2\phi_{tt}+c^2(q^2g^{11}-v^1v^1)\phi_{ss}+c^2(q^2g^{22}-v^2v^2)\phi_{tt}$$

= cross derivatives + low order terms

(b) Continuity of normal velocity across common boundary

$$\left[\frac{v}{\sqrt{g^{11}}}\right]_A = \left[\frac{v}{\sqrt{g^{11}}}\right]$$

i.e.

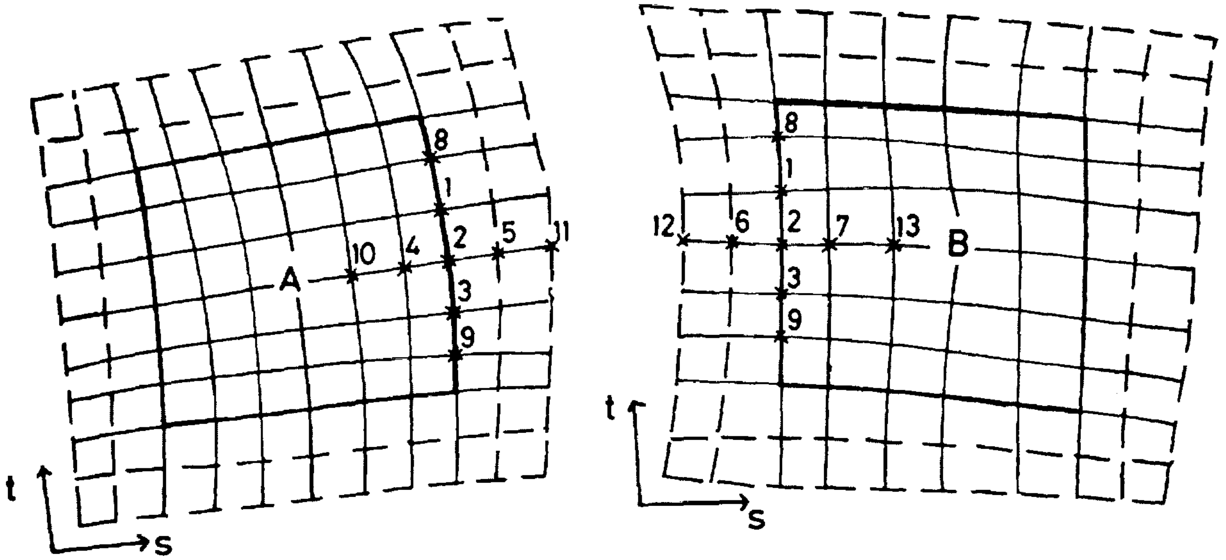
$$\left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s+x_s)+\frac{g^{12}}{\sqrt{g^{11}}}(\phi_t+x_t)\right]_A = \left[\frac{g^{11}}{\sqrt{g^{11}}}(\phi_s+x_s)+\frac{g^{12}}{\sqrt{g^{12}}}(\phi_t+x_t)\right]_B$$

FIGURE 10 SUPERSONIC PATCH BOUNDARY CONDITIONS -DIFFERENTIAL FORM

10. SUBSONIC PATCH BOUNDARY CONDITIONS - DIFFERENTIAL FORM

When the flow at some points on a patch boundary is supersonic there is a further problem in applying the patch boundary conditions. This is due to the backward differences used to approximate some of the flow equation derivatives at supersonic points. Fig 10 again shows a pair of segments A and B patched along a common boundary. In this case, however, there are two rows of dummy points around each segment to allow for backward differencing and the difference stars have nine rather than five points.

For any specific case only seven of the nine points are actually used, which seven depending on the local flow direction. If we assume that the local flow is from bottom left to top right then in segment A points 1,2,3, 4,5,9,10 are used while points 1,2,3,6,7,9,12 are used in segment B. Comparing with the subsonic case there are now three points (4,7,10) for which updated values of ϕ are available from the solution of the flow equation at internal grid points and three dummy points (5,6,12) to be eliminated.



(1) Finite difference approximation to flow equation at point 2 in segment A

$$a_1(\phi_2-2\phi_4+\phi_{10})+a_2(\phi_2-2\phi_3+\phi_9)+a_3(\phi_5-2\phi_2+\phi_4)+a_4(\phi_1-2\phi_2+\phi_3)=a_5$$

(2) Finite difference approximation to flow equation at point 2 in segment B

$$b_1(\phi_2-2\phi_6+\phi_{12})+b_2(\phi_2-2\phi_3+\phi_9)+b_3(\phi_7-2\phi_2+\phi_6)+b_4(\phi_1-2\phi_2+\phi_3)=b_5$$

(3) Continuity of normal velocity at point 2 (central differenced)

$$c_1(\phi_5-\phi_4)+c_2(\phi_1-\phi_3)+c_3=d_1(\phi_7-\phi_6)+d_2(\phi_1-\phi_3)+d_3$$

(4) Continuity of normal velocity at point 2 (backward differenced)

$$e_1(3\phi_2-4\phi_4+\phi_{10})+e_2(3\phi_2-4\phi_3+\phi_9)+e_3=$$

$$f_1(3\phi_2-4\phi_6+\phi_{12})+f_2(3\phi_2-4\phi_3+\phi_9)+f_3$$

FIGURE 11 SUPERSONIC PATCH BOUNDARY CONDITIONS -DIFFERENCE FORM

Thus four equations are required to eliminate the dummy points for the supersonic case rather than three. The first three equations are essentially the same as for the subsonic case except that appropriate second derivatives in the two approximations to the flow equation are now backward differenced rather than centrally differenced. After some experimentation we find the best equation to use for the fourth equation is another finite difference approximation to the continuity of normal velocity condition but this time approximating the velocities by second order backward differences in the usual upstream sense. This ensures that the value of ϕ at the extra dummy point (point 12 in this case) is not influenced by downstream values of ϕ which would violate the domain of dependence conditions.

When ϕ_5, ϕ_6 and ϕ_{12} have been eliminated from these four equations a single equation with four unknowns (ϕ_1, ϕ_2, ϕ_3, ϕ_9) is left. Applying the same technique at each point along the common boundary leads to a quadradiagonal system of equations which may be solved for ϕ_1, ϕ_2, ϕ_3, ϕ_9 etc. In practice, we reduce this set to a tridiagonal system, which is easier to solve, by fixing ϕ_9 at its value from the previous iteration.

Experience so far suggests that supersonic points on a patch boundary are more likely to lead to instability than are subsonic points. However, with some care it has been possible to satisfactorily compute cases with all subsonic, all supersonic and with mixed boundary points including one case where a strong shock crossed the boundary.

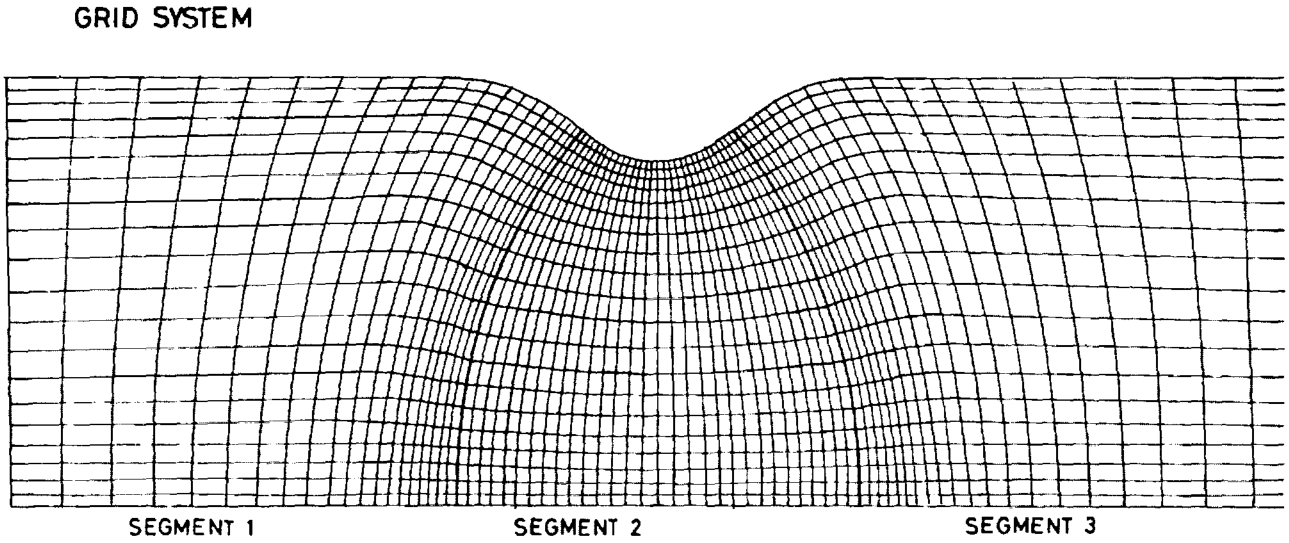


FIGURE 12 CONDI NOZZLE

12. EXAMPLE CONDI NOZZLE GRID

In order to demonstrate the use of the method, we considered the case of a duct flow. The configuration is that of a convergent-divergent nozzle produced by a cosine distortion on the upper surface of a two-dimensional duct with an area ratio of 0.8.

Clearly, it is possible to produce a single segment system to solve this problem but for the sake of demonstration we have divided it into three segments. The interfaces between the segments are denoted by the more pronounced lines and these patches are normal to both upper and lower surfaces. Note that, although the lines appear to have continuous derivatives through the segment boundaries, this is not the case. The grid points have been distributed in an appropriate manner with a much finer grid near the bump on the upper surface. For this example the grid extends to finite distances upstream and downstream and uniform onset flow is assumed at the upstream end.

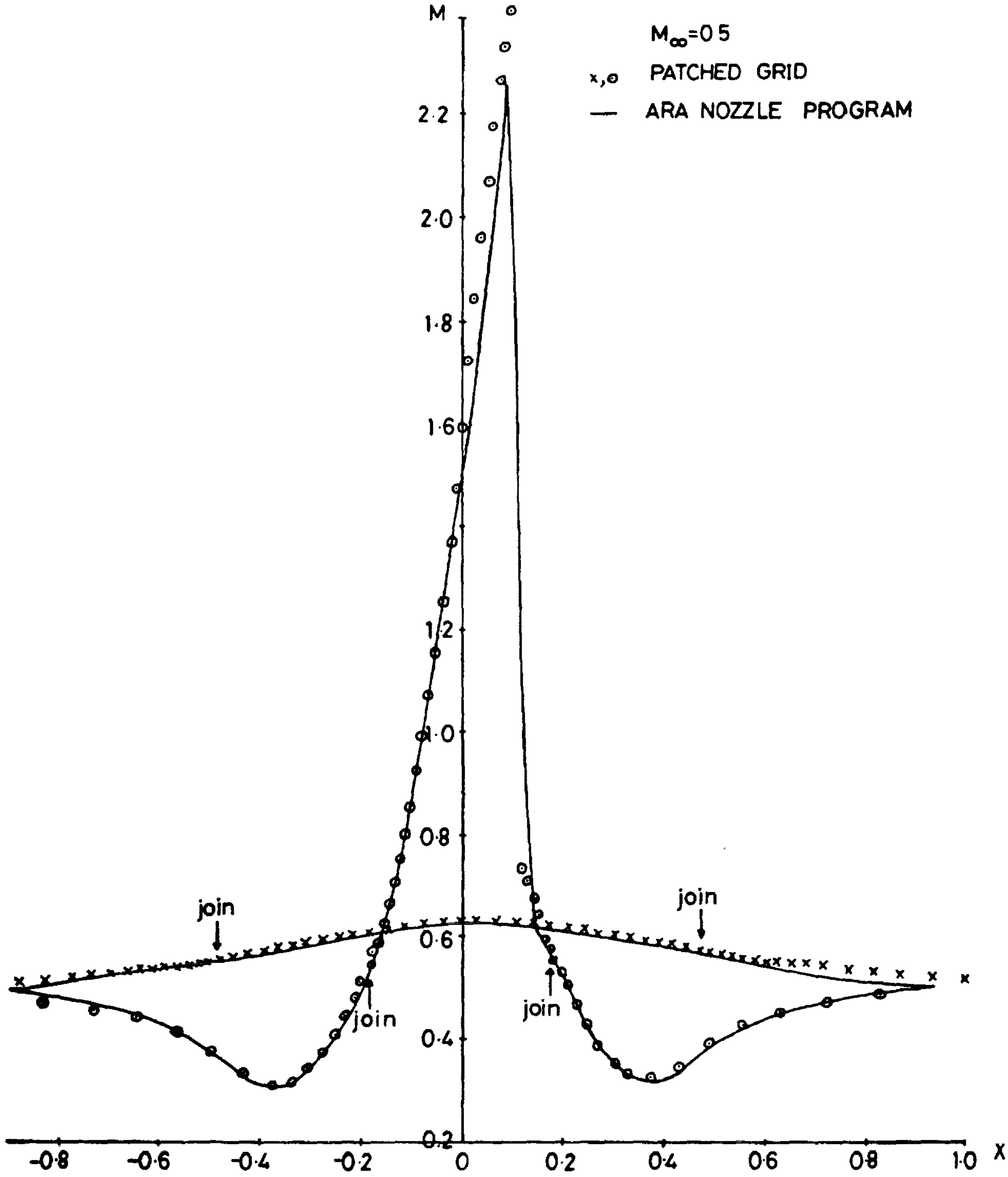


FIGURE 13 CONDI NOZZLE

13. EXAMPLE CONDI NOZZLE RESULTS

In order to check the results, comparisons have been made with a well established nozzle program developed by Baker at ARA. The case shown here is for an onset Mach number of 0.5. The inviscid flow solution, which is outside the range of validity of a potential method, is nevertheless an appropriate test case with a very strong shock. The agreement between the two methods is very encouraging and the patching does not appear to have affected the solution. However, possibly due to slow convergence, there was a small discontinuity across the patch but when the mean value is used, the result is reasonably smooth. The locations of the segment interfaces are shown on the figure. Although they are not shown here, changes in the position and number of patches did not affect the result significantly.

CONCLUSIONS AND PROPOSALS FOR FURTHER WORK

1 Initial use of grid patching is encouraging

2 The range of application has been limited

3 Further cases are now being attempted

e.g. aerofoil in wind tunnel

cascade flows

intake flows

4 Modifications required for unrestricted far field

5 Extension to three dimensions

FIGURE 14

14. CONCLUSIONS AND PROPOSALS FOR FURTHER WORK

The grid patching technique has been investigated using two-dimensional test cases and initial results are encouraging. However, the range of application has, so far, been limited and cases with a greater number of segments are now being attempted. These include an aerofoil in a wind tunnel together with the two configurations shown earlier, cascade and intake flows. For the latter case some work is required in introducing extra transformations for an unrestricted far field.

We should then be in a position to deal with most two-dimensional problems and this should form the basis of extending the techniques into three dimensions.