JOURNAL OF AIRCRAFT
Vol. 37, No. 1, January–February 2000

Chimera Donor Cell Search Algorithm Suitable for Solving the Full Potential Equation

Terry L. Holst*
NASA Ames Research Center, Moffett Field, California 94035

An approximate iterative search algorithm for finding donor cells associated with the chimera zonal grid approach is presented. This new algorithm is both fast and simple. It is used in conjunction with a chimera-based full potential solver for computing transonic flow solutions about wing and wing/fuselage configurations. Within each grid zone a fully implicit approximate factorization scheme is used to advance the solution one iteration. This is followed by the explicit advance of all common intergrid boundaries using a trilinear interpolation of the velocity potential. The presentation is highlighted with numerical result comparisons, a grid refinement study, and parametric variation of pertinent algorithm parameters. The new search algorithm produces donor cells for the two-zone wing problem at a rate in excess of 60,000 cells/s (single processor Cray C90). The approximate nature of the search algorithm, which causes some of the donor cells to be approximated by nearest neighbor cells, does not cause any impact on solution accuracy. Overall the results indicate that the present chimera zonal grid approach is a viable technique for solving the full potential equation for aerodynamic applications.

Nomenclature

b	= wing span
C	= wing chord
C_D	= wing pressure drag coefficient
C_L	= wing lift coefficient
C_P	= pressure coefficient
H	= transformation Jacobian
i, j, k	= grid indices corresponding to the ξ, η, and ζ computational coordinates, respectively
L	= fuselage length
$\boldsymbol{M}$	= column vector of i, j, and k
M_∞	= freestream Mach number
NI	= maximum value for the ith index
NJ	= maximum value for the jth index
NK	= maximum value for the kth index
$\boldsymbol{R}$	= column vector of x, y, and z
R_M	= computing rate in millions of floating point operations per second for the donor cell search algorithm only
x, y, z	= spatial coordinates in the physical domain
α	= angle of attack, deg
ξ, η, ζ	= spatial coordinates in the computational domain
$\xi_x, \xi_y, \ldots$	= computational domain transformation metrics

Superscripts

d	= donor cell quantity
n	= iteration index (nth iteration)
r	= intergrid boundary point or receiver point quantity

Introduction

THE goal of this paper is to present a new donor cell search algorithm for the chimera zonal grid approach that is simple and computationally efficient. The new search algorithm is presented in conjunction with a new chimera-based full potential solver used to obtain transonic flow numerical solutions about wing and wing/fuselage configurations. The chimera or overset grid approach is a versatile technique for obtaining aerodynamic results for complex geometric configurations including reasonably complete aircraft. In this zonal grid approach, a separate boundary conforming grid is generated about each major feature of a complex aerodynamic configuration. For example, for a transport aircraft consisting of a wing/body/pylon/nacelle a total of five grid zones might be used, one for the near field surrounding each of the major geometric features (wing, body, pylon, and nacelle) and a fifth background grid to connect the near-field grid zones to freestream. Each of the component grids is generated without significant regard to any of the other component grids. Each boundary grid point receives its boundary condition information from either freestream, flow tangency (assuming the solver is inviscid), symmetry, or from another component grid using interpolation. Some of the grid cells generated for one component of the geometry may have grid points that lie inside other components of the geometry. Computations at such points are handled by an IBLANK array multiplier that has a value associated with each grid point. IBLANK is equal to one for points in valid flow regions and to zero for points in invalid or blanked out flow regions. Computations proceed in an identical fashion at all grid points, which permits efficient code vectorization and/or parallelization, but because of the IBLANK array multiplication, only grid points in valid regions of flow get updated as the iteration proceeds.

Early work in the development of the multiple-zone overset or chimera approach may be found in Refs. 1 and 2. In Refs. 1 and 2, the basic concepts of the approach were developed with applications consisting of relatively simple two- and three-dimensional simulations. The approach was further developed and applied to a large variety of complex geometry applications involving both steady and unsteady flow.[3–14] In the cited examples, the Euler or Navier–Stokes governing equations were used. Two example applications in which a chimera approach was used to solve the full potential equation include Refs. 15 and 16. Finally, information on error analysis associated with the chimera approach is presented in Refs. 17 and 18.

The new donor cell search algorithm presented herein is especially important to the present full potential chimera scheme because of speed. Typical efficiencies for older donor cell search algorithms may range from 100 to about 1500 donor cells found per CPU second[19] (steady applications on the Cray 2 computer). With a typical application consisting of perhaps 25,000 intergrid boundary points (IGBPs), total search times could range from about 17 to 250 s. This time does not include surface or volume grid generation

Presented as Paper 97-2259 at the AIAA 15th Applied Aerodynamics Conference, Atlanta, GA, 23–25 June 1997; received 9 November 1997; revision received 4 June 1999; accepted for publication 21 July 1999.

*Research Scientist, Computational Physics and Simulation Branch. Fellow AIAA.

and/or hole cutting, which are also required for any chimera approach. Because the full potential solver itself will require only about 30–300 s (depending on the application and level of grid refinement), it is imperative to keep problem setup, including the donor cell search, as efficient as possible. For applications that require repeated updates of the chimera logic, for example, moving gird or numerical optimization problems, having an efficient donor cell search procedure is even more important. Thus, the new donor cell search algorithm is ideally suited to the present chimera full potential approach and could be useful in a number of other applications as well. The resulting chimera full potential scheme more favorably fills the role of a fast, albeit approximate, alternative to the Euler equations for preliminary aerodynamic analysis.

This paper begins with a brief presentation of the governing equation formulation and the numerical scheme used. Next, the new donor cell search algorithm is presented and discussed in detail. The computational results presented are for transonic flow simulations about two-grid-zone wing and three-grid-zone wing/fuselage configurations. A variety of computer timings to establish the donor cell search scheme's computational efficiency is presented throughout the results section. Grid refinement and search algorithm parametric variations are also included to allow complete assessment of the new chimera full potential scheme. Finally, the paper ends with concluding remarks.

Governing Equation Formulation

The steady, three-dimensional, full potential equation written in strong conservation-law form is given by

$$(\rho\phi_x)_x + (\rho\phi_y)_y + (\rho\phi_z)_z = 0 \tag{1a}$$

$$\rho = \left\{1 - [(\gamma - 1)/(\gamma + 1)](\phi_x^2 + \phi_y^2 + \phi_z^2)\right\}^{1/(\gamma-1)} \tag{1b}$$

where ρ is the fluid density; x, y, and z are Cartesian coordinates; γ is the ratio of specific heats; and ϕ is the full or exact velocity potential. The mere existence of the velocity potential implies that the curl of the velocity vector must vanish. Thus, flows governed by the full potential equation must be irrotational. In addition, derivation of the density relation [Eq. (1b)] requires the assumption of isentropic flow. In Eqs. (1) the density and velocity components (ϕ_x, ϕ_y, and ϕ_z) are nondimensionalized by the stagnation density ρ_s and the critical speed of sound a_*, respectively. The full potential equation is completed with freestream boundary conditions applied at the outer boundary and flow tangency conditions applied at the wing, fuselage, and symmetry plane boundaries.

Equation (1) expresses mass conservation for flows that are isentropic and irrotational. Despite these limiting assumptions, the full potential formulation can be used in a shock-capturing context providing the shock waves are weak. The corresponding shock-jump conditions are valid approximations to the Rankine–Hugoniot shock-jump conditions (derived from the Euler equations) for many applications. The normal component of the Mach number just upstream of the shock wave in question must be below about 1.3 for the full potential formulation to be a reasonable approximation to the Euler equations. This is well within the scope of many transonic and low supersonic flow applications and includes the cruise conditions for most transonic transport aircraft. More discussion on this point, including a comparison of the Euler and isentropic full potential shock polars, is presented by Steger and Baldwin.[20]

Equation (1) is transformed from the physical domain (Cartesian coordinates) into the computational domain using a general, independent-variable transformation. This general transformation, indicated by

$$\xi = \xi(x, y, z), \qquad \eta = \eta(x, y, z), \qquad \zeta = \zeta(x, y, z) \tag{2}$$

maintains the strong conservation-law form of Eqs. (1) and has been used in many formulations for a wide variety of applications. The final transformed version of the full potential equation along with the transformed boundary conditions is presented in detail in Ref. 16 and will not be discussed further.

Numerical Approach

The numerical scheme used in the present study to solve the full potential equation in the context of a zonal chimera grid approach is described in detail in Ref. 16 and will not be presented in detail here. The spatial discretization scheme used for all calculations presented herein is centrally differenced and second-order accurate at all subsonic points and utilizes a first-order-accurate upwind evaluation of the density to stabilize all supersonic points. A fully second-order-accurate upwind density evaluation that uses a density coefficient limiter, somewhat like the flux limiters used for the Euler equations, is available,[16] but is not used in the present study. A discretization scheme for evaluating all transformation metrics that preserves freestream and generally improves solution accuracy is used.[21,22] The iteration scheme used is a fully implicit approximate factorization scheme (called AF2) with certain modifications for C grids.[16] Additional modifications to the iteration scheme have been made for the chimera approach, that is, the so-called IBLANK array has been added in the appropriate places.[16] The numerical approach is used in each grid zone to advance the solution one iteration. Then, trilinear interpolation is used at all IGBPs to provide for grid-to-grid communications. The location of each IGBP's donor cell is determined before iteration begins. How the location of these donor cells is determined is the topic of the next section.

Donor Cell Search Algorithm

The donor cell search algorithm is described in detail as it is the newest aspect of the present chimera full potential approach. To begin, it is assumed that L grid zones ($G1$, $G2$, ..., GL) are defined by three-dimensional arrays of points and that each point is defined by a set of three coordinates using a base Cartesian coordinate system (x, y, z). This set of grids is given by

$$\begin{aligned}
&G1\colon x^1_{i,j,k},\ y^1_{i,j,k},\ z^1_{i,j,k}, && \text{where} \quad i = 1, 2, \ldots, NI^1, \quad j = 1, 2, \ldots, NJ^1, \quad k = 1, 2, \ldots, NK^1\\
&G2\colon x^2_{i,j,k},\ y^2_{i,j,k},\ z^2_{i,j,k}, && \text{where} \quad i = 1, 2, \ldots, NI^2, \quad j = 1, 2, \ldots, NJ^2, \quad k = 1, 2, \ldots, NK^2\\
&\ \ \vdots \qquad\qquad \vdots\\
&GL\colon x^L_{i,j,k},\ y^L_{i,j,k},\ z^L_{i,j,k}, && \text{where} \quad i = 1, 2, \ldots, NI^L, \quad j = 1, 2, \ldots, NJ^L, \quad k = 1, 2, \ldots, NK^L
\end{aligned} \tag{3}$$

where the NI, NJ, and NK quantities are the maximum index limits for i, j, and k, respectively. The ordering provided by the i, j, and k subscripts defines (in effect) a mapping or transformation between the physical domain (x, y, z) and the computational domain (ξ, η, ζ). This is the same general transformation represented by Eqs. (2). The connection between i, j, and k and ξ, η, and ζ is obtained using

$$\xi = i\Delta\xi, \qquad \eta = j\Delta\eta, \qquad \zeta = k\Delta\zeta \tag{4}$$

where the quantities $\Delta\xi$, $\Delta\eta$, and $\Delta\zeta$ are usually defined to be one.

Each of the grid zones defined by Eqs. (3) consists of structured sets of hexahedral cells, that is each cell consists of six sides, but the shape is essentially arbitrary and may involve cells with collapsed edges or faces providing the cell volume is not zero. Last, the origin of each grid zone's physical coordinate system is assumed to be

the same, that is all x, y, and z values in all of the grid zones are measured from the same coordinate system origin.

With these definitions the chimera donor cell search problem can be stated as follows: Given the location of an arbitrary IGBP that is associated with one of the grid zones in three-dimensional space x^r, y^r, and z^r find the appropriate grid cell that contains or surrounds this IGBP. The result of this search process (called the donor cell) must be from a grid zone other than the one supplying the IGBP. In the present approach the specific goal of the donor cell search is to find three index values i^d, j^d, and k^d that in turn define the donor cell using the following eight points:

$$x_{i^d,j^d,k^d},\ y_{i^d,j^d,k^d},\ z_{i^d,j^d,k^d}$$

$$x_{i^d+1,j^d,k^d},\ y_{i^d+1,j^d,k^d},\ z_{i^d+1,j^d,k^d}$$

$$x_{i^d,j^d+1,k^d},\ y_{i^d,j^d+1,k^d},\ z_{i^d,j^d+1,k^d}$$

$$x_{i^d,j^d,k^d+1},\ y_{i^d,j^d,k^d+1},\ z_{i^d,j^d,k^d+1}$$

$$x_{i^d+1,j^d+1,k^d},\ y_{i^d+1,j^d+1,k^d},\ z_{i^d+1,j^d+1,k^d}$$

$$x_{i^d+1,j^d,k^d+1},\ y_{i^d+1,j^d,k^d+1},\ z_{i^d+1,j^d,k^d+1}$$

$$x_{i^d,j^d+1,k^d+1},\ y_{i^d,j^d+1,k^d+1},\ z_{i^d,j^d+1,k^d+1}$$

$$x_{i^d+1,j^d+1,k^d+1},\ y_{i^d+1,j^d+1,k^d+1},\ z_{i^d+1,j^d+1,k^d+1}$$

If two or more cells from different grid zones contain the same IGBP, then the preferable donor cell is typically from the finest grid zone providing a reasonable amount of grid overlap is maintained, that is, donor cells generally should not be taken from or near the boundary of a grid zone, especially if the boundary contains points that are themselves IGBPs. This will provide the most accurate interpolated information for the IGBP and create the fewest numerical difficulties during flow solver iteration.

The new donor cell search algorithm utilizes the relational information between the base Cartesian coordinate system shared by all grid points (including the IGBP in question) and the local computational coordinate system associated with the particular grid zone being searched for the donor cell. This relationship is expressed by Eq. (2) and is utilized in the following differential form:

$$d\xi = \xi_x\,dx + \xi_y\,dy + \xi_z\,dz, \qquad d\eta = \eta_x\,dx + \eta_y\,dy + \eta_z\,dz$$
$$d\zeta = \zeta_x\,dx + \zeta_y\,dy + \zeta_z\,dz \tag{5}$$

where the metric quantities ξ_x, ξ_y, etc., are the same metric quantities used by the flow solver. Suitable numerical approximations to Eqs. (5) are given by

$$\begin{aligned} i^d - i^n &= \xi_x^n(x^r - x^n) + \xi_y^n(y^r - y^n) + \xi_z^n(z^r - z^n) \\ j^d - j^n &= \eta_x^n(x^r - x^n) + \eta_y^n(y^r - y^n) + \eta_z^n(z^r - z^n) \\ k^d - k^n &= \zeta_x^n(x^r - x^n) + \zeta_y^n(y^r - y^n) + \zeta_z^n(z^r - z^n) \end{aligned} \tag{6}$$

where all quantities with the n superscript are associated with an arbitrary starting cell from the donor grid zone and the ξ, η, and ζ coordinates have been replaced using Eqs. (4). To improve accuracy the quantities x^n, y^n, and z^n and all metrics, ξ_x^n, ξ_y^n, etc., in Eqs. (6), are evaluated at the cell center, that is, at $i^n + 1/2$, $j^n + 1/2$, and $k^n + 1/2$.

A compact alternate notation to that of Eqs. (6) is given by

$$\boldsymbol{M}^d = \boldsymbol{M}^n + H^n(\boldsymbol{R}^r - \boldsymbol{R}^n) \tag{7}$$

where

$$\boldsymbol{M} = \begin{pmatrix} i \\ j \\ k \end{pmatrix}, \qquad H = \begin{pmatrix} \xi_x & \xi_y & \xi_z \\ \eta_x & \eta_y & \eta_z \\ \zeta_x & \zeta_y & \zeta_z \end{pmatrix}, \qquad \boldsymbol{R} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$$

Equation (7) represents a simple, explicit set of computations for the desired donor cell location. The implementation of Eq. (7) is accomplished using mixed-mode arithmetic, that is, the left-hand-side result of Eq. (7) is integer, and the right-hand side has one integer term and one floating-point term. Thus, the information to the right of the decimal point associated with the floating-point term is lost as the integer left-hand sides are computed. To compensate for this integer chopping operation, the following modification is required:

$$\boldsymbol{M}^d = \boldsymbol{M}^n + \left[H^n(\boldsymbol{R}^r - \boldsymbol{R}^n) + 0.5\boldsymbol{I}_s\right] \tag{8}$$

where $\boldsymbol{I}_s$ is a column vector defined by

$$\boldsymbol{I}_s = \text{sgn}[H^n(\boldsymbol{R}^r - \boldsymbol{R}^n)]$$

In this definition, sgn is used to indicate replacement of each column vector element by its sign.

If the starting cell x^n, y^n, or z^n is very far removed from the desired donor cell, the evaluation of Eq. (8) will yield inaccurate results, that is, the desired donor cell may be missed by several grid cells. This can easily be corrected by iterating Eq. (8) as follows:

$$\boldsymbol{M}^{n+1} = \boldsymbol{M}^n + \left[H^n(\boldsymbol{R}^r - \boldsymbol{R}^n) + 0.5\boldsymbol{I}_s\right] \qquad n = 1, 2, \ldots, \text{MAXIT} \tag{9}$$

where the maximum number of iterations allowed (MAXIT) is usually around 10.

During the course of iteration, especially on the first iteration if the initial guess is a poor one, Eq. (9) can produce results for i^{n+1}, j^{n+1}, and k^{n+1} that are outside the current grid zone's index limits. (This can also occur if the grid zone being searched is not the proper donor grid zone for the present IGBP.) If this happens, the iteration will fail because metric information does not exist outside these grid index limits. Thus, the following limits must be placed on each iteration:

if	$i^{n+1} < 1$,	then	$i^{n+1} = 1$
if	$i^{n+1} > NI - 1$,	then	$i^{n+1} = NI - 1$
if	$j^{n+1} < 1$,	then	$j^{n+1} = 1$
if	$j^{n+1} > NJ - 1$,	then	$j^{n+1} = NJ - 1$
if	$k^{n+1} < 1$,	then	$k^{n+1} = 1$
if	$k^{n+1} > NK - 1$,	then	$k^{n+1} = NK - 1$

These limits are minimums required for proper operation of the donor cell search iteration. More stringent limits may be implemented using this logic to keep donor cell selections away from boundaries as desired.

The donor cell search iteration is declared converged with the following convergence criteria:

$$\left|\left[H^n(\boldsymbol{R}^r - \boldsymbol{R}^n) + 0.5\boldsymbol{I}_s\right]\right| \le 1.0 + \text{TOL} \tag{10}$$

where a positive user-specified tolerance (TOL) is typically set to a value around 0.05. The preceding convergence criteria require the absolute value of each component of the last term in Eq. (9) to be bounded by one (within some tolerance). Satisfaction of Eq. (10) with TOL = 0.0 is a statement that $\boldsymbol{R}^r$ lies inside the cell defined by $\boldsymbol{R}^n$. This is an approximate donor cell definition. Therefore, it does not guarantee that the IGBP lies inside the so-defined donor cell, but, as will be seen in the results section, represents an adequate criteria for the present chimera full potential solver. When the donor cell is missed, the interpolation process automatically becomes an extrapolation process. More on what this approximate donor cell philosophy means will be provided in the results section.

The parameter TOL introduced in Eq. (10) is required for the situation when an IGBP lies on or near the boundary between two cells in the donor grid zone. The iteration can produce a limit cycle oscillation between the two cells and never converge. Use of the TOL parameter solves this problem. The TOL parameter can also be used to study the effect of using extrapolation (instead of interpolation) to satisfy IGBP values. By using large values of TOL, for example, on the order of one, the number of nearest neighbor cells selected

as actual donor cells can be very large. Thus, the number of IGBPs receiving extrapolated information will also be large. Comparisons of these solutions with TOL = 0.05 solutions can help answer the question of how sensitive the chimera scheme is to extrapolation at intergrid boundaries.

If convergence occurs, then the IGBP/donor-point information just obtained is saved, and the search algorithm proceeds to the next IGBP. If convergence does not occur, that is, if $n = \text{MAXIT}$ without the convergence criteria being satisfied, then the current IGBP is flagged. After donor cell searches are performed for all IGBPs for the initial donor grid zone, the search algorithm proceeds to the next donor grid zone and initiates a search for all remaining flagged IGBPs. After all appropriate grid zones have been searched, all IGBPs should have donor cells. If flagged IGBPs still exist, they are orphans and an appropriate error message is written for each orphan point.

The donor cell search iteration does not always converge, for example, near grid singularities when TOL is relatively small the numerically evaluated metrics may not be accurate enough to guide the search iteration to the proper donor cell. This problem could perhaps be solved with a more accurate evaluation of the metrics in the vicinity of singularities or with a local stencil-walk search. In addition, in some applications, usually involving coarse grids, the overlap between two grid zones is actually negative, that is, the two grid zones do not overlap. The result in this latter case is an orphan IGBP with execution termination and probably a tedious regridding step. Both of these problems have been solved in the present implementation using another philosophy, which is compatible with the approximate nature of the full potential formulation. If Eq. (10) has not been satisfied and

$$n = \text{MAXIT}$$

or

$$i^{n+1} = i^n, \qquad j^{n+1} = j^n, \qquad k^{+1} = k^n$$

then an alternative convergence criteria is used

$$\left| \left[H^n(\boldsymbol{R}^r - \boldsymbol{R}^n) + 0.5\boldsymbol{I}_s \right] \right| \leq 2.0 \tag{11}$$

This new criteria [after giving Eq. (10) every chance to be satisfied] picks a donor cell that is nominally within one cell of the IGBP's actual donor cell. For the sake of discussion, this type of cell will be hereafter called a nearest neighbor donor cell or just a nearest neighbor. The trilinear interpolation logic used to satisfy the IGBP velocity potential value automatically converts to extrapolation for nearest neighbor cells. Use of this philosophy to supply donor cells for near orphan IGBPs has been used for numerous two- and three-zone computations with good success and will be discussed in more detail in the results section.

One last search algorithm issue needs to be discussed. This issue is associated with grid zones that are highly curved in the physical domain such as the C–H wing grid zone used in the present study (to be described in the next section). For grid zones of this type, an initial donor cell location guess that is above the wing for an IGBP that is below the wing will lead to difficulties. The donor cell search iteration as described earlier will become trapped above the wing, bumping up against the wing boundary, unsuccessfully trying to move from above the wing to below the wing on every iteration. The fix for this problem is to determine which side of the full potential vortex sheet a particular IGBP is on and then force the initial donor cell location to be on the same side. A variety of different locations for the initial donor cell (all on the same side of the vortex sheet as that of the IGBP in question) have been tested, and all produced the proper donor cell.

Numerical Results

Numerically computed results are presented for two configurations, an isolated-wing geometry involving two grid zones and a wing–fuselage configuration involving three grid zones. In each case solutions are computed for two to three levels of grid refinement and for several values of the parameter TOL. For the two-zone isolated wing case, comparisons are made with an exact search algorithm in terms of computational efficiency and solution accuracy.

Isolated-Wing Case

To evaluate the attributes of the new donor cell search algorithm within the chimera full potential scheme, the familiar ONERA M6 wing geometry is chosen. The two-zone chimera grid consists of an inner C–H grid near the wing (C topology in the chordwise direction and H topology in the spanwise direction) embedded in a much larger outer Cartesian-like grid. The Cartesian grid is sheared so that one of the grid planes is approximately aligned with the wing leading edge and another with the wing trailing edge. In addition, the Cartesian-like grid is clustered in all three directions (streamwise, spanwise, and vertical) in the vicinity of the wing to provide a more consistent interface at the inner chimera interpolation boundary. For the present isolated-wing test case, both the inner and outer grids are constructed such that each $\eta = \text{const}$ surface is also a $y = \text{const}$ plane.

All C–H grids used are generated using the HYPGEN hyperbolic grid generation code[23] with full three-dimensional capability. Then after the grid is generated, the $\eta = \text{const}$ surfaces are adjusted to achieve the desired constant y-plane distributions. Figures 1 and 2 show selected views from a typical two-zone grid generated in this fashion about the ONERA M6 wing. Figure 1 shows a closeup of both the inner C–H and outer Cartesian grids plotted in the wing-root symmetry plane. Note the set of grid points from the outer Cartesian-like grid that have been removed near the wing. These

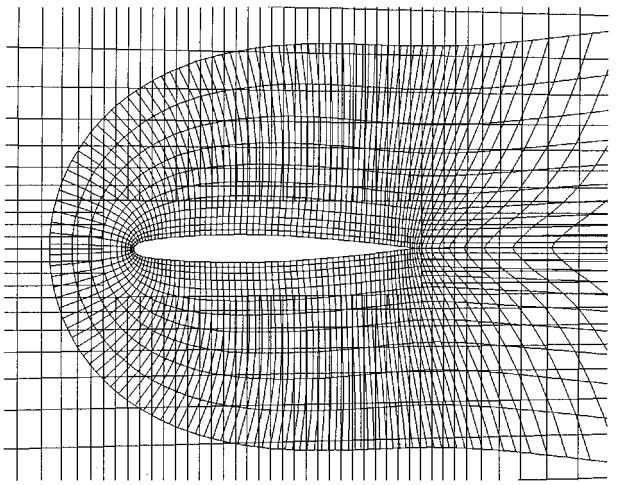


Fig. 1 Expanded view of a typical two-zone grid in the wing-root plane, $y = 0.0$, inner grid = 151 × 32 × 13, and outer grid = 73 × 33 × 50.

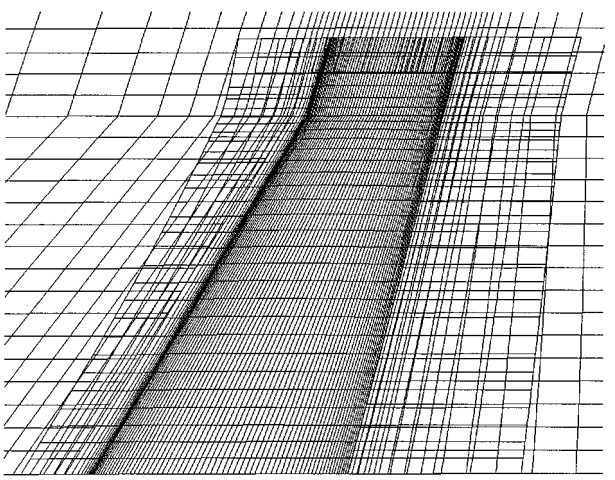


Fig. 2 Expanded view of a typical two-zone grid in the wing planform plane, $z = 0.0$, inner grid = 151 × 32 × 13, and outer grid = 73 × 33 × 50.

Table 1 Fine grid results summary of donor cell search statistics showing effect of TOL variation[a]

TOL	NMISS[b]	NNN	NAVG	CPU, s	R_M	C_L	C_D
——	——	——	——	5.980	120	0.2913	0.01081
0.00	427	633	2.17	0.265	234	0.2912	0.01081
0.02	1,131	18	1.83	0.253	244	0.2912	0.01081
0.10	3,426	0	1.81	0.234	263	0.2912	0.01081
0.30	7,991	0	1.77	0.243	254	0.2912	0.01080
1.00	10,710	0	1.71	0.228	271	0.2911	0.01079

[a]First line from an exact search algorithm.
[b]Number of IGBPs outside their specified donor cells (as determined by an exact search algorithm).

Table 2 Coarse grid results summary of donor cell search statistics showing effect of TOL variation[a]

TOL	NMISS[b]	NNN	NAVG	CPU, s	R_M	C_L	C_D
——	——	——	——	0.726	85	0.2850	0.01031
0.00	185	338	2.59	0.066	125	0.2846	0.01030
0.02	601	12	1.95	0.060	134	0.2846	0.01030
0.10	1795	0	1.90	0.056	143	0.2845	0.01029
0.30	2878	0	1.84	0.055	147	0.2841	0.01026
1.00	3448	0	1.77	0.052	154	0.2844	0.01026

[a]First line from an exact search algorithm.
[b]Number of IGBPs outside their specified donor cells (as determined by an exact search algorithm).

points still exist and are used in the computation, but their effect on all other points outside the hole has been removed using the IBLANK array multiplier. The inner grid is not orthogonal as one would expect with a hyperbolic grid generation technique because of a large amount of smoothing used in the grid generation process. Note that the outer Cartesian grid is somewhat coarser than the inner C–H grid. This produces a somewhat larger truncation error on the Cartesian side of the interface boundary, a characteristic that will be particularly noticeable near all high-frequency components of the solution, especially shock waves. Figure 2 shows both grids plotted around the wing in a planform view, where the y = const plane character is clearly evident. The outer grid shearing/stretching and the inner C–H grid clustering at the wing tip are also clearly visible.

A study of the effect of the TOL parameter defined in Eq. (10) on the donor cell search process is presented in Table 1 for an inner grid $= 201 \times 41 \times 17$ and outer grid $= 109 \times 49 \times 74$ and in Table 2 for an inner grid $= 101 \times 23 \times 9$ and outer grid $= 55 \times 25 \times 38$. Computed results for the flow about the ONERA M6 wing geometry ($M_\infty = 0.84$ and $\alpha = 3.06$ deg) for the two different levels of grid refinement and five values of TOL ranging from 0.0 to 1.0 are compared. The number of IGBPs is 15,299 for the fine grid case (Table 1) and 4,259 for the coarse grid case (Table 2). For each computation, a set of donor cells was also computed using an exact donor cell search algorithm. Results from the exact approach are displayed in the first row of both Tables 1 and 2. The number of donor cells computed by the new approach that do not match the exact search algorithm is listed under the NMISS column. This number (as expected) is smallest for the smallest values of TOL. For TOL $= 0.0$, the present donor cell search algorithm chooses the correct donor cell over 97% of the time for the fine grid and over 95% of the time for the coarse grid.

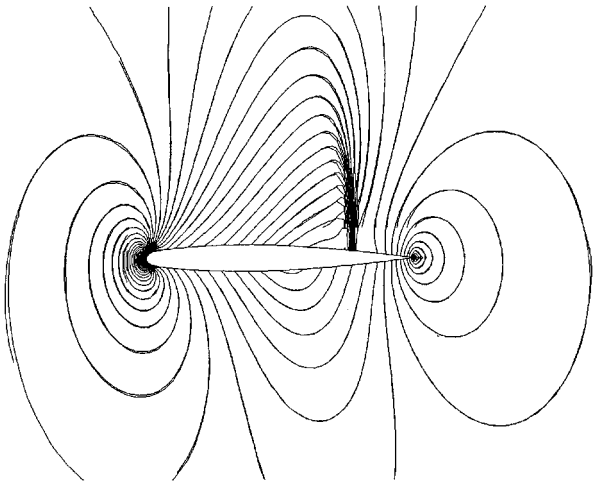
Note that the exact donor cell search algorithm used to compute the first lines of Tables 1 and 2 uses a series of position-vector/outward-normal-vector dot products to determine if a point is inside or outside a hexahedral cell. If the sides of the candidate donor cell are all planar, then six dot-product tests are required. If the hexahedral cell is general, that is, has nonplanar sides, then a tetrahedral decomposition of each hexahedral cell is required. This cell-by-cell search, although general, is exceptionally tedious and computationally expensive. In the present approach, this exact algorithm is significantly accelerated by reducing the donor grid zone test domain. The y = const nature of each η = const computational surface significantly enhances this test-domain reduction. As a result, far fewer than 1% of the total number of grid cells need to be searched. As a result, this exact algorithm is quite efficient; however, it is not a general approach that can be applied to a wide variety of cases.

The number of nearest neighbor cells selected by the present search algorithm (NNN) to be donor cells [as defined by the convergence criteria in Eq. (11)] is listed in the NNN column. The number of orphan IGBPs for each case is zero. The larger values of TOL allow the convergence criteria given by Eq. (10) to be satisfied more easily, and thus, fewer nearest neighbor cells are selected as donor cells. Of course, the actual number of selected donor cells that are not the actual donor cell increases as TOL increases as can be seen by the increasing number in the NMISS column. Note that when TOL $= 0.0$, the NNN is larger than the number in the corresponding NMISS column. In other words, some of the cells selected by the present search algorithm as nearest neighbor cells are actually the unique donor cells themselves and not nearest neighbor cells. This emphasizes the point that the present donor cell test (even with TOL $= 0.0$) is an approximation, in this case (more often than not) a pessimistic one. Nevertheless, the selected donor cell is always either the correct donor cell or an immediate neighbor that is very close to the corresponding IGBP. This is exemplified by the TOL $= 0.1$ result in Table 1. For this value of TOL, 100% of all found donor cells pass the Eq. (10) test and, thus, find either the correct donor cell or are within 0.1 cells of the correct donor cell.

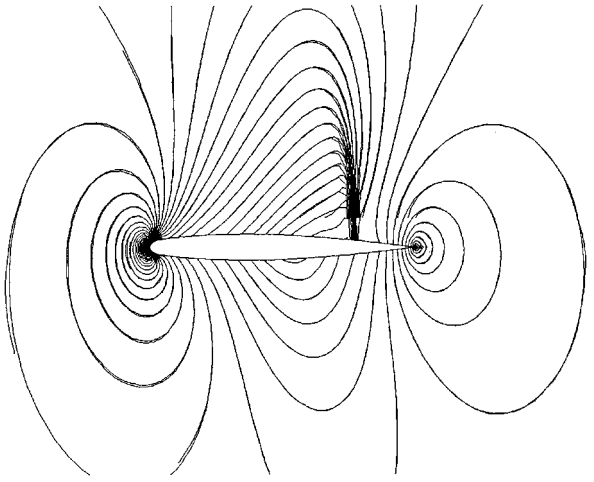
The average number of iterations required to produce a donor cell (NAVG), the computing time for the donor cell search (CPU), and the processing rate for the search algorithm R_M are all listed in Tables 1 and 2. A typical average value for the number of iterations to find one donor cell is about two. If the initial donor cell guess is close, for example, from the previous IGBP immediately adjacent to the current IGBP, convergence is obtained in usually a single iteration. If the initial donor cell guess is not close, for example, when beginning a new line of IGBPs, convergence may take several iterations. As can be seen, more iterations are required for smaller values of TOL, which is not surprising because the resulting test is more sensitive and, thus, more difficult to satisfy. This also slightly decreases computational efficiency, as can be seen from the R_M statistics. The search algorithm computing time includes the computation of a complete three-dimensional set of metrics for each grid zone, which is very efficient and has a large processing rate. The actual search process is logic driven with a relatively small number of floating-point operations and, therefore, has a relatively low processing rate. Thus, as NAVG increases, the resulting R_M value decreases. The number of donor cells found per second for the two-zone fine-grid cases of Table 1 is typically in excess of 60,000 cells/s, which is one to two orders of magnitude faster (with differences in computer speeds taken into account) than the rates quoted in Ref. 17.

The last important issue to be discussed for this case is the effect of the present donor cell search scheme's approximate nature on solution accuracy. In other words, what is the effect of using extrapolation to supply chimera intergrid boundary information on solution accuracy? This potential inaccuracy is a direct result of using nearest neighbor cells instead of the actual donor cells. This question can be answered (to an extent) by looking at the lift and drag coefficient values listed in Tables 1 and 2. As can be seen, there is an extremely small variation among the different cases, including the case using the exact search algorithm, that is, no extrapolation. In fact, the maximum differences displayed are one to two orders of magnitude less than the expected truncation and formulational errors associated with these computations.

The accuracy question can be examined further using the results of Fig. 3, in which Mach number contours are displayed about the ONERA M6 wing-root section using the same freestream conditions as earlier ($M_\infty = 0.84$ and $\alpha = 3.06$ deg). The fine-grid results associated with Table 1 are used for these contour plots. Figure 3a shows contours for TOL $= 0.0$, and Fig. 3b shows contours for TOL $= 1.0$. Solutions are plotted for both the inner and outer grid zones in Figs. 3a and 3b. The abrupt thickening of the shock wave



a) TOL = 0.0



b) TOL = 1.0

Fig. 3 Mach number contours about the ONERA M6 wing in the wing root symmetry plane, $M_\infty = 0.84$, $\alpha = 3.06$ deg, inner grid = 201 × 41 × 17, and outer grid = 109 × 49 × 74.

on the upper surface of the wing in both contour plots is a typical result of the chimera approach. The outer grid is not as fine as the inner grid in the vicinity of the shock wave and, thus cannot support the high-frequency detail of the shock wave as well as the inner grid zone. Note that in each solution overlap region away from the shock wave (Figs. 3a and 3b), most of the contour lines from the inner and outer solutions agree very well with each other. However, the key point of this comparison is that even though Fig. 3b was computed using extrapolated information from 10,710 nearest neighbor cells (approximately 70% of all IGBPs) the results are almost identical to the case using extrapolation for only 3% of all IGBPs (Fig. 3a). Thus, for the present case at least, the impact of using nearest neighbor cells to supply intergrid boundary information instead of actual donor cells is negligible.

Wing–Fuselage Case

Computed results for a wing–fuselage geometry[24] [Royal Aircraft Establishment (RAE) wing A mounted symmetrically on the cylindrical body B2] are presented next. The chimera grid for this geometry consists of three grid zones, which are highlighted in Fig. 4. The first grid zone is a C–H grid around the wing and is generated using the HYPGEN code. After the wing grid generation step, the wing–fuselage line of intersection is computed using an iterative algorithm, and then the wing-root grid is interpolated onto the fuselage using bicubic spline interpolation. The remaining three-dimensional wing grid is then sheared to match the root section. Thus, flow tangency boundary conditions can be implemented on both the wing

Table 3 Summary of donor cell search statistics for the three-grid zone wing–fuselage case[a] showing the effect of grid refinement, TOL = 0.1

Grid dimensions	Total points	IGBP	NNN	NAVG	CPU, s	R_M
		Grid 1				
101 × 23 × 9 73 × 13 × 30 61 × 25 × 38	107,390	7,631	6	5.72	0.220	60
		Grid 2				
151 × 32 × 13 109 × 19 × 44 81 × 33 × 50	287,590	15,480	2	5.35	0.433	81
		Grid 3				
201 × 41 × 17 145 × 25 × 58 101 × 41 × 62	607,089	26,423	6	4.99	0.714	102

[a]RAE wing A with B2 fuselage.

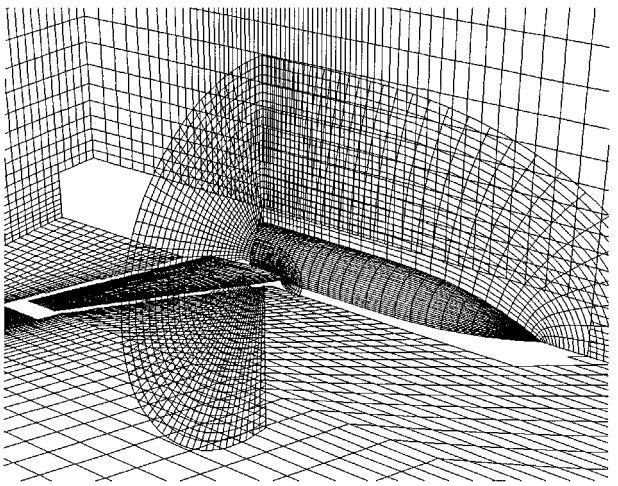


Fig. 4 Expanded view of a typical three-zone grid showing selected grid surfaces about the RAE wing A mounted symmetrically on cylindrical body B2.

($k = 1$) and fuselage ($j = 1$) surfaces associated with the C–H wing grid. The second grid zone is a C–O grid around the fuselage (C for $k =$ const surfaces and O for $i =$ const surfaces) and is also generated using the HYPGEN code. This grid has a hole cut from its middle around the wing using the chimera hole-cutting philosophy. The third grid is a sheared–stretched Cartesian-like outer grid. This grid has holes cut for both the wing and the fuselage, again, using the chimera hole-cutting philosophy. IBLANK values for all grid points lying in these hole regions are zero. Selected surfaces showing these details are shown in Fig. 4.

A summary of statistics showing the effect of grid refinement on the donor cell search algorithm is given in Table 3. Three sets of grids are utilized ranging from just over 100,000 total points to just over 600,000 total points. For all of these computations, the TOL parameter is set to 0.1. The processing rate for these computations (number of donor cells found per second) is approximately the same for all three cases. However, the processing rate obtained from these wing–fuselage cases is about a factor of two slower than for the two-zone isolated-wing cases. The reason for this is that the average number of search iterations has increased for the present three-zone cases (compare the NAVG columns in Tables 1 and 2 and Table 3). The reason for the NAVG increase is that for three grid zones the search process must not only find the correct donor cell but must also find the correct donor grid zone. This inevitably leads to many search iterations that produce no donor cells because the randomly chosen grid zone being searched may be the wrong grid zone. Thus, the average number of search iterations per found donor cell is larger. Despite this decrease in efficiency, the resulting processing rate is still quite attractive.

Table 4 Summary of donor cell search statistics for the three-grid zone wing–fuselage case[a] showing the effect of TOL variation

TOL	NNN	NAVG	CPU, s	R_M	IGBP
0.00	903	5.11	0.738	100	26,423
0.03	34	5.01	0.721	102	26,423
0.10	6	4.99	0.714	102	26,423
0.30	4	4.95	0.714	103	26,423
1.00	0	4.89	0.692	106	26,423

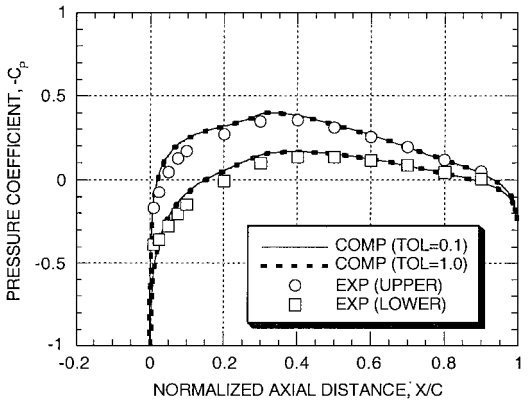
[a]RAE wing A with B2 fuselage.

A statistical summary showing the effect of TOL on the donor cell search algorithm for the wing-fuselage case is displayed in Table 4. All cases have been computed using the fine grid from Table 3, wing grid = 201 × 41 × 17, fuselage grid = 145 × 25 × 58, and outer grid = 101 × 41 × 62. Many of the same trends displayed by Tables 1 and 2 for the two-zone wing case are similar to the three-zone wing–fuselage case, that is, as TOL decreases, NNN increases, NAVG and CPU increase slightly, and $\boldsymbol{R}_M$ decreases slightly. Note that the number of nearest neighbor cells for TOL = 0.0 is only slightly more than 3% of the total number IGBPs. This is very similar to the TOL = 0.0 case from Table 1. In other words, based on the test of Eq. (10), almost 97% of the IGBPs for the present fine-grid wing–fuselage case have found actual donor cells and only 3% nearest neighbor cells.

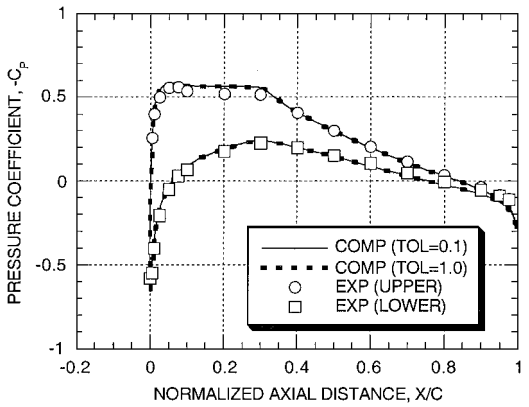
The TOL = 0.0 case displayed in Table 4 required a special adjustment in the donor cell search logic for sensible results. Occasionally, an IGBP on the fuselage from either the wing or fuselage grid would not find a donor cell because the truncation error associated with the numerical evaluation of the Eq. (10) convergence criteria would place the IGBP inside the fuselage. Thus, with TOL = 0.0, a proper donor cell was not found. Almost any value of TOL above zero (including 0.01, the smallest value tried) would save the situation. To produce proper TOL = 0.0 results for Table 3, a special tolerance of 0.02 was added to TOL for the jth index computation when $j^r = 1$, that is, when the IGBP was on the fuselage surface. Of course, a simple, and yet general fix for this situation is to always use a value of TOL above 0.02.

A detailed comparison of wing–fuselage computed results compared with experimental results from Ref. 24 is presented in Figs. 5 and 6. Figure 5 shows comparisons for the wing surface pressures at three semispan locations, $2y/b = 0.17$, 0.40, and 0.85. Figure 6 shows comparisons at four different meridinal angles around the fuselage, $\varphi = -15$, -0, $+0$, and $+15$ deg. In this case, the meridinal angle is measured positive from bottom to top with $\varphi = -90$ deg denoting the fuselage keel line and $\varphi = +90$ deg denoting the fuselage crown line. The value $\varphi = -0$ deg corresponds to the fuselage side just below the wing and $+0$ deg corresponds to the fuselage side just above the wing. The computational results for this comparison were all computed at $M_\infty = 0.82$ and $\alpha = 2$ deg using the fine three-zone grid described in Table 3. In each comparison there are two computed results, corresponding to TOL values of 0.1 and 1.0.

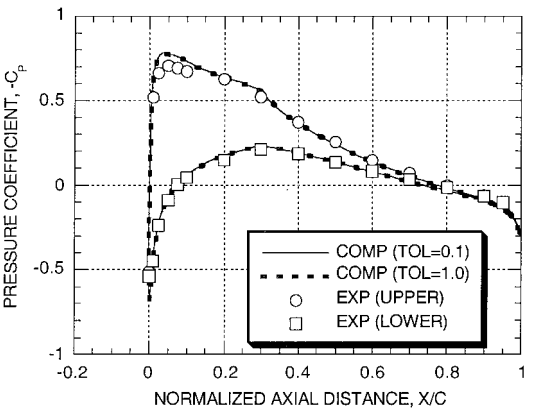
The computational results at all three wing stations are in good agreement with experiment, as seen from Fig. 5. A small disagreement between computation and experiment exists at the wing-root station ($2y/b = 0.17$), where lack of viscous flow modeling in the present approach is perhaps the cause. This case is supercritical with a small pocket of supersonic flow on the outboard, upper wing surface, but essentially has no shock wave. The two computed results (TOL = 0.1 and 1.0) overplot at all locations, indicating that the number of IGBPs using extrapolation has essentially no impact on solution accuracy. Convergence histories for these two cases (not shown) are also nearly identical. Each of these cases required 63 s of Cray C-90 computer time to reach 99.9% of the final converged lift and 73 s to achieve an average residual of 10^{-8}, which corresponds to an overall average residual reduction of 3.5 orders of magnitude. These times are for the flow solver iteration portion of the solution only. The computational time associated with all other aspects of the solution, including grid generation (biggest part of solution setup), I/O, PLOT3D output file generation, donor-cell search, hole cutting, metric computation, chimera interpolation coefficient computation, and flow initialization, was 17 s.



a) $2y/b = 0.17$



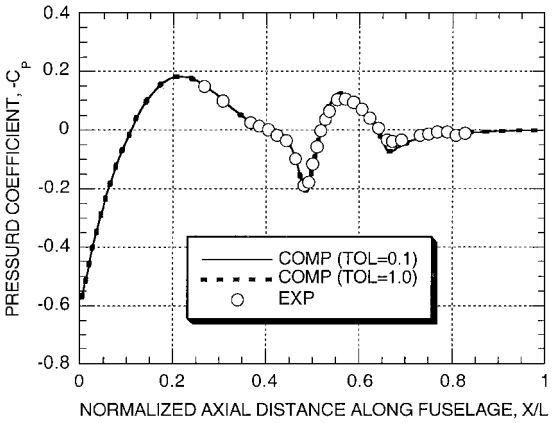
b) $2y/b = 0.40$



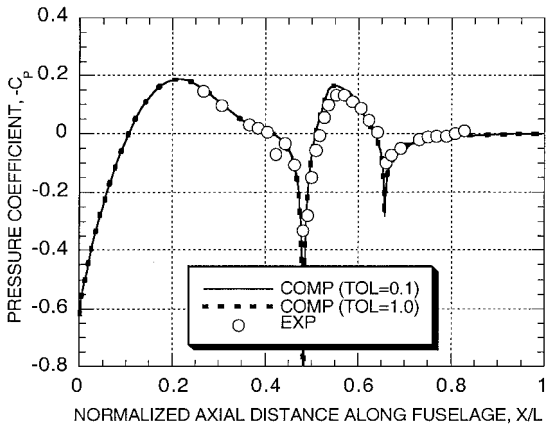
c) $2y/b = 0.85$

Fig. 5 Wing surface pressure coefficient comparisons showing the effect of TOL on solution accuracy, RAE wing A + body B2, $M_\infty = 0.82$, and $\alpha = 2$ deg.

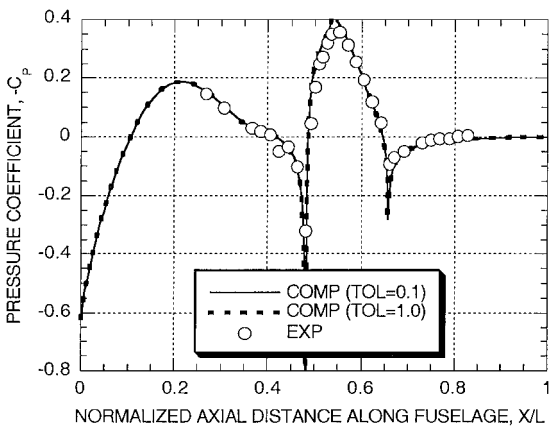
The computational results at all four fuselage stations are also in good agreement with experiment, as seen from Fig. 6. Also, the two computed results (TOL = 0.1/1.0) essentially overplot at all locations on the fuselage. Good agreement on the fuselage is particularly significant because the plots at $\varphi = \pm 0$ deg include segments from both the fuselage and wing grids, that is, they pass right through two chimera interfaces at about $x/L = 0.44$–0.47 and 0.70–0.73. In each overlap region, all available information from each grid and each solution is plotted. All computational solutions in these chimera overlap regions are in excellent agreement. The $\varphi = +15$ and -15 deg stations are also of special interest because they lie along the top and bottom chimera overlaps, respectively, between the wing and fuselage grids (from about $x/L = 0.44$ to 0.73). Agreement for these chimera overlap areas is also excellent.



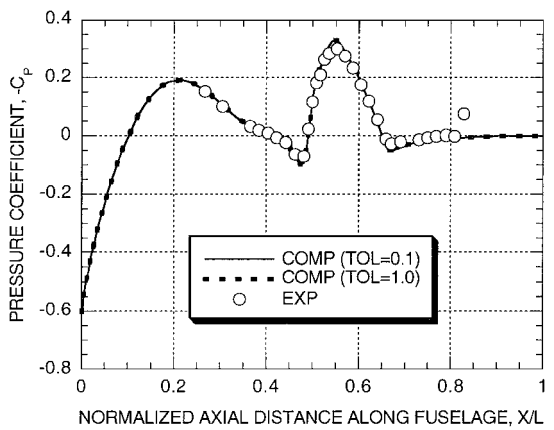
a) $\varphi = -15$ deg



b) $\varphi = -0$ deg



c) $\varphi = +0$ deg



d) $\varphi = +15$ deg

Fig. 6 Fuselage surface pressure coefficient comparisons between experiment and computation showing the effect of TOL on solution accuracy, RAE wing A + body B2, $M_\infty = 0.82$, and $\alpha = 2$ deg.

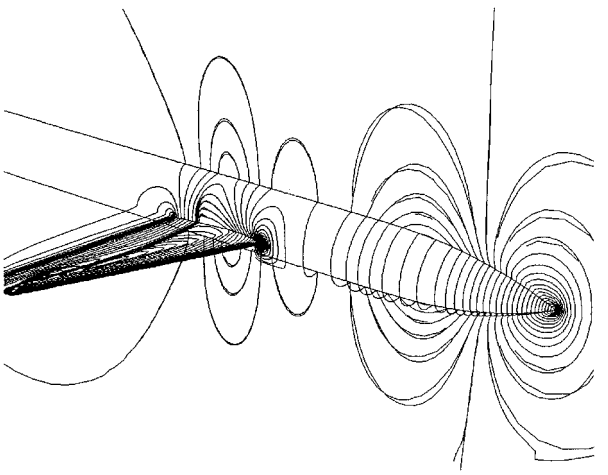


Fig. 7 Mach number contours about the RAE wing A with B2 fuselage on selected surfaces: $M_\infty = 0.9$, $\alpha = 0$ deg, fine-grid results, wing grid = 201 × 41 × 17, fuselage grid = 145 × 25 × 58, and outer grid = 101 × 41 × 62.

The last result consists of Mach number contours shown in Fig. 7 for the RAE wing-fuselage case just presented. Contours are plotted on the wing, fuselage, and symmetry plane surfaces for all grid zones that have solutions in these locations. This result was computed at $M_\infty = 0.9$ and $\alpha = 0$ deg with TOL = 0.1. As can be seen, the contours from the different solutions generally agree favorably with each other in regions of overlap. The amount of discrepancy between two overlapping contours is largest in regions where the grid refinement has the largest amount of discrepancy and where the grid is coarsest, for example, on the symmetry plane above and below the fuselage nose. This is an expected result of the chimera approach. On the fuselage just above the wing/fuselage juncture, the agreement is quite good. Both the wing and fuselage solutions predict the rapid expansion to supercritical flow followed by a moderate strength shock wave, which passes smoothly through both the wing and fuselage grids.

Conclusion

A new donor cell search algorithm suitable for chimera zonal grid applications has been presented. It has been tested in conjunction with a chimera full potential flow solver suitable for both wing and wing-fuselage configurations. The new search algorithm is extremely fast and simple, producing donor cells as fast as 60,000/s (Cray C90 single processor). One characteristic of the new algorithm is that it is approximate, that is, a donor cell may be approximated by a nearest neighbor cell. However, in the best situations this happens only about 3–4% of the time, with no impact on the solution accuracy or convergence efficiency. As a demonstration, calculations were performed using the chimera full potential solver for situations in which the number of nearest neighbor cells exceeded 70% of all donor cells. In this case, the resulting errors in the solution were negligible compared to expected truncation and formulational errors.

Acknowledgment

Computer time supplied by the Numerical Aerodynamic Simulation Systems Division at NASA Ames Research Center and by the Computational Aerosciences Project of the High Performance Computing and Communications Program is gratefully acknowledged.

References

[1]Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, edited by K. Ghia and U. Ghia, ASME FED-5, American Society of Mechanical Engineers, Fairfield, NJ, 1983, pp. 59–69.

[2]Benek, J. A., Buning, P. G., and Steger, J. L., "A Three-Dimensional Grid Embedding Technique," AIAA Paper 85-1523, July 1985.

[3]Buning, P. G., Chiu, I. T., Martin, F. W., Meakin, R. L., Obayashi, S., Rizk, Y. M., Steger, J. L., and Yarrow, M., "Flow Field Simulation of the Space Shuttle Vehicle in Ascent," *Proceedings of the Fourth International Conference on Supercomputing*, May 1989.

[4]Martin, F. W., and Slotnick, J., "Flow Computations for the Space Shuttle in Ascent Mode Using Thin-Layer Navier–Stokes Equations," *Applied Computational Aerodynamics*, edited by P. Henne, Vol. 125, Progress in Astronautics and Aeronautics, AIAA, Washington, DC, 1990, pp. 863–886.

[5]Murman, S. M., Schiff, L. B., and Rizk, Y. M., "Numerical Simulation of the Flow About an F-18 Aircraft in the High Alpha Regime," AIAA Paper 93-3405, Aug. 1993.

[6]Gee, K., Rizk, Y. M., and Schiff, L. B., "Forebody Tangential Slot Blowing on an Aircraft Geometry," *Journal of Aircraft*, Vol. 31, No. 4, 1994, pp. 922–928.

[7]Smith, M., Chawla, K., and Van Dalsem, W., "Numerical Simulation of a Complete STOVL Aircraft in Ground Effect," AIAA Paper 91-3293, Sept. 1991.

[8]Rizk, Y., Guruswamy, G., and Gee, K., "Numerical Investigation of Tail Buffet on F-18 Aircraft," AIAA Paper 92-2673, June 1992.

[9]Meakin, R. L., "Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulations," AIAA Paper 93-3350, July 1993.

[10]Atwood, C. A., and Van Dalsem, W. R., "Flowfield Simulation About the Stratospheric Observatory for Infrared Astronomy," *Journal of Aircraft*, Vol. 30, No. 5, 1993, pp. 719–727.

[11]Lijewski, L. L., and Suhs, N., "Time-Accurate Computational Fluid Dynamics Approach to Transonic Store Separation Trajectory Prediction," *Journal of Aircraft*, Vol. 31, No. 4, 1994, pp. 886–891.

[12]Atwood, C. A., "Computation of a Controlled Store Separation from a Cavity," AIAA Paper 94-0031, Jan. 1994.

[13]Kiris, C., Rogers, S., Kwak, D., and Chang, I.-D., "Computation of Incompressible Viscous Flows Through Artificial Heart Devices with Moving Boundaries," *Contemporary Mathematics*, Vol. 141, 1993, pp. 237–259.

[14]Meakin, R., and Street, R., "Simulation of Environmental Flow Problems in Geometrically Complex Domains, Part 2: A Domain-Splitting Method," *Computer Methods in Applied Mechanics and Engineering*, Vol. 68, June 1988, pp. 311–331.

[15]Atta, E. H., and Vadyak, J., "A Grid Interfacing Zonal Algorithm for Three-Dimensional Transonic Flows About Aircraft Configurations," AIAA Paper 82-1017, 1982.

[16]Holst, T. L., "Multizone Chimera Algorithm for Solving the Full-Potential Equation," *Journal of Aircraft*, Vol. 35, No. 3, 1998, pp. 412–421.

[17]Meakin, R., "On the Spatial and Temporal Accuracy of Overset Grid Methods for Moving Body Problems," AIAA Paper 94-1925, June 1994.

[18]Holst, T. L., "Numerical Solution of the Full Potential Equation Using a Chimera Grid Approach," NASA TM 110360, July 1995.

[19]Meakin, R., "A New Method for Establishing Intergrid Communication Among Systems of Overset Grids," AIAA Paper 91-1586, June 1991.

[20]Steger, J. L., and Baldwin, B. S., "Shock Waves and Drag in the Numerical Calculation of Isentropic Transonic Flow," NASA TN D-6997, Oct. 1972.

[21]Flores, J., Holst, T. L., Kwak, D., and Batiste, D. M., "A New Consistent Spatial Differencing Scheme for the Transonic Full Potential Equation," *AIAA Journal*, Vol. 22, No. 8, 1984, pp. 1027–1034.

[22]Thomas, S. D., and Holst, T. L., "A Consistent Spatial Differencing Scheme for the Transonic Full Potential Equation in Three Dimensions," NASA TM-86716, Dec. 1985.

[23]Chan, W. M., Chiu, I., and Buning, P. G., "User's Manual for the HYPGEN Hyperbolic Grid Generator and the HGUI Graphical User Interface," NASA TM 108791, Oct. 1993.

[24]Treadgold, A., Jones, A., and Wilson, K., "Pressure Distribution Measured in the RAE 8 ft ×6 ft Transonic Wind Tunnel on RAE Wing 'A' in Combination with an Axi-Symmetric Body at Mach Numbers of 0.4, 0.8 and 0.9," *Experimental Data Base for Computer Program Assessment*, AGARD-AR-138, May 1979.