Titre: Title:	A multi-block Euler method for the complete Challenger configuration
Auteur: Author:	Fassi Kafyeke
Date:	1993
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Kafyeke, F. (1993). A multi-block Euler method for the complete Challenger configuration [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/57981/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/57981/
Directeurs de recherche: Advisors:	
Programme: Program:	Non spécifié

UNIVERSITÉ DE MONTRÉAL

A MULTI-BLOCK EULER METHOD FOR THE COMPLETE CHALLENGER CONFIGURATION

par

Fassi KAFYEKE
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU GRADE DE PHILOSOPHIAE DOCTOR (Ph.D.)
GÉNIE MÉCANIQUE
Juillet 1993

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE

Cette thèse intitulée :

A MULTI-BLOCK EULER METHOD FOR THE COMPLETE CHALLENGER CONFIGURATION

présentée par : **Fassi KAFYEKE**

en vue de l'obtention du grade de **PHILOSOPHIAE DOCTOR (Ph.D)**

a été dûment acceptée par le jury d'examen constitué de :

M. TRÉPANIER Jean-Yves, Ph.D., président

M. PARASCHIVOIU Ion, Ph.D., membre, directeur de recherche

M. COLPIN Jean, Dr. Sc. Appl., membre

M. JONES Denis, M.Sc.A., membre

Je voudrais dédier cette thèse à ma femme

Solange et à mes enfants Terri et Jessie

SOMMAIRE

La modélisation sur ordinateur des écoulements transsoniques autour de configurations complexes est devenu un élément vital du développement de l'aérodynamique des avions modernes. Ce travail concerne la mise en oeuvre d'une méthodologie particulière de solution des équations d'Euler, qui régissent les évolutions de fluides compressibles non visqueux. Cette méthodologie a été développée afin de prédire de tels écoulements autour de configurations complexes d'avions. L'application principale est la modélisation de l'avion d'affaires Challenger CL-601 de Canadair dont une des caractéristiques est un couplage fort moteurs-voilure. Le travail a commençé par une étude de l'état actuel de l'aérodynamique numérique en milieu industriel et de l'application réservée aux équations d'Euler. L'étude a couvert la signification physique de ces équations, leurs propriétés mathématiques ainsi que la panoplie de méthodes numériques disponibles pour résoudre ces équations. La méthode de résolution des équations d'Euler proposée a été implémentée dans le programme **MBTEC** (**Multi-Block Transonic Euler Code**).

Dans le modèle numérique proposé, Les équations d'Euler sont utilisées sous leur forme instationnaire conservative, traduisant des bilans de masse, de quantité de mouvement et d'énergie. Cette forme instationnaire permet d'obtenir un système d'équations hyperboliques par rapport au temps qui peut être résolu comme un problème à valeurs initiales bien posé. L'écoulement de départ est une approximation arbitraire de la solution recherchée. La solution stationnaire est obtenue en intégrant les équations dans le temps jusqu'à convergence. Cette formulation permet également d'obtenir à l'avenir, des solutions instationnaires de l'écoulement. La solution numérique est obtenue en discrétisant ces équations selon une méthode de volumes finis explicite. La discrétisation s'appuie sur

un maillage curviligne quadrilatéral adapté aux contours de l'avion. Les dérivées spatiales sont discrétisées par des différences centrées. Le schéma est rendu suffisamment dissipatif par l'addition de termes de viscosité artificielle, qui permettent la capture d'ondes de choc et de discontinuités sans oscillation. L'intégration des équations dans le temps se fait par un algorithme de Runge-Kutta à cinq pas ayant d'excellentes propriétés de stabilité. Afin d'accélérer la convergence des calculs, un pas de temps local est sélectionné à chaque noeud du maillage, limité par la condition de Courant, Friedrichs et Lewy de l'écoulement local. Les inconnues du calcul sont les valeurs de la densité, de la vitesse et de l'énergie de l'écoulement à chaque noeud du maillage. Le volume de contrôle servant à l'application de la méthode en chaque noeud est la somme de tous volumes adjacents à ce noeud. Ce schéma possède de bonnes caractéristiques de précision et permet le calcul direct des variables à la surface de l'avion. La convergence des calculs est également améliorée par le remplacement du résidu en chaque noeud par une moyenne, calculée de manière implicite, des résidus des noeuds voisins. Finalement, une condition sur l'enthalpie, qui est une variable dans les calculs, permet un gain supplémentaire de taux de convergence.

L'application à des configurations tridimensionnelles complexes se fait à l'aide d'une stratégie multi-bloc particulière qui fait l'originalité de la méthode que nous avons développée. L'espace autour de l'avion est divisé en sous-domaines de forme simplement connexe et topologiquement équivalents à des cubes avec six faces et huit sommets. Les équations d'Euler sont résolues séquentiellement dans chacun de ces sous-domaines. L'information numérique est transférée de bloc en bloc grâce à une connectivité simple des noeuds des blocs qui se font face. Le programme est contraint à un seul type de condition aux limites sur chacune des faces de chaque bloc mais n'importe quelle face d'un bloc peut être associée à un

type arbitraire de conditions aux limites. On distingue : des conditions d'imperméabilité, de symétrie, d'entrée et de sortie du domaine fluide et surtout, des conditions d'entrée et de sortie des turboréacteurs. Les turboréacteurs à double flux et à large taux de dilution du Challenger influencent fortement l'aérodynamique de l'avion. Une modélisation précise de l'effet de ces moteurs était requise. Plusieurs méthodes d'implémentation des conditions de sortie ont été testées et une approche originale sélectionnée.

Une méthode progressive de validation de la stratégie multi-bloc a été utilisée qui garantit la consistence du modèle. En partant de configurations simples d'aile isolée, les conditions d'interface, la sommation des flux, l'addition de la viscosité artificielle ainsi que d' autres détails numériques ont été programmés de manière à assurer la convergence des calculs dans les cas les plus compliqués. La précision du calcul a été calibrée en comparant les résultats de calculs sur l'aile ONERA M6 avec les données expérimentales. Un test sur le modèle isolé du turboréacteur à double flux CF-34 de General Electric, qui sert à propulser le CL-601 Challenger, a servi à développer les conditions d'entrée et de sortie des moteurs.

L'application initiale au Challenger s'est faite en calculant des écoulements sur des configurations aile-fuselage et aile-fuselage-ailettes du CL-601 (40 blocs, 550,000 noeuds). Des comparaisons faites avec des données de soufflerie ont souligné la précision de la méthode. Finalement, des calculs ont été faits pour le Challenger CL-601 complet avec les turboréacteurs en fonctionnement. Ce modèle est constitué de 600 blocs et 1,571,580 noeuds. Une très bonne comparaison avec des résultats de soufflerie et d'essais en vol a démontré une fois de plus la précision des calculs, même dans les cas les plus complexes. L'application de la méthode au CL-601 Challenger est l'autre volet original de ce travail. Les résultats présentés ici sont les premières et seules prédictions obtenues à ce jour d'écoulements transsoniques autour d'une géométrie complète d'un avion de Canadair. Un autre avion complet, le CF-18 des forces canadiennes, a été modélisé en utilisant 108 blocs et 784,168 noeuds. Des comparaisons avec des résultats obtenus en soufflerie ont aussi confirmé la bonne précision de la méthode. Le programme utilise des maillages adaptés produits par des logiciels écrits spécifiquement pour ce projet. une interpolation algébrique est utilisée pour générer un maillage de départ qui est ensuite amélioré par résolution d'équations elliptiques suivant la méthode de Thompson.

La formulation originale des conditions d'interface des blocs ainsi que celle développée pour les conditions d'entrée et de sortie des réacteurs apportent la flexibilité, la fiabilité et la précision requises pour la conception d'avions à haute performance. La solution des équations d'Euler autour d'un avion génère une quantité et une qualité d'information importantes pour l'interprétation de l'aérodynamique d'un avion. Il reste à manipuler le programme pour y ajouter les avantages de temps de calcul plus réduits.

Avec l'autorisation de l'École Polytechnique, le texte de la thèse est rédigé en Anglais.

ABSTRACT

The numerical simulation of transonic flows around complex configurations has become a vital part of the aerodynamic design process of modern airplanes. The present work proposes a methodology for solving the Euler equations governing the flow of inviscid compressible fluids. This methodology was developed with the objective of predicting such flows around complex aircraft configurations. The main application sought is the modelling of Canadair's Challenger CL-601 aircraft featuring turbofan nacelle closely coupled with the wing. The work started with a study of the present status of Computational Fluid Dynamics in the aircraft industry and of the particular use made of the Euler models. The study then focussed on the physical and mathematical properties of the Euler equations and a review of possible methods for solving these equations was made. The particular Euler method proposed in this work was implemented in the **MBTEC** program (**Multi-Block Transonic Euler Code**).

In the MBTEC model, the Euler equations are cast in their unsteady conservative form, indicating balances of mass, momentum and energy. The unsteady formulation yields a system of equations which is hyperbolic with respect to time. This system can be solved as an initial value problem. Starting with a uniform flow approximation, the final steady flow is obtained by integrating the equations in time until convergence is reached. This formulation will allow in the future predictions of unsteady transonic flows as well. The numerical solution is obtained by discretizing the equations using a finite volume explicit method. The discretization is made using a structured body-fitted grid defined around the aircraft. The spatial derivatives are centrally differenced. The scheme is stabilized by the addition of artificial viscosity, leading to the definition of shock waves without

oscillations. The integration of the equations in time is made using a five-stage Runge-Kutta algorithm with excellent stability properties. To further improve the convergence of the computations, local time-stepping is used, advancing everywhere the solution at the largest time-step allowed by the Courant, Friedriechs and Lewy local condition. The unknowns are the values of density, the three components of velocity and the energy at every node of the grid. The control volume used to accumulate the fluxes at a node point is the sum of all the cells meeting at that node. This scheme has excellent properties of accuracy and allows direct calculation of the unknowns on the aircraft surface. The convergence of the computations is further enhanced by replacing the residual at each node point by an average of the residuals at the surrounding nodes calculated implicitly. Finally, enthalpy damping is used to further improve the convergence to steady state.

The application to complex three-dimensional configurations is by way of a particular multi-block technique which is original in the method that we have developed. The space around the aircraft is divided into subdomains or blocks, each simply connected and topologically equivalent to a cube, with six faces and eight corners. The Euler equations are solved in each block in sequence. The numerical information is transferred from block to block using an effective block interfacing scheme. The program uses one type of boundary condition per block face but any one of the following boundary conditions can be specified for any face of any block in the field : solid surface condition, symmetry condition, inflow and outflow conditions and, in particular, engine inlet and exhaust conditions. The high by-pass turbofan engines of the CL-601 are closely coupled to the wing and affect significantly its aerodynamics. An accurate modelling of the nacelle inlet and exhaust conditions was required. Several formulations of these boundary conditions were tested and an original approach proposed.

A progressive testing method was used in order to validate the multi-block code. Starting with simple isolated wing cases, we have adjusted the block interface boundary conditions, the convective flux accumulation, the addition of artificial viscosity in a multi-block grid and other numerical details to guarantee satisfactory convergence of the computations even in the most complex cases. The accuracy of the code was checked by comparing computed results with experimental data on the ONERA M6 wing. The program was then used to model the isolated nacelle ot the General Electric CF-34 engine powering the CL-601 Challenger. This test case was used to develop the inlet and exhaust boundary conditions.

The initial application to the Challenger was made by modelling a wing / body and a wing / body / winglet configurations with 40 blocks and more than 550,000 nodes. The computed results were compared with wind tunnel data, indicating good correlations. Finally, calculations were made for the complete, powered CL-601. This model, made of 600 blocks and 1,571,580 nodes yielded excellent comparisons with experimental data, demonstrating once more the excellent accuracy of the code in the most complex cases. The application of the method to the CL-601 Challenger is also an original part of this work. The results presented here are the first and only solutions ever made of transonic flow around complete Canadair aircraft configurations. Another full aircraft configuration, the Canadian Forces CF-18 clean aircraft, was also modelled, using 108 blocks and 784,168 nodes. Comparisons with wind tunnel data further confirmed the accuracy of the method. All computations were made using structured body-fitted grids generated by programs written at Canadair specifically for this project. Algebraic interpolations are used to obtain initial grids which are then smoothed using elliptic solvers, as proposed by Thompson.

The new formulation proposed here for the block interface conditions and that of the nacelle inlet and exhaust provide the required flexibility, reliability and accuracy required for regular use in the aircraft design cycle. Euler solutions provide a wealth of high quality data about the aerodynamics of an airplane. It remains to add the benefits of lower computing times to further reduce the turnaround times.

With the permission of École Polytechnique, this thesis is written in English.

RÉSUMÉ

Canadair, une compagnie de la division aéronautique de Bombardier, conçoit et fabrique des "jets d'affaires" à longue distance franchissable ainsi que des "jets régionaux" pour le transport de passagers sur courte distance. Ces avions volent en croisière à des vitesses variant entre Mach 0.70 et 0.90, donc en régime transsonique. Pour doter ces avions des meilleures performances possibles (faible traînée, stabilité et manoeuvrabilité) et leur permettre de se tailler une place sur un marché très compétitif, il faut leur donner des formes aérodynamiques optimales. La recherche de ces formes optimales peut se faire de façon empirique, par des essais systématiques en soufflerie, mais elle est mieux faite sur ordinateur, par une étude paramétrique de l'aérodynamique de diverses configurations.

Il faut donc être en mesure de prédire avec précision l'écoulement transsonique autour d'une géométrie donnée dans des conditions de vol spécifiques. Pour cela, il faut résoudre les équations de Dynamique des Fluides qui régissent l'évolution de l'air autour de l'avion. Ces équations, connues sous le nom d'équations de Navier-Stokes forment un système d'équations aux dérivées partielles non linéaires, de nature mixte, difficiles à résoudre. L'expérience indique qu'il est possible de négliger, en première approximation, la viscosité de l'air et obtenir encore des résultats significatifs. Les équations régissant des écoulements compressibles non visqueux sont connues sous le nom d'**équations d'Euler**. Ces équations permettent de représenter des écoulements rotationnels et non isentropiques et sont donc valides dans une enveloppe de vol comprenant des nombres de Mach fortement transsoniques (avec des ondes de choc fortes) et des angles d'incidence élevés (avec des écoulements tourbillonnaires). Jusqu'à ce jour, à Canadair, seuls des écoulements potentiels subsoniques pouvaient être résolus dans le cas de l'avion complet (grâce à des méthodes de panneaux).

Les calculs transsoniques autour de configurations complètes étaient faits au moyen du programme des petites perturbations KTRAN. Des formulations plus sophistiquées sont utilisées mais sont limitées à des cas plus simples (Potentiel Compressible FLO22 our Euler FLO67 pour des ailes isolées, Navier-Stokes pour des profils bi-dimensionnels).

Le but de ce travail est de développer une méthode précise, fiable et pratique, de résolution des équations d'Euler pour des géométries complexes d'avion de type CL-601 Challenger, avec voilure supercritique, ailettes marginales, turboréacteurs montés sur le fuselage et fortement couplés avec l'aile et un empennage en T. Plusieurs méthodes existent pour résoudre ces équations dans des cas plus ou moins simples, mais toutes celles qui pourraient s'appliquer au Challenger sont la propriété confidentielle de constructeurs aéronautiques concurrents et l'information sur ces méthodes, leurs succès et leurs échecs est partielle. Nous avons passé en revue ces diverses méthodes et nous les avons évaluées pour une application au Challenger. Nous avons ensuite sélectionné une formulation de support au développement d'un logiciel pour avion complet. Nous avons finalement implémenté une stratégie de résolution multi-block et un traitement des conditions d'entrée et de sortie des moteurs qui, parce qu'elles sont conçues pour des besoins très précis, sont originales à notre programme.

Une revue non exhaustive de différentes méthodes de résolution des équations d'Euler est reprise au début du texte : méthodes explicites et implicites, discrétisations centrées des opérateurs spatiaux, schémas à contre-courant. Cette revue couvre aussi les caractéristiques des solutions des équations d'Euler, les types de conditions aux limites permissibles et touche aux raisons du succès de certaines mèthodes et aux problèmes rencontrés par d'autres. Le choix de la formulation de base adoptée dans notre travail est expliqué par notre expérience précédente, des contraintes pratiques de disponibilité, et du degré de confiance requis des résultats.

Les équations d'Euler sont utilisées dans ce programme sous forme instationnaire conservative. Elles traduisent sous cette forme des bilans de masse, de quantité de mouvement et d'énergie. La solution recherchée est celle d'un écoulement stationnaire en régime établi. La forme instationnaire des équations est utilisée parce qu'elle permet d'obtenir un système d'équations hyperbolique par rapport au temps qui peut être résolu de manière explicite comme un problème à valeurs initiales bien posé. L'écoulement de départ est une approximation arbitraire de la solution recherchée (écoulement uniforme). La solution stationnaire est obtenue en intégrant les équations dans le temps jusqu'à ce qu'une convergence satisfaisante soit obtenue. Cette approche permet également d'obtenir, en utilisant un pas de temps uniforme dans tout le domaine, des solutions instationnaires de l'écoulement. La solution numérique est obtenue en discrétisant ces équations selon une méthode de volumes finis explicite. Dans cette approche, les équations sont intégrées afin d'exprimer les relations sous forme d'un équilibre de flux entrant et sortant à travers les frontières d'un volume de contrôle et de la variation des propriétés de l'écoulement à l'intérieur de ce volume. L'intégration des équations, qui est présentée dans le texte, transforme les équations aux dérivées partielles en équations différentielles ordinaires qui peuvent être résolues par des méthodes classiques. La discrétisation s'appuie sur un maillage curviligne quadrilatéral adapté aux contours de l'avion. Les dérivées spatiales sont discrétisées par des différences centrées. Le schéma est rendu suffisamment dissipatif par l'addition de termes de viscosité artificielle, qui permettent la capture d'ondes de choc et de discontinuités sans oscillation. L'utilisation des coefficients de viscosité artificielle se fait de la manière proposée par A.

Jameson et modifiée par E. Turkel. L'intégration des équations dans le temps se fait par un algorithme de Runge-Kutta à cinq pas de Jameson et Schmidt décrit dans le texte et ayant d'excellentes propriétés de stabilité. Afin d'accélérer la convergence des calculs, un pas de temps local est sélectionné à chaque noeud du maillage, limité par la condition de Courant, Friedrichs et Lewy de l'écoulement local. Cette condition spécifie que les calculs ne peuvent demeurer stables que si le domaine de dépendance du schéma numérique contient le domaine de dépendance des équations aux dérivées partielles de départ. Les inconnues du calcul sont les valeurs de la densité, des trois composantes cartésiennes du vecteur vitesse et de l'énergie à chaque noeud du maillage. Le volume de contrôle servant à l'application de la méthode est la somme de tous volumes adjacents au noeud. Ce schéma possède de bonnes caractéristiques de précision et permet le calcul direct des variables à la surface de l'avion. La méthode consiste donc à évaluer les flux convectifs et dissipatifs à travers l'enveloppe du volume de contrôle de chaque noeud et de réévaluer itérativement les valeurs des inconnues en chaque noeud par l'intégration d'équations différentielles ordinaires. La convergence des calculs est également améliorée par le remplacement du résidu en chaque noeud par une moyenne, calculée de manière implicite, des résidus des noeuds voisins. Finalement, une condition sur l'enthalpie, qui est une variable dans les calculs, permet un gain supplémentaire de taux de convergence. Cette méthode a fait ses preuves dans des applications simples telles que des ailes isolées, des ensembles aile-fuselage, etc. L'adaptation de cette méthode pour son application à des configurations tridimensionnelles complexes tout-à-fait arbitraires ainsi que le calcul de l'aérodynamique du CL-601 Challenger font toute l'originalité de notre contribution et constituent l'essentiel de cette thèse. L'écriture d'un code Euler pour cette configuration particulière a nécessité le développement d'une stratégie multi-bloc avec des conditions d'interface nouvelles, et une attention particulière aux conditions d'entrée et de sortie des moteurs dont une formulation originale est proposée. Le calcul a également nécessité la mise en oeuvre d'un système d'analyse complet, commençant par la modélisation géométrique sur CADAM et finissant par la visualisation des résultats sur des stations graphiques adéquates.

L'application à des géométries complexes se fait à l'aide d'une stratégie multi-bloc. Dans notre approche, l'espace autour de l'avion est divisé logiquement en sous-domaines de formes simplement connexes et topologiquement équivalents à des cubes avec six faces et huit sommets. Les équations d'Euler sont résolues séquentiellement dans chacun de ces sous-domaines. L'information numérique est transférée de bloc en bloc grâce à une connectivité simple des noeuds des blocs qui se font face. La nécessité d'organiser ce transfert sans détériorer la précision du modèle ni la convergence des calculs est une des difficultés technologiques principales que nous avons dû résoudre pour obtenir des résultats satisfaisants. La stratégie multi-bloc et sa corrolaire d'utilisation de maillages de type H-H ont en effet des répercussions importantes sur tous les aspects du résoluteur : l'évaluation et l'accumulation des flux convectifs, l'agencement des flux dissipatifs de façon à garantir dans tous les cas les propriétés de conservation du schéma numérique (pas d'addition ni de soustraction de masse, moment ou énergie), le calcul des valeurs propres locales entrant dans la définition des pas de temps et finalement l'intégration des équations dans chaque bloc en garantissant l'unicité de la solution aux interfaces. Les techniques que nous avons adoptées au terme de nombreux essais sont décrites dans le texte et constituent la trame de la méthode que nous proposons.

Le programme est actuellement contraint à un seul type de condition aux limites par face de chaque bloc. On distingue :

- Les conditions d'imperméabilité sur les surfaces solides.
- Les conditions aux frontières d'entrée et de sortie du domaine.
- Les conditions d'entrée de la soufflante du turboréacteur (débit d'entrée spécifié).
- Les conditions de sortie des gaz chauds du moteur et froids de la soufflante (rapport de pression totale et de température totale du moteur spécifiés. Une formulation nouvelle, s'inspirant des conditions imposées aux limites extérieures du domaine, a été testée avec succès et adoptée.
- Les conditions de symmétrie.
- Les conditions d'interface de blocs voisins, originales à notre approche.

Le résoluteur est structuré en modules distincts :

- Le module d'entrée de données reçoit les informations sur la topologie des blocs, les coordonnées des noeuds du maillage, les paramètres numériques et les conditions de vol.
- Le module d'interprétation de la topologie décode l'information reçue du mailleur, positionne les surfaces de l'avion et arrange la correspondance entre faces adjacentes de blocs voisins. Ce module décèle et étiquette tous les cas "spéciaux" qui nécessitent un traitement particulier.
- Le module de calcul géométrique évalue les volumes des cellules du maillage, les surfaces des faces et les composantes des normales aux faces de chaque cellule.
- Le module de gestion des blocs orchestre les calculs séquentiels dans chaque bloc, met en mémoire vive les données pertinentes au bloc actif et stocke sur un vecteur tampon les données des blocs inactifs.
- Le module de résolution des équations : calcul des pas de temps, des flux convectifs,

des flux dissipatifs, intégration des équations, application des conditions aux limites, moyennage des résiduels, évaluation des paramètres de convergence. Tous ces éléments sont largement affectés par la topologie H-H et la stratégie multi-bloc du programme.

- Le module de calcul des forces et des moments, essentiel dans un programme qui subdivise la surface de l'avion en zones appartenant à des centaines de blocs différents. Le programme reconstitue la surface extérieure de l'avion à partir des données topologiques. et calcule ensuite les coefficients de pression, les forces et les moments.
- Le module de sortie des résultats.

Une méthode progressive de validation de la stratégie multi-bloc a été utilisée, qui garantit la consistence du modèle. En partant de configurations simples d'aile isolée (1 bloc, 3 blocs, 6 blocs, 12 blocs), nous avons développé les conditions d'interface, la sommation des flux, la formulation de la viscosité artificielle et d'autres détails numériques pour assurer la convergence des calculs dans les cas les plus compliqués. Les conditions aux limites d'entrée et de sortie des moteurs, importantes en raison du couplage fort aile-nacelle du CL-601, ont été progressivement développées et testées sur une version largement modifiée d'un programme mono-bloc. Ces évaluations, conduites sur un modèle simple d'entrée d'air et un autre modèle simple de moteur à simple flux, ont permis de valider une approche originale des conditions de sortie des moteurs. La précision du calcul a été calibrée en comparant les résultats de calculs sur maillage fin (12 blocs, 293964 noeuds) de l'aile ONERA M6 avec des résultats expérimentaux. Les distributions de pression obtenues à $M = 0.84$ et $\alpha = 3.06^\circ$ démontrent qu'une précision nominale pour ce type de formulation est atteinte. Le programme a ensuite servi à modéliser le turboréacteur à double flux CF-34 de General Electric, qui sert à propulser le CL-601 Challenger. Ce test, utilisant 11 blocs et 250 000 noeuds, a servi à valider les conditions aux limites d'entrée et de sortie du moteur dans l'optique multi-bloc. Des résultats sont donnés pour un débit d'entrée correspondant à un régime de croisière ("Mass Flow Ratio" $MFR = 0.6838$) à Mach 0.74 et pour deux conditions de sortie des jets différentes. L'application initiale au Challenger s'est faite en modélisant des configurations aile-fuselage et aile-fuselage-ailettes du CL-601 (40 blocs, 550,000 noeuds). La comparaison des résultats avec des données de soufflerie démontre que le code multi-bloc permet d'obtenir les prédictions les plus précises jamais faites de l'écoulement non visqueux sur cet avion. On montre également que le couplage des résultats de ce calcul avec une évaluation interactive de couche limite promet de donner des résultats très près de la réalité. Ces bons résultats ont permis l'application finale à des configurations complexes d'avion. Des calculs ont été faits pour le Challenger CL-601 complet avec les turboréacteurs en fonctionnement. Ce modèle est constitué de 600 blocs et 1,571,580 noeuds. La formulation proposée permet d'obtenir une convergence adéquate des calculs dans tous les 600 blocs et une transparence à l'écoulement calculé dans les milliers d'interfaces entre blocs voisins. La complexité de la configuration introduit de nombreux points spéciaux (dégénérescence de faces entières en ligne ou en point, voisinage de surface solides, de coins, etc) qui sont autant de défis à la stabilité du calcul. La capacité d'obtenir de bons résultats dans ce cas constitue une validation nette de l'approche proposée. Ces résultats sont les premières et seules solutions jamais obtenues en écoulement transsonique sur la configuration complète du Challenger et la précision obtenue s'apparente très bien avec celle de solutions Euler obtenues par d'autres méthodes sur des configurations beaucoup moins complexes. Un autre avion complet, le CF-18 des forces canadiennes, a été modélisé en utilisant 108 blocs et 784,168 noeuds. Des comparaisons avec des résultats de soufflerie confirment une fois de plus la bonne précision de la méthode. Ce cas a permis de modéliser des écoulements à grand angle d'incidence et de capturer les structures tourbillonaires générées par les bords d'attaque tranchants des extensions avants de l'aile du CF-18.

Le code utilise des maillages adaptés produits par des logiciels écrits à Canadair spécifiquement pour ce projet. Une interpolation algébrique est utilisée pour générer un maillage de départ qui est ensuite amélioré par la résolution d'équations elliptiques suivant la méthode de Thompson. La topologie utilisée pour la génération du maillage multi-bloc est directement liée aux choix faits pour le programme Euler. Ces calculs ont été effectués sur l'ordinateur vectoriel CONVEX C-220 de CANADAIR à double CPU qui fait 50 MFLOPS par CPU. Les résultats ont été analysés sur un terminal graphique SILICON GRAPHICS R-4000-50 GTX. Le code est fortement vectorisé. Un calcul typique demande 300 à 500 iterations pour obtenir un niveau de convergence pratique des distributions de pression. Les calculs prennent entre 5 heures et 10 heures CPU sur l'ordinateur CONVEX, suivant la complexité de la géométrie et le nombre de noeuds du maillage.

La formulation multi-bloc que nous proposons compte allie la flexibilité, la fiabilité et la précision requises pour la conception d'avions à haute performance. Le code est utilisé sur une base régulière pour la conception du nouvel avion de Canadair, le Global Express. Il reste à manipuler le programme pour y ajouter les avantages de temps de calcul plus réduits, puis à introduire la modélisation des effets visqueux et de la turbulence pour posséder un code Navier-Stokes pour avion complet.

ACKNOWLEDGEMENTS

The developments described here were made possible by funds from Canadair and from the Defence Industrial Research program (D.I.R.) of the Department of National Defence of Canada. The experimental data for the CF-18 aircraft were provided by DND through the Canadair Defence Systems Division.

I wish to express my gratitude to my advisor, Professor Ion Paraschivoiu for giving me the opportunity of presenting this thesis. I thank him also for his patience and understanding during the course of the project. I am also indebted to Mr. Fotis Mavriplis, of Canadair, who encouraged me constantly in this project.

I would like to thank also the staff of Canadair Advanced Aerodynamics : Mr. Pat Piperni and Dr. F. Mokhtarian for the generation of the grids used in this work, Dr. François Pépin for his support with the TEX editor. I am also indebted to them for numerous technical discussions which resulted in improvements of the method. My thanks also to John Vazquez and Stéphane Major, who worked as a stagiaires with me on the testing of engine boundary conditions and on the comparison with experimental data.

Finally, I wish to thank the readers, Dr. J.Y. Trépanier, Dr. Jean Colpin and Mr. Denis Jones, for taking the time to read this thesis and making useful suggestions.

TABLE OF CONTENTS

Dédicace. . . . . **iv**

Sommaire. . . . . **v**

Abstract. . . . . **ix**

Résumé. . . . . **xiii**

Acknowledgements. . . . . **xxii**

Table of Contents. . . . . **xxiii**

List of Figures. . . . . **xxix**

List of Tables. . . . . **xxxvii**

List of Symbols. . . . . **xxxviii**

Chapter 1 : Introduction. **1**

1.1. Background Information. **1**

1.1.1. Transport aircraft aerodynamic design and analysis. **1**

1.1.2. Computational Fluid Dynamics development at Canadair. **2**

1.2. Objectives of the Euler Code Development. **7**

1.3. Outline. **10**

Chapter 2 : Solving Euler Equations : A Review. **11**

2.1. Formulation of the Euler Equations. **11**

2.2. Numerical Solutions of the Euler Equations. **14**

2.2.1. Lax-Wendroff Types of Space-centered Schemes. **18**

2.2.1.1. Explicit Schemes. **19**

a - First Order Schemes. **19**

b - Second Order Schemes. **20**

Lax-Wendroff Scheme. . . . 20

MacCormack Scheme. . . . 20

2.2.1.2. Implicit Schemes. . . . 22

2.2.2. Central Schemes with Independent Time Integration. . . . 23

2.2.2.1. Beam-Warming Schemes. . . . 23

2.2.2.2. LU Decomposition. . . . 26

2.2.2.3. Jameson's Multistage Scheme. . . . 26

2.2.3. Upwind Schemes. . . . 29

2.2.3.1. Basic Principles. . . . 29

2.2.3.2. Flux Vector Splitting Schemes. . . . 31

a - Steger and Warming flux splitting. . . . 32

b - Van Leer flux splitting. . . . 32

2.2.3.3. Godunov-type Schemes. . . . 33

a - Roe's approximate Riemann's solver. . . . 34

b - Osher's approximate Riemann solver. . . . 36

c - Other approximate Riemann solvers. . . . 36

d - Comparison of approximate Riemann solvers with other schemes. . . . 37

2.2.4. Second Order Upwind and High Resolution Schemes. . . . 38

2.2.5. Spectral Methods. . . . 40

2.2.6. Artificial Viscosity. . . . 40

2.2.7. Boundary Conditions for Inviscid Transonic Flow. . . . 44

2.2.7.1. Farfield Boundary Conditions. . . . 48

2.2.7.2. Wall Boundary Conditions. . . . 49

2.2.7.3. Kutta Condition. . . . 50

2.3. Applications of Euler Codes. **52**

2.3.1. Nacelles, inlets and exhaust nozzles. **52**

2.3.2. Propellers, rotors and propeller/wing/airframe interference. **53**

2.3.3. High Angle-of-Attack Aerodynamics. **54**

2.3.4. Analysis of Arbitrary Aircraft Configurations. **55**

2.3.5. Application of Euler codes to the design of configurations. **57**

2.4. Conclusions and Choice of a Methodology. **58**

2.4.1. The Mesh : Structured Grid versus Unstructured Grid. **58**

2.4.2. The numerical Method : Finite Volume versus Finite Element. **59**

2.4.3. The Space Discretization : Central versus Upwind Discretization **60**

2.4.4. Time Stepping : Explicit versus Implicit Solutions. **61**

2.4.5. Body-fitted Grids versus Rectangular Grids. **61**

2.4.6. Single-Block versus Multi-Block. **62**

2.4.7. Cell-Vertex versus Cell-centered Schemes. **63**

2.4.8. Formulation of the Proposed Euler Method. **64**

Chapter 3 : Geometry Modelling and Grid Generation. **65**

3.1. Modelling of the Aircraft Geometry. **65**

3.2. Grid Generation. **66**

3.2.1. Algebraic Grid Generation. **66**

3.2.2. Elliptic Grid Generation. **67**

3.2.3. Multi-Block Grid Generation. **68**

3.2.3.1. Topological Block Decomposition. **69**

3.2.3.2. Physical Block Decomposition on CADAM. **70**

3.2.3.3. Automatic Geometry and Topology extraction from CADAM **70**

3.2.3.4. Multi-Block Algebraic Grid Generation. 70

3.2.3.5. Multi-Block Elliptic Grid Generation. 71

3.2.4. Multi-Block Grid Generation Package. 72

Chapter 4 : Equations, Discretization, Time Integration. 78

4.1. Flow Equations. 78

4.2. Discretization of the Euler Equations. 78

4.3. Time Integration. 81

4.4. Convergence Acceleration Techniques. 82

4.4.1. Residual Smoothing. 82

4.4.2. Enthalpy Damping. 83

Chapter 5 : Multi-Block Boundary Conditions. 84

5.1. Farfield boundary conditions. 84

5.2. Solid Surface and Symmetry Boundary Conditions. 87

5.3. Engine Inlet Boundary Conditions. 88

5.3.1. Nacelle Inlet Grid. 89

5.3.2. Symmetry Boundary Condition at k=KL. 91

5.3.3. Inlet Boundary Conditions. 92

5.3.4. Singular Line on the Axis of Symmetry. 94

5.3.5. Domain Inflow Boundary Conditions. 95

5.3.6. Testing of Inlet Model. 95

5.4. Engine Exhaust Boundary conditions. 101

5.4.1. Complete Nacelle Grid. 101

5.4.2. Far Field Boundary Conditions. 101

5.4.3. Singular Line Downstream of the Nacelle. 102

5.4.4. Engine Exhaust Boundary Conditions. 102

5.4.5. Testing of the Complete Nacelle Model. 111

5.5. Internal Block Face Boundary Conditions. 125

5.6. Singular lines and Singular Points. 127

Chapter 6 : Multi-Block Algorithm. **131**

6.1. Topology Definition. 131

6.2. Algorithm Structure. 135

6.2.1. Multi-Block Input Module. 135

6.2.2. Multi-Block Data Interpretation Module. 138

6.2.3. Geometric Data Computation Module. 144

6.2.4. Multi-Block data Management Module. 145

6.2.5. Multi-Block Flow Solver Module. 148

6.2.5.1 Calculation of the Permissible Time Step. 151

6.2.5.2 Calculation of the Convective Fluxes. 155

6.2.5.3 Calculation of the Artificial Viscosity Fluxes. 156

6.2.5.4 Time Integration. 158

6.2.5.5 Application of the Boundary Conditions. 158

6.2.5.6 Residual Averaging. 159

6.2.5.7 Convergence Monitoring. 161

6.2.6. Forces and Moments Module. 162

6.2.7. Output Module. 164

Chapter 7 : Applications. **167**

7.1. Validation of the Multi-Block Procedures. 167

7.2. ONERA M6 Wing Test Case. 177

7.3. Challenger Forward Fuselage Test Case. 182

7.4. Challenger Wing/Body/Winglet Test Case. 183

7.5. CF-34 Transonic Nacelle Test Case. 184

7.6. Challenger Wing-Body Configuration. 186

7.7. Challenger Complete Configuration. 187

7.8. CF-18 Complete Configuration. 218

7.9. Computing Costs, Pre and Post Processing. 229

Chapter 8 : Conclusions and Further Work. 231

8.1. Conclusions. 231

8.2. Further Work. 233

Chapter 9 : References. 235

LIST OF FIGURES

Figure 1 : Canadair CL-601 Challenger configuration.

Figure 2 : CADAM model of the CL-601 Challenger aircraft.

Figure 3 : CADAM model of the CF-34 turbofan nacelle.

Figure 4 : Conventional numbering of block faces, edges and corners.

Figure 5(a) : Topological block decomposition for the Challenger wing/body/winglet.

Figure 5(b) : Physical block decomposition for the Challenger wing/body/winglet.

Figure 5(c) : View of an intermediate grid generated for the Challenger wing/body/winglet.

Figure 6 : Topology of the grid used to validate nacelle inlet conditions.

Figure 7 : Topology of the C-H grid used for isolated wings.

Figure 8 : Overall view of the grid generated for nacelle inlet calculations.

Figure 9 : Close view of the grid generated for nacelle inlet calculations.

Figure 10 : Pressure distribution in the plane of symmetry of the nacelle inlet at M = 0.75, $\alpha = 0^\circ$ and $MFR = 0.75$ (Full discretization of the equations on the axis of symmetry.

Figure 11 : Streamtube calculated in front of the nacelle inlet at M = 0.60, $\alpha = 0^\circ$ and $MFR = 0.25$.

Figure 12 : Streamtube calculated in front of the nacelle inlet at M = 0.30, $\alpha = 0^\circ$ and $MFR = 2.0$.

Figure 13 : Topology of the grid used for the complete nacelle calculations.

Figure 14 : Overall view of the grid generated for complete nacelle calculations.

Figure 15 : Close view of the grid generated for complete nacelle calculations.

Figure 16 : Pressure distribution in the plane of symmetry of the complete nacelle at M = 0.85, $\alpha = 0^{\circ}$ and $MFR = 0.75$ for a supersonic exhaust with $M_{jet} = 1.15, TPR = 1.3$ and $TTR = 2.0$.

Figure 17 : Total pressure distribution in the plane of symmetry of the complete nacelle at M = 0.85, $\alpha = 0^{\circ}$ and $MFR = 0.75$ for a supersonic exhaust with $M_{jet} = 1.15, TPR = 1.3$ and $TTR = 2.0$.

Figure 18 : Velocity vectors calculated in the plane of symmetry of the complete nacelle at M = 0.85, $\alpha = 0^{\circ}$ and $MFR = 0.75$ for a supersonic exhaust with $M_{jet} = 1.15, TPR = 1.3$ and $TTR = 2.0$.

Figure 19 : Pressure distribution in the plane of symmetry of the complete nacelle at M = 0.74, $\alpha = 0^{\circ}$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

Figure 20 : Temperature distribution in the plane of symmetry of the complete nacelle at M = 0.74, $\alpha = 0^{\circ}$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

Figure 21 : Iso-Mach lines calculated in the plane of symmetry and on the cowl of the complete nacelle at M = 0.74, $\alpha = 0^{\circ}$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

Figure 22 : Pressure distribution calculated along the top line of the complete nacelle at M = 0.74, $\alpha = 0^{\circ}$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

Figure 23 : Mach plot showing vortical flow in the plane of symmetry of the

complete nacelle at M = 0.75, $\alpha = 8^\circ$ and $MFR = 0.375$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.30$ and $TTR = 2.0$.

Figure 24 : Pressure distribution calculated along the top line of the complete nacelle at M = 0.75, $\alpha = 8^\circ$ and $MFR = 0.375$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.30$ and $TTR = 2.0$.

Figure 25 : Pressure distribution in the plane of symmetry near the exhaust of the complete nacelle showing shock and expansion patterns. Solution calculated at M = 0.75, $\alpha = 8^\circ$ and $MFR = 0.375$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.30$ and $TTR = 2.0$.

Figure 26 : Current block, extended block and halo grid planes.

Figure 27 : Block boundary interfaces.

Figure 28 : Block interface boundary conditions.

Figure 29 : H-H grid topology used in the multi-block code.

Figure 30 : Discretization cells used in the multi-block code. (a) Standard cell; (b) Solid boundary cell; (c) Inflow boundary cell.

Figure 31 : Wing cut options at block boundary interfaces.

Figure 32 : Special problems of block interfacing near the wing leading edge.

Figure 33 : Control volumes used in the multi-block code.

Figure 34 : Topological block decomposition of the space around an isolated wing used in the 12-block test case.

Figure 35 : Topology definition of the 3-block test case.

Figure 36 : Isobars on the floor bump (j=1) and velocity vectors calculated in. the plane of symmetry (k=1) and upstream and downstream of the floor bump. 3-block test case, M = 0.80 and $\alpha = 0^\circ$.

Figure 37 : Topology definition of the 6-block test case.

Figure 38 : Sectional pressure distribution calculated at midspan on the NACA L51F07 wing at M = 0.84 and $\alpha = 0^{\circ}$ using 6 blocks.

Figure 39 : Upper surface pressure distribution calculated on the NACA L51F07 wing at M = 0.84 and $\alpha = 0^{\circ}$ using 6 blocks.

Figure 40 : Upper surface pressure distribution calculated on the NACA L51F07 wing at M = 0.94 and $\alpha = 3^{\circ}$ using 12 blocks.

Figure 41 : Test cases used to validated various parts of the multi-block Euler code.

Figure 42 : Convergence characteristics of the multi-block Euler code. Evolution of the maximum residual and the number of supersonic points. MBTEC calculations on ONERA M6 wing M = 0.90 and $\alpha = 0^{\circ}$, (6 blocks).

Figure 43 : Planform isobars for the ONERA M6 wing. M = 0.84 $\alpha = 3.06\ ^{\circ}$.

Figure 44 : Pressure distributions for the ONERA M6 wing. M = 0.84 $\alpha = 3.06\ ^{\circ}$.

Figure 45 : Pressure distributions for the ONERA M6 wing. M = 0.84 $\alpha = 3.06\ ^{\circ}$. Comparison of single-block FLO67 results with MBTEC 12-block results.

Figure 46 : View of the grid used to compute solutions on the Challenger forward fuselage.

Figure 47 : Topology definition of the Challenger forward fuselage grid using 8 blocks. The inboard faces of blocks 2 and 3 have degenerated to single lines in the plane of symmetry.

Figure 48 : Surface pressure distribution calculated on the Challenger forward fuselage at M = 0.82 $\alpha = 1.5^{\circ}$ using 8 blocks.

Figure 49 : Velocity distribution calculated in the plane of symmetry for the Challenger forward fuselage at M = 0.82 $\alpha = 1.5^{\circ}$ using 8 blocks.

Figure 50 : Pressure distribution calculated in the plane of symmetry for the Challenger forward fuselage at M = 0.82 $\alpha = 1.5^{\circ}$ using 8 blocks.

Figure 51 : Surface isobars for the Challenger wing/body/winglet at M = 0.82 $\alpha = 1.5\ ^{\circ}$. 40-block MBTEC solution (100+100+500 iterations).

Figure 52 : Pressure distributions calculated on the Challenger wing/body/winglet configuration at M = 0.82 $\alpha = 1.5\ ^{\circ}$.
40-block MBTEC solution (100 + 100 + 500 iterations).
(a) Fuselage top and bottom centerlines;
(b) wing break section, station 148;
(c) Winglet mid-span station, waterline 80.

Figure 53 : View of the 11-block body-fitted grid of the isolated CF-34 nacelle. The block boundaries are outlined on the drawing.

Figure 54 : Pressure distribution in a vertical section of the CF-34 turbofan nacelle. 11-block solution at M = 0.74 and $\alpha = 0^{\circ}$.
Case (1) engine operating conditions.

Figure 55 : OMNI3D display of isobars on the CF-34 turbofan nacelle showing the expansion and compression cells. 11-block MBTEC solution.
M = 0.74 and $\alpha = 0^{\circ}$. Case (1) engine operating conditions.

Figure 56 : Pressure distribution in a vertical section of the CF-34 turbofan nacelle. 11-block MBTEC solution at M = 0.74 and $\alpha = 0^{\circ}$.
Case (2) engine operating conditions.

Figure 57 : SSV-4D display of Mach numbers in the field around the CF-34

turbofan nacelle. 11-block MBTEC solution at M = 0.74 and $\alpha = 0^{\circ}$. Case (2) engine operating conditions.

Figure 58(a) : Challenger wing/body upper surface grid.

Figure 58(b) : Challenger wing/body lower surface grid.

Figure 59 : Challenger wing/body surface isobars 40-block solution at M = 0.819 and $\alpha = 1.494^{\circ}$.

Figure 60 : Challenger wing/body chordwise pressure distributions at six spanwise stations. M = 0.819 and $\alpha = 1.494^{\circ}$.

Figure 61 : Challenger wing/body single-block Euler/boundary layer solution. M = 0.819 and $\alpha = 1.494^{\circ}$.
(a) C-H grid plane K=1 covering body and plane of symmetry
(b) Chordwise pressure distributions at six spanwise stations
(Mokhtarian [108])

Figure 62(a) : Block boundaries outlined on the upper surface of the CL-601 aircraft.

Figure 62(b) : Block boundaries outlined on the lower surface of the CL-601 aircraft.

Figure 63 : Multi-block grid for the Challenger CL-601 full aircraft

Figure 64(a) : Challenger CL-601 aircraft upper surface grid (intermediate)

Figure 64(b) : Challenger CL-601 aircraft lower surface grid (intermediate)

Figure 65 : Challenger complete configuration surface isobars at M = 0.799 and $\alpha = 0.93^{\circ}$. (a) upper surface isobars (b) lower surface isobars

Figure 66 : Challenger complete configuration upper surface isobars at M = 0.799 and $\alpha = 0.93^{\circ}$. With outlined block edges.

Figure 67 : CL-601 complete configuration chordwise pressure distributions at six spanwise stations. M = 0.799 and $\alpha = 0.93^{\circ}$.

Figure 68 : CL-601 complete configuration; fuselage centerline pressure distributions. M = 0.799 and $\alpha = 0.93^\circ$.

Figure 69 : CL-601 complete configuration; pressure distribution at 45% semi-span of horizontal tailplane. M = 0.799 and $\alpha = 0.93^\circ$.

Figure 70 : CL-601 complete configuration; pressure distribution at mid-span of nacelle pylon. M = 0.799 and $\alpha = 0.93^\circ$.

Figure 71 : CL-601 complete configuration; pressure distribution at buttline 88 on nacelle centerline and wing. M = 0.799 and $\alpha = 0.93^\circ$.

Figure 72 : CL-601 complete configuration; pressure distribution at buttline 88 on nacelle centerline surfaces. M = 0.799 and $\alpha = 0.93^\circ$.

Figure 73 : CL-601 complete configuration; surface streamlines predicted from Euler solution at M = 0.799 and $\alpha = 0.93^\circ$.
(a) upper surface streamlines; (b) lower surface streamlines.

Figure 74 : CL-601 complete configuration chordwise wing pressure distributions at six spanwise stations. M = 0.76 and $\alpha = 2.10^\circ$.

Figure 75 : CL-601 complete configuration chordwise winglet pressure distributions at two spanwise stations. M = 0.76 and $\alpha = 2.10^\circ$.

Figure 76 : CL-601 complete configuration lift curve slope at Mach 0.76

Figure 77 : CL-601 complete configuration at Mach 0.76 and $\alpha = 2.10^\circ$.
Pressure distributions on fuselage near the nacelle.

Figure 78 : CL-601 complete configuration at Mach 0.76 and $\alpha = 2.10^\circ$.
Pressure distributions near the nacelle pylon centerline.

Figure 79 : CL-601 complete configuration at Mach 0.76 and $\alpha = 2.10^\circ$.
Convergence characteristics in blocks above the wing :

(a) average residual, (b) number of supersonic points.

Figure 80 : Drag rise curve for the CL-601 wing-body configuration at $C_L = 0.6$.

Figure 81(a) : CADAM physical domain decomposition for the CF-18.

Figure 81(b) : close-up view of domain decomposition and geometry modelling for the CF-18 aircraft.

Figure 82 : Multi-block grid for the CF-18 clean aircraft.

Figure 83(a) : CF-18 aircraft upper surface grid (intermediate).

Figure 83(b) : CF-18 aircraft lower surface grid (intermediate).

Figure 84 : CF-18 clean configuration surface isobars at M = 0.95 and $\alpha = 2.79^\circ$. (a) upper surface isobars (b) lower surface isobars.

Figure 85 : CF-18 clean configuration upper surface isobars at M = 0.95 and $\alpha = 2.79^\circ$. With outlined block edges.

Figure 86 : CF-18 clean configuration chordwise pressure distributions at six spanwise stations. M = 0.95 and $\alpha = 2.79^\circ$.

Figure 87 : CF-18 clean configuration surface isobars at M = 0.60 and $\alpha = 16.9^\circ$. (a) upper surface isobars (b) lower surface isobars.

Figure 88 : CF-18 clean configuration normalized stagnation pressures showing vortices emanating from the leading edge extensions. M = 0.60 and $\alpha = 16.9^\circ$.

Figure 89 : Canadair CFD computing setup

LIST OF TABLES

Table 1 : Number of Boundary Conditions.

Table 2 : Definition of Block Index Limits.

Table 3 : CL-601 complete configuration results at Mach 0.76.

LIST OF SYMBOLS

a : Wave speed.
C : Speed of sound.
$D(W)$: Discretized dissipative fluxes
E : Normalized total energy per unit volume
f : Flux function.
F_ℓ : Mean flux across face ℓ of control volume.
$\vec{F}, \vec{G}, \vec{H}$: Vectors of mean convective fluxes
H : Normalized total enthalpy
i, j, k : Curvilinear grid indices
J : Jacobian
k : Coefficient of thermal conductivity
M : Mach number
p : Normalized pressure
$Q(W)$: Discretized convective fluxes
q_i : Component of the heat conduction term.
$R(W)$: Residual of the discretized equation
$\vec{R}, \vec{S}, \vec{T}$: Flux vectors representing viscous terms.
Re : Reynolds number
S : Surface of the computational cell outer boundary
S_ℓ : Directed area of face ℓ of control volume
s : Entropy per unit mass.
T : Normalized temperature.

t : Time independent variable

u : Dependent variable

u, v, w : Normalized Cartesian components of velocity in the x, y and z directions

V : Computational cell volume or flow velocity

$\vec{v}$: Velocity vector with Cartesian components u, v and w

$\vec{W}$: Vector of conservative variables $\rho, \rho u, \rho v, \rho w, \rho E$ used by the Euler code

x, y, z : Cartesian coordinates in the three spatial directions

α : Angle of attack or incidence.

γ : Ratio of specific heats

$\partial\Omega$: Surface bounding the fluid domain

$\delta^{(2)}$: Second difference operator

Δ : Symbol for variation of a quantity

Δt : Time step.

Δx : Grid spacing in x-direction.

$\varepsilon^{(2)}, \varepsilon^{(4)}$: Second and fourth order adaptive artificial viscosity coefficients

$\vec{\zeta}$: Vorticity vector

$\kappa^{(2)}, \kappa^{(4)}$: Second and fourth order constant artificial viscosity coefficients

μ : Coefficient of dynamic viscosity.

ρ : Density

σ : Courant number $= a\Delta t/\Delta x$

τ_{ij} : Component of the viscous stress.

τ : ratio $\Delta t/\Delta x$

ν : Normalized second difference of pressure used as shock sensor

Ω : Three-dimensional fluid domain

Subscripts

e : values extrapolated from inside the flow domain

f : final values calculated at domain boundaries

i, j, k : Mesh point locations in x,y,z directions

j : values corresponding to specified jet exhaust boundary conditions

t : Tangential component or derivative w.r.t. time.

n : Normal component.

o : Stagnation values.

x, y, z : Partial differentiation w.r.t. x,y,z or component in x,y,z direction

0 : values corresponding to specified freestream boundary conditions

∞ : Freestream values

Superscripts

n : Iteration level or time level

CHAPTER 1 : INTRODUCTION

1.1 BACKGROUND INFORMATION

1.1.1 Transport Aircraft Aerodynamic Design and Analysis

The aerodynamic design of airplanes, once a mostly experimental exercise, has become a process involving the application of advanced theoretical methods as well as experimental testing. In the highly competitive world of the Air Transport Industry, aircraft manufacturers have been thriving to develop and use methods which enable them to put together quickly configurations which are optimal for a required mission (i.e. the safe transport of passengers and cargo over various distances at minimum cost). For this purpose, the aerodynamicist is called upon to achieve two important objectives :

1) The identification of the shapes of airfoils, wings and complete aircraft configurations giving the best combination of lift, drag and stability and control characteristics.
2) The accurate prediction of these characteristics in support of performance guarantees, well before the aircraft is built and flown.

To achieve these objectives, the aerodynamicist can rely on theoretical analyses (analytical or numerical) and on the results of appropriate wind tunnel tests. In the past, the usual procedure was to define, from theory and previous experience, several promising configurations and then test them in the tunnel in order to select the optimal one. The tunnel data was also a reliable source of information for predicting the aerodynamics of the full scale airplanes, which were relatively small and flew at slow subsonic speeds. However, the recent growth of the size and speed of modern airplanes flying at high Mach numbers and Reynolds numbers has taken them beyond the range for which the tunnel data can be relied upon with complete confidence. The increased competition is also calling for lower development costs and dictating fewer, more focussed, wind tunnel tests. The emphasis is therefore shifting towards more and more theoretical analyses. In the transonic flow range, in order to obtain information on the non linear aerodynamic characteristics of aircraft it is necessary to resort to elaborate Computational Fluid Dynamics (CFD) methods.

1.1.2 Computational Fluid Dynamics Development at Canadair

The CFD method presented here was developed as a good example of collaboration between government, university and industry. The code was developed in order to satisfy a specific need of Canadair, a division of Bombardier : the accurate modelling of inviscid transonic flows about Business Aircraft configurations similar to the CL-601 Challenger, featuring advanced supercritical wings, a wide-body fuselage, a T-tail and fuselage mounted high by-pass turbofan engines (figure 1). The development of this CFD method was supported in part by the Department of National Defence of Canada which saw an interest in the modelling of vortical flows around the Canadian forces CF-18 airplane at high angles of incidence. Finally, an agreement between Canadair and École Polytechnique de Montréal was reached which made this effort part of a Ph.D. program granting the author access to the expertise and technical guidance available at the university.

In order to understand the relevance of the Euler code in the overall aircraft design cycle of Canadair, it is useful to review the past and present CFD capability of the company. Canadair uses computational aerodynamic methods to improve the aerodynamic performance of its present products (Business Jets and Regional Jets with aft fuselage mounted nacelles) and to design new aircraft that can compete successfully in the market. These methods can be used to study a wide range of configuration options during the preliminary design phase of an aircraft, thus increasing the prospect of identifying the optimum design. This approach is more cost effective than wind tunnel testing. These codes are also used to study problems involving complex flow physics that cannot be simulated realistically or in sufficient detail in the wind tunnel. In addition, many codes developed at Canadair with the support of the Department of National Defence (DND) of Canada have been used to obtain aerodynamic data on military aircraft. Through two decades of continuing development, the company has developed or acquired a large array of computational codes for the design and analysis of aerodynamic configurations in subsonic and transonic flow. For the analysis of complete aircraft configurations in 3-D subsonic flow, panel methods are used. Three such codes are available at Canadair. The WBAERO code is the original, well established, low order code [1]. In 1983, the author initiated the development of the CANAERO code, an improved version of WBAERO, using vortex sheets instead of vortex lines to represent the lifting characteristics of wings [2]. The code was later further developed to include the relaxation of wing and body wakes and the modelling of separated body vortices [3]. A third panel code, VSAERO, purchased from Analytical Methods Inc. (AMI) of Redmond, Washington (USA), is also used. VSAERO has a powerful post processing suite (OMNI3D) allowing color display of pressures, wakes, velocity vectors and streamlines on a graphics workstation.

Figure 1 : Canadair CL-601 Challenger configuration.

The main thrust in the use of computational tools for transonic aerodynamics came with the initiation of the Challenger Executive Jet program, in 1976. Several codes were brought in the company for the analysis and design of transonic airfoil sections and wings. The TRANDES code for wing section analysis and design uses a Transonic Small Perturbation (TSP) formulation [4], whereas the BGK code solves the Transonic Full Potential equation [5]. Both codes have the inherent limitation to quasi-isentropic and irrotational flows. The BGK program has an interactive compressible boundary-layer calculation of the Nash-MacDonald type. The program was recently coupled with an optimizer routine to produce an airfoil design code [6]. In this approach, the coordinates of a starting airfoil are automatically modified by the program in an iterative manner to produce a specified pressure distribution. The design of three-dimensional wings has relied mostly on the well known FLO22 Full Potential non-conservative code written by A. Jameson of Princeton University [7]. This code, together with the BGK 2D code, was used in 1976 for the design of the Challenger. Its value was clearly illustrated in the successful design of the Challenger advanced supercritical wing. The program, originally developed for isolated wings in transonic flows, was modified at Canadair to account for wing/body interactions. It gave then reliable predictions of inviscid pressure distributions and drag rise characteristics for wing/body combinations. However, since this code does not take into account the close-coupled nacelles, the effects of these nacelles had to be investigated experimentally. More recently, the modified FLO22, the BGK program and an optimizer code, FLO6QNM, were used by the author in the design of the wings of the Canadair Regional Jet (RJ) [8]. Wind tunnel and flight test results have confirmed the new design and the aircraft has recently entered commercial service.

The first major step towards the modelling of transonic flow fields for complete aircraft configurations was taken in 1984, when the Department of National Defence of Canada (DND) contracted Canadair to develop a Transonic Small Disturbance code. The initial code, called KTRAN was developed by the author who applied it to wing/body/pylon/store configurations in 1986 [9]. Later, an enhanced version was produced, which can model complex military aircraft configurations such as the CF-18 with under-wing pylons and stores [10]. It can also model commercial airplanes with a fuselage mounted powered nacelle and winglets [11]. The code uses simple Cartesian grids. A grid embedding technique is used to provide flow details near specific components. A major advantage of the KTRAN code is the modest amount of computer resources (CPU time, memory capacity) needed to calculate the flow around realistic configurations. Because of the Small Perturbation formulation, the KTRAN program is restricted to a certain range of transonic Mach numbers and to configurations in symmetrical flight at relatively small angles of incidence. The program is also restricted to flows properly described by the Full Potential assumptions, i.e. irrotational and quasi-isentropic flows. With the advent of more powerful computers and the development of fast algorithms to solve the equations of fluid motion, the emphasis in Computational Fluid Dynamics has shifted towards the use of Euler and Navier-Stokes codes. The Euler equations represent a fluid with zero viscosity. They can be seen as the limit of the Navier-Stokes equations for vanishing viscosity (infinite Reynolds number). They are a better model of inviscid flow than the Full Potential equation because of their ability to represent rotational effects and entropy variations in the flow field. Unlike the Full Potential equations, they are not restricted to transonic flows with weak shock waves but can be used for flows with strong shock waves and highly supersonic flows. They can provide, for instance, a better prediction of inviscid airfoil characteristics at off-design conditions. The Euler equations allow also the modelling of vortical flow fields such as those which develop around modern fighter aircraft with swept or delta wings at the high angles of incidence required for transonic maneuverability. This is particularly true for aircraft such as the Canadian Forces CF-18, which has a large wing strake at the junction with the fuselage. Other applications of Euler codes include the analysis of missiles and stores at high angles of attack. The highly non-linear flow that exists on these configurations in subsonic, transonic and supersonic regimes can only be predicted by the Euler and Navier-Stokes methods.

1.2 OBJECTIVES OF THE EULER CODE DEVELOPMENT

The subject of this thesis is the development of a transonic multi-block Euler flow solver for complete aircraft configurations and its application to the Canadair CL-601 Challenger airplane. This work is part of an even larger research effort by the staff of Canadair. The research was conducted in the past three years, under the supervision of the author who was the principal investigator of the project. This total project, funded partly by Canadair and partly by the department of National Defence of Canada had four main objectives :

1. The coupling of a two-dimensional Euler code for isolated airfoils with laminar and turbulent compressible boundary layer programs to predict airfoil characteristics in viscous transonic flow. A subsequent link with an optimization procedure allowed the design of airfoils producing prescribed pressure distributions.
2. The expansion of a two-dimensional unstructured grid Euler code into a Navier-Stokes code by the addition of physical viscous fluxes and the modelling of turbulence (Baldwin-Lomax algebraic model).

3. The development of a three-dimensional Euler code for arbitrary configurations using a multi-block procedure and its application to the Canadair CL-601 Challenger and the CF-18 aircraft configurations.
4. The development of supporting two-dimensional and three-dimensional grid generation codes.

The author, as principal investigator in charge of the complete project was responsible for selecting the formulations for each part of the project and for supervising the subsequent development. In addition, He was personally responsible for the writing of the three dimensional multi-block Euler code (item 3 above) which is the subject of this thesis. Our multi-block Euler code was developed to provide the most accurate inviscid flow solution around business aircraft like the Challenger and military aircraft like the CF-18. A link of the Euler code with boundary layer calculations on wing surfaces was also planned to provide a better estimate aerodynamics of the aircraft in cruise. This Euler code will be used by Canadair:

1. To confirm the predictions of a wing/body Full Potential code for newly developed aircraft configurations with supercritical wings (Global Express).
2. To estimate the effect of the aft fuselage mounted nacelle on the pressure distributions of the wing at various flight conditions. Because of its accurate geometry model and its sophisticated formulation, the Euler code gives better predictions of these effects than the Transonic Small Disturbance code KTRAN presently used.
3. To improve the understanding of the nacelle aerodynamics in the presence of the downwash of the wing, analyze the flow in the inlet stream tube at various flight conditions and estimate engine installed performance. By predicting isolated and installed nacelle

pressure distributions, the code will serve to determine powerplant installation effects.

4. To optimize aft fuselage shapes, nacelle and pylon positions and shapes1‘ in order to minimize interference drag and obtain clean flow for this highly three-dimensional geometry.
5. To obtain accurate predictions of the aerodynamic loads acting on the complete aircraft configuration in the transonic regime. Up to now, loads on the complete aircraft could only be obtained using subsonic panel methods. Additional aerodynamic extrapolation and wind tunnel testing were still required to account for transonic effects.

The development of a reliable and accurate Euler code is therefore a major step forward in the application of CFD to the design and analysis of aircraft configurations at Canadair, which should result in more refined products and reduced amounts of wind tunnel and flight testing. The first application of the new code is the Global Express long range, high-speed executive airplane, planned for first flight in 1997. It must be emphasized that although several Euler codes have been developed by the CFD community, only very few of them can be considered as capable, accurate and reliable enough to be useful in an aircraft design cycle and those codes are typically proprietary to the organizations that have developed them. Many elegant codes, with impressive results in simple generic cases, have not withstood the test of complete usefulness in an aircraft manufacturer environment (ease of use, confidence in the results,etc.). These were primary goals for the development of the code described here. A longer term goal is to use the Euler code as a support for developing a Navier-Stokes program for complete aircraft configurations. A proposal for such development was prepared and the development is scheduled to begin in mid 1993 [12].

1.3 OUTLINE

The text in chapter 2 is divided in four parts : a study of the Euler equations and of the nature of the solutions of these equations; a short summary of the best known schemes developed to solve these equations (a background for evaluation of our proposed formulation); a review of applications of Euler solvers in the industry (illustrating the objectives of this work and indicating the successes and limitations of some formulations for practical problems); a presentation of choices we made in selecting the formulation proposed here. Chapter 3 is a description of the surface modelling and grid generation techniques developed in support of the three-dimensional multi-block Euler code. Chapter 4 is a description of the flow equations used in the present work, the discretization formulation and the time-integration technique. The elements of theory specific to our method are discussed in chapter 5 and 6. They are related to the application of boundary conditions for complete aircraft configuration and to the formulation of the multi-block technique. Chapter 7 is a presentation of the results obtained during the development of the code and the aerodynamic characteristics calculated for various configurations of the CL-601 Challenger and the CF-18 complete aircraft configurations.

CHAPTER 2 : Solving Euler Equations : A Review

2.1 FORMULATION OF THE EULER EQUATIONS

The best model available today for describing mathematically the fluid motion phenomena encountered in most aeronautical problems is the set of unsteady compressible Navier-Stokes equations representing the conservation of mass, momentum and energy. The Navier-Stokes equations represent the flow of a gas in thermodynamic equilibrium. If ρ, u, v, w, E and p represent the density, Cartesian velocity components, total energy and pressure, respectively, the Navier-Stokes equations can be written in three-dimensional form as :

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = \frac{\partial \vec{R}}{\partial x} + \frac{\partial \vec{S}}{\partial y} + \frac{\partial \vec{T}}{\partial z} \tag{2.1.1}$$

where $\vec{W}$ is the vector of dependent variables and $\vec{F}$, $\vec{G}$ and $\vec{H}$ the convective flux vectors.

$$\vec{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} , \quad \vec{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho uH \end{pmatrix} , \quad \vec{G} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho vH \end{pmatrix} , \quad \vec{H} = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho wH \end{pmatrix}$$

H is the total enthalpy which is given by :

$$H = E + \frac{p}{\rho} \tag{2.1.2}$$

The pressure is obtained from the equation of state :

$$p = (\gamma - 1)\rho[E - \tfrac{1}{2}(u^2 + v^2 + w^2)] \tag{2.1.3}$$

The flux vectors representing the viscous terms are :

$$\vec{R} = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ \tau_{xz} \\ u\tau_{xx} + v\tau_{xy} + w\tau_{xz} \\ -\ q_x \end{pmatrix} , \quad \vec{S} = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ \tau_{yz} \\ u\tau_{xy} + v\tau_{yy} + w\tau_{yz} \\ -\ q_y \end{pmatrix} , \quad \vec{T} = \begin{pmatrix} 0 \\ \tau_{xz} \\ \tau_{yz} \\ \tau_{zz} \\ u\tau_{xz} + v\tau_{yz} + w\tau_{zz} \\ -\ q_z \end{pmatrix}$$

The viscous stresses are given by :

$$\tau_{ij} \ = \ \mu \ [\ (\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}) - \frac{2}{3}\delta_{ij}\vec{\nabla}\vec{v} \ ] \tag{2.1.4}$$

The heat conduction terms are :

$$q_x = -\kappa\frac{\partial T}{\partial x} \ , \quad q_y = -\kappa\frac{\partial T}{\partial y} \ , \quad q_z = -\kappa\frac{\partial T}{\partial z} \tag{2.1.5}$$

However, the complexity of these equations and the large computer resources needed to solve them for useful configurations have sponsored the development of simpler models. One of them is the use of an inviscid model for the outer flow coupled with boundary layer calculations for representing the viscous effects near the surface of the airplane. The Euler equations are obtained from the Navier-Stokes equations by neglecting all shear stresses and heat conduction terms. They represent the most general description of inviscid, non-heat conducting flows. They can be written in conservation form and in an absolute frame of reference as :

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} \ = \ 0 \tag{2.1.6}$$

They form a system of first-order partial differential equations which is hyperbolic in time. An analysis of the entropy equation for inviscid continuous flow, in the absence of heat sources indicates that entropy remains constant along a flow path [13]. The Euler equations represent therefore isentropic flow along a flow path in the absence of discontinuity. However, the entropy can vary from one flow path to another. This variation of entropy in the direction normal to the local flow velocity is related to the variation of vorticity, as seen from Crocco's form of the momentum equation in stationary inviscid flow without external forces :

$$-\vec{v} \times \vec{\zeta} = T\vec{\nabla}s - \vec{\nabla}H \tag{2.1.7}$$

In this eqation $\vec{\zeta}$ is the vorticity vector, s the entropy, T the temperature and H the total enthalpy. In addition, the Euler equations allow discontinuities in the solution across which the entropy can vary. The properties of discontinuous solutions are given by the local form of the conservation laws over a discontinuity called the *Rankine-Hugoniot* relations [13]. The various forms of discontinuities physically possible are :

- *Contact discontinuities :* with no mass flow through the discontinuity ($\Delta v_n = 0$), continuity of pressure ($\Delta p = 0$) and possible discontinuities in specific mass ($\Delta\rho \neq 0$) and tangential velocity ($\Delta v_t \neq 0$).
- *Vortex sheets or slip lines :* also defined by a condition of zero normal mass flow ($\Delta v_n = 0$), continuity of pressure ($\Delta p = 0$) but allowable jumps in tangential velocity ($\Delta v_t \neq 0$) and density ($\Delta\rho \neq 0$).
- *Shock surfaces :* these are solutions of the Rankine-Hugoniot relations with non-zero normal mass flow ($\Delta v_n \neq 0$), discontinuity of pressure ($\Delta p \neq 0$) and density ($\Delta\rho \neq 0$). but continuity of tangential velocity ($\Delta v_t = 0$)

These types of discontinuities are important in the transonic flow solutions developed in the present work. For physical reasons the discontinuous variation of entropy occurring through a shock wave must be positive (compression shock). Expansion shocks with negative jumps of entropy are valid solutions of the inviscid equations. It is therefore necessary to add an *entropy condition* to the equations in order to suppress these non-physical solutions. A good review of the properties of these equations can be found in a book by Hoffman [14].

2.2 NUMERICAL SOLUTIONS OF THE EULER EQUATIONS

The solution of the Euler equations is a non linear convection dominated problem. During the seventies, there were widespread efforts to develop methods for predicting transonic flows with shock waves using the Transonic Small Disturbance and Full Potential equations. In parallel, beginning in 1970, several algorithms were devised to solve the Euler and Navier-Stokes equations. The pioneering work was done by Magnus and Yoshihara [15] but the real impetus came with the introduction in 1970 by MacCormack of his famous explicit finite difference scheme [16]. Other schemes followed, trying to improve the efficiency of the computation. This section looks at some of the most notable methods that were devised to solve the Euler equations. The successes and failures of some methods must be linked to the performance levels required for effective use in the aeronautical industry [17] :

- The ability to predict the flow past airplanes at different flight regimes.
- The possibility of interactive calculations for immediate improvement of designs.
- The possibility of integration of the predictive capability into an automatic design method using computer optimization and artificial intelligence.

To meet these performance levels, CFD codes are developed with regard to :

- The ability to simulate main features of the flow (shock waves, vortices).
- The ability to predict viscous effects.
- The ability to handle complex geometry.
- The efficiency in computational and human effort.

The steady Euler equations can be solved by introducing a space discretization procedure that reduces the problem to the solution of a large number of coupled non linear equations. These equations can be solved by iterative methods : for example, the Newton iteration method (Giles, Drela and Thompkins [18]) and the least squares methods (Bristeau, Glowinski, Periaux, Perrier, Pironneau, Poirier [19]). The other way of obtaining steady state solution is through the use of time-dependent solutions which have the following advantages :

- Simplicity.
- The same code can be used for steady and unsteady calculations.
- They are a natural framework for designing non-oscillatory shock capturing schemes reflecting the physics of wave propagation.
- They can be used on vector, pipelined and parallel computers : explicit time stepping or iterative procedure at each time step for implicit schemes.

The present review will focus on time-dependent solutions, since the method selected for the present work belongs to this family. During this review, it should be remembered that good numerical schemes should be able to satisfy the following requirements :

- The numerical approximations should satisfy the conservation laws.
- Shock waves and contact discontinuities should be automatically captured.
- The final steady state should be independent of the time step.
- Physical invariant quantities in the flow field (entropy upstream of shocks and enthalpy in steady flow) should remain invariant in the numerical solution.
- Uniform flow should be an exact solution.

The reasons that led to the choice of the formulation selected for the present application will be explained at the end of this chapter. To put those arguments in proper light, it is useful to review various methods which have been developed to address the solution of the Euler equations and to outline their strengths and weaknesses. This review will focus only on those methods which have had the greatest impact in the computation of fluid flows in aeronautics. Excellent reviews in more thorough terms can be found in the classical works of Hirsch [13] and Hoffmann [14]. During the following discussion, various terms related to the performance of the various schemes will be used : consistency, stability, convergence, accuracy and overall computational efficiency. Prior to the discussion, it is useful to restate what these terms mean in the present context.

Consistency : A system of algebraic equations generated by the discretization process is said to be consistent with the original partial differential equation if, in the limit of grid spacing tending to zero, the system of algebraic equations is equivalent to the partial differential equation at each grid point. The consistency condition establishes therefore a relation between a differential equation and its discretized formulation.

Stability : is the tendency of any spontaneous perturbation such as a round-off error in the solution of the system of algebraic equations to decay. The stability condition establishes therefore a relation between the computed solution and the exact solution of the discretized equation. The two most common methods of stability analysis are the matrix method and the von Neumann method. Both predict whether there will be a growth of the error between the true solution of the numerical algorithm and the actually computed solution. the von Neumann analysis is the most commonly used method of determining stability criteria and is the easiest to apply. However it can only be used to establish necessary and sufficient conditions for the stability of linear initial value problems with constant coefficients. For problems with variable coefficients, non linearities and complex boundary conditions, the method can only be applied locally with non linearities temporarily frozen. Although it is strictly for interior points, it has been used to provide some information on the influence of boundary conditions on numerical stability.

Convergence : A solution of algebraic equations which approximate a given partial differential equation is said to be convergent if the approximate solution approaches the exact solution of the partial differential equation for each value of the independent variables as the grid spacing tends to zero. The difference between the exact solution of the partial differential equation and the exact solution of the algebraic equations is called the solution error. The convergence condition establishes therefore a relation between the computed solution and the exact solution of the differential equation. According to Lax theorem, for a well posed initial value problem and a consistent discretization scheme, stability is a necessary and sufficient condition for convergence.

Solution accuracy : in practice, approximate solutions are obtained in finite grids. the accuracy of a particular algorithm can be determined by applying it to a simple problem which has exact solutions. The accuracy of Euler solvers has often been tested by their application to the Burgers equation for which exact solutions can be found for various initial and boundary conditions.

Computational efficiency : is the accuracy achieved per unit of execution time.

In the following discussion, we will distinguish between explicit and implicit schemes. We will also distinguish between the space-centered schemes and the upwind schemes.

Explicit methods are simple, easily vectorizable and allow large flexibility in the treatment of boundary conditions. The biggest problem of the explicit schemes is their conditional stability which may place severe limitations on the size of the time step, especially with fine meshes. This is a consequence of the famous Courant, Friedrichs and Lewy condition (CFL condition) : If a_{max} is the maximum speed of propagation of a one-dimensional problem, the time step Δt is limited by :

$$\Delta t < \sigma \frac{\Delta x_{min}}{a_{max}} \tag{2.2.1}$$

where Δx_{min} is the minimum grid spacing and σ the maximum Courant number or maximum CFL number :

$$\sigma = a \frac{\Delta t}{\Delta x} \tag{2.2.2}$$

As a consequence, explicit schemes may require large number of time steps (several thousands) in order to reach a satisfactory level of convergence for steady state problems. This situation has somewhat been improved by the use of several devices which will be discussed later, including the use of multiple grids.

An alternative for problems when time accuracy is not required (steady state problems) or when the time-scale of the unsteady phenomena is much larger than the time step allowed by the CFL condition is to use *implicit schemes.* These schemes are unconditionally stable and are limited only by accuracy requirements. In practice, however, too large time steps may create non linear instability. In addition, the choice of a boundary condition implementation may introduce a stability limit. Using implicit schemes, the solution can be obtained therefore with a smaller number of iterations, at the expense of a higher operations count per iteration.

2.2.1 Lax-Wendroff Types of Space-centered Schemes

These schemes for the Euler equations were the first to be derived. The most important property of the Lax-Wendroff schemes is the combination of time and space-centered discretization. All centered explicit or implicit schemes of second order accuracy with combined space-time discretization are generally considered as belonging to the Lax-Wendroff family.

2.2.1.1 Explicit Schemes

a - First Order Schemes.

The Lax or Lax-Friedrichs schemes [20] were the first modern development in the field of numerical discretizations of the Euler equations. Consider the linearized convection equation with the dependent variable u :

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{2.2.3}$$

The basic idea of the Lax scheme was to replace the unstable explicit central scheme :

$$u_i^{n+1} - u_i^n = -\frac{\sigma}{2}(u_{i+1}^n - u_{i-1}^n) \tag{2.2.4}$$

by the stabler procedure :

$$u_i^{n+1} = \frac{1}{2}(u_{i+1}^n + u_{i-1}^n) - \frac{\sigma}{2}(u_{i+1}^n - u_{i-1}^n) \tag{2.2.5}$$

A similar procedure can be described for the conservative form of the equations; when analyzed, the stabilization procedure is seen to be the addition of a dissipative term proportionnal to the second derivative of u. Schemes for non-linear equations and two-dimensional equations were derived but they are rarely used because of their low accuracy. First order schemes have truncation errors proportional to second derivatives acting as added numerical viscosity which damp high-frequency components and smooth out strong gradients.

There have been attempts to improve the accuracy of the Lax scheme by introducing corrected viscosity terms, but these will not be discussed here. The most useful Lax-Wendroff schemes are the second order schemes presented in the next section.

b - Second Order Schemes

Lax-Wendroff scheme.

The Lax-Wendroff scheme is the only second order, space-centered discretization on a three-point support for the linear one-dimensional convection equation. The variants of this scheme vary in the treatment on non-linearities and in their multi-dimensional aspects. The most useful version is the two-step version described in 1964 [21]. Consider the conservative version of the one-dimensional convection equation :

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{2.2.6}$$

The two-steps Lax-Wendroff scheme can be written as :

Step 1 :

$$u_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \frac{1}{2}(u_i^n + u_{i+1}^n) - \frac{1}{2}\tau(f_{i+1}^n - f_{i-1}^n) \tag{2.2.7}$$

Step 2 :

$$u_i^{n+1} = u_i^n - \tau(f_{i+\frac{1}{2}}^{n+\frac{1}{2}} - f_{i-\frac{1}{2}}^{n+\frac{1}{2}}) \tag{2.2.8}$$

where $\tau = \Delta t / \Delta x$.

MacCormack scheme.

The most popular version of the Lax-Wendroff scheme is the two-step predictor-corrector MacCormack explicit method [16] which is a second order time-accurate scheme. The first step is a first order forward discretization in space. The second step is a first order backward scheme. The first step in isolation is unstable for positive eigenvalues of the Jacobian matrix A of the equation (for supersonic velocities) while the second step is unstable for negative characteristic speeds of propagation. However, the combined scheme is stable and, due to the cancellation of the truncation errors of each step, it is second order accurate. For the equation presented above, the scheme can be written as follows :

Predictor step :

$$u_i^p = u_i^n - \tau(f_{i+1}^n - f_i^n) \tag{2.2.9}$$

Corrector step :

$$u_i^{n+1} = \frac{1}{2}(u_i^n + u_i^p) - \frac{1}{2}\tau(f_i^p - f_{i-1}^p) \tag{2.2.10}$$

Note that all second order three-points central schemes of the Lax-Wendroff family produce oscillations around sharp discontinuities. They require generally the addition of artificial dissipation terms. The issue of artificial viscosity will be discussed separately later in the text. The Lax-Wendroff family of schemes is still widely used to discretize Euler and Navier-Stokes equations, particularly in the form of the MacCormack predictor-corrector formulation. The author used successfully this formulation in 1980 to calculate two-dimensional transonic blade-to-blade flow in turbine cascades [22],[23]. Another example of the use of the MacCormack finite difference algorithm is the code of Peery and Forrester of Boeing [24] which was written to calculate Euler and Navier-Stokes solutions in unsteady two-dimensional flow. The results that they obtained for turbine engine exhausts were a useful reference in the construction of the engine model developed in the present work. Charles

Koek of MATRA [25], in France calculated unsteady Euler equations using a finite volume one-step scheme of Ron Ho Ni with the multiple grid procedure of Ni [26]. The basic scheme used is a one-step Lax-Wendroff scheme said to be second-order accurate in space and time. Chima of NASA Lewis and Johnson of the Colorado State University use time-marching procedures to solve Euler and Navier-Stokes procedures [27] for two-dimensional flow. They use the MacCormack version of the two-steps Lax-Wendroff method. A coarse grid acceleration following the fine grid solution procedure of Ni [26] is also used. Arthur Rizzi of FFA made calculations on wing-body combinations using an O-O transfinite interpolation mesh [28]. He used a similar explicit time-marching finite volume method with an internal mechanism for temporal damping and use of local time step. All spatial derivatives are centrally differenced. The scheme is second-order accurate, has a simple structure and is easily vectorizable.

2.2.1.2 Implicit Schemes

Implicit schemes of the Lax-Wendroff family have been developed by Lerat [29]. These schemes are second-order accurate, unconditionally stable and have interesting dissipative properties. However, they have not been widely used for aeronautical applications, as far as we were able to determine. They will not, therefore, be discussed further. The Lax-Wendroff family of schemes is derived from a combined space and time discretization. As a consequence, the steady state numerical solution depends on the time-step used in the computations. However small this error can be, it constitutes a drawback since it introduces a numerical parameter in the predicted steady state flow.

2.2.2 Central Schemes with Independent Time Integration

The idea behind these schemes is to make the final steady state solution independent of the time step. To do this, one convenient way is to separate the time marching procedure from the space marching procedure entirely. This is done by applying first a semi-discretization. The problems of spatial discretization error, artificial dissipation and shock modelling can then be studied independently of the time marching stability and convergence acceleration. These schemes are based on central space discretizations and therefore do not distinguish between upstream and downstream influences. The physical propagation of perturbation along characteristics, which is typical of hyperbolic equations, is not considered in the definition of this numerical model. The central differencing generates odd-even decoupling (high frequency error) which, for non linear problems can affect the smoothly varying solution (interaction of modes creating low frequency errors). In addition, these schemes, like the Lax-Wendroff schemes, generate oscillations in the vicinity of discontinuities. They require therefore the addition of artificial dissipation terms to damp these oscillations. The two most important schemes in this class are the implicit Beam-Warming schemes and the Jameson explicit multistage Runge-Kutta scheme.

2.2.2.1 Beam-Warming Schemes

Beam and Warming introduced their scheme in two papers published in 1976 [30] and 1978 [31]. A good account of their approach can be found in their notes for a lecture series given at the Von Karman Institute in 1982 [32]. In one dimension, the time discretization and linearization are obtained by applying linear multi-step integration schemes to the space discretized equations considered as a system of ordinary differential equations in time.

Beam and Warming use implicit operators reducing to tridiagonal or block tridiagonal inversions which are performed efficiently with a Thomas algorithm. Considering again the one dimensional convection equation :

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{2.2.11}$$

The Beam-Warming approach leads to the semi-discretized scheme :

$$(1+\xi)\Delta u^n - \xi\Delta u^{n-1} = -\Delta t[\theta f_x^{n+1} + (1-\theta)f_x^n] \tag{2.2.12}$$

with $\Delta u^n = u^{n+1} - u^n$.

An important aspect of the scheme is the linearization of the flux derivative term f_x^{n+1} as follows :

$$f^{n+1} = f^n + \Delta t f_t^n = f^n + A^n(U^{n+1} - U^n) \tag{2.2.13}$$

which leads to the form :

$$[(1+\xi) + \Delta t\theta A_x^n]\Delta u^n = -\Delta t f_x^n + \xi\Delta u^{n-1} \tag{2.2.14}$$

This scheme is a three-level scheme containing u^{n-1}, u^n and u^{n+1} (for $\xi \neq 0$) and is second order accurate for $\theta = \xi + \frac{1}{2}$. Depending on the values of θ and ξ, one can obtain various schemes : Euler explicit, explicit leapfrog, implicit trapezoidal, Euler implicit and a three-point implicit scheme. When second order central differences are applied to the implicit operators, the implicit matrix transforms into block tridiagonal systems which can be economically solved. Since the time integration is independent of the space discretization, the scheme can be readily generalized to two or three dimensions. However, this leads to costly matrices to invert (block pentadiagonal matrix in a two-dimensional case). Therefore, an

Alternating Direction Implicit (ADI) factorization is used. The principle of ADI is to separate the operators into one dimensional components and split the scheme into a number of steps equivalent to the number of dimensions of the problem. Each step involves only the implicit operations originating from a single coordinate. The idea behind the ADI method is the factorization of the operator in a product of one-dimensional operators. This factorization introduces an error which is of the same size as the truncation error and therefore does not affect the overall accuracy of the scheme. One of the best known application of the Beam-Warming scheme is the ARC2D program developed at NASA Ames [33]. This is a two-dimensional implicit Euler and Navier-Stokes code. The equations are written in general curvilinear coordinates. The physical domain is mapped on a computational rectangular domain. The method is in fact a variation of the approximate factorization algorithm in delta form (Beam-Warming) due to Pulliam and Chaussee [34] which diagonalizes the implicit operators. The diagonal form has advantages over the standard block tridiagonal algorithm in terms of efficiency and convergence characteristics. The method can be first order or second-order accurate in time. All the space derivatives are second-order central differences. J. Vadyak of Lockheed-Georgia uses a fully implicit ADI algorithm (Fully implicit approximate factorization scheme) to study nacelle inlet flow fields [35]. The non-iterative Beam-Warming algorithm used is cast in correction form. Second-order accurate spatial discretization and first or second order temporal differencing are used in the code. The ADI scheme is used to split the three-dimensional difference operator in three one-dimensional operators. The code performs the solution of a block tridiagonal system of linear simultaneous equations in each direction at each time step. To obtain the steady state solution, the equations are integrated in time to convergence of the surface solution properties. Explicit boundary conditions are used.

2.2.2.2 LU Decomposition

An alternative approximate factorization method is the LU decomposition method : the idea is to replace the operator in the Euler equation by the product of lower and upper block triangular factors L and U. Two factors are used independently of the dimensions and the inversion of each can be accomplished by the inversion of its diagonal block [36].

2.2.2.3 Jameson's Multistage Scheme

This finite volume scheme presented by Jameson, Schmidt and Turkel [37] in 1981 applies a Runge-Kutta multistage time integration scheme to the central discretization of the flux balance. The method is an efficient combination of dissipation terms, convergence acceleration devices and multi-grid techniques and has produced very accurate and computationally efficient codes. The method selected in the present work is based on Jameson's formulation which will therefore be fully developed in the next chapter. The main features of the method are :

- It uses the Euler time dependent equations in Cartesian coordinate system cast in strong conservation form.
- A finite volume method is used to transform the system of hyperbolic equations into a set of ordinary differential equations to be integrated in time.
- Discontinuities such as shock waves and contact discontinuities are automatically captured, spreading over a small number of cells.
- All spatial derivatives are centrally differenced. The scheme is second order accurate if the mesh function is smooth enough.
- To prevent the tendency for odd-even decoupling and the appearance of wiggles in re-

gions of severe pressure gradients, artificial dissipative terms are added. These are a blend of second and fourth differences with coefficients depending on local flow caracterisics. The overall level of dissipation is controlled by a user-defined parameter.

- A Runge-Kutta multistage algorithm is used to integrate the equations in time. Additional stages are added to improve accuracy and extend the stability region of the scheme. Four-stages and Five-stages schemes have been used. These schemes are fourth order accurate in time for a linear equation in one dimesion. They are stable for a Courant number : $|\sigma| \leq 2\sqrt{2}$.
- Various devices are used to accelerate convergence to the steady state :
    - Local time step, to advance the solution everywhere at the maximum rate compatible with a fixed Courant number and the local cell size.
    - Enthalpy damping. An additional term proportional to the difference $(H - H_\infty$ is added to the residual which has a damping effect on the problem. Jameson's discretization guarantees $H = H_\infty$ at convergence, making this term tend to zero.
    - Residual smoothing. After each explicit step, an additional implicit step is added to average the variation of the residuals. This has the result of increasing the maximum allowable Courant number and of smoothing the high-frequency variations of the residuals. The latter property makes residual smoothing an essential feature of the multi-grid convergence acceleration scheme.
    - Multigrid acceleration. The convergence to steady state is accelerated by transferring to coarser meshes since larger time steps can be used there. The corrections are interpolated back to the fine grid where a smoothing operator removes the high-frequency errors.

The method has been developed into very efficient codes for two-dimensional and three-dimensional applications. Jameson's best known codes are the FLO-52 and FLO-82 codes for isolated airfoils [37] and the FLO-57 and FLO-67 codes for isolated wings which will be discussed later. Several aircraft companies have used these programs as the basis for developing more complex codes which have remained proprietary to each organization :

- At Lockeed, Raj [38] and Olling and Mani [39] have developed efficient multi-block Euler and Navier-Stokes codes based on this formulation (TEAM and TRANSAM). The codes can model complete aircraft configurations and have been used mainly to study military aircraft configurations at high angles of incidence.
- At Douglas, Shmilovich and Chang have had applications to simple aircraft-nacelle configurations [40].
- Northrop was one of the first companies to develop a multi-block application, with the particular application for the F-20 tigersharck (Busch, Jager and Bergman [41]).
- British Aerospace has also developed a version of the program for application to multiple store carriage. (Shepherd and Tod [42]).
- Israel Aircraft industry and Analytical Methods Inc have developed a version of the program formulated for Cartesian grids (MGAERO). Although not as accurate as the body-fitted grid version, it offers the advantage of automated grid generation for fairly complex configurations (Epstein, Luntz and Nachshon [43].

Jameson has continually cooperated with several research establishments and aircraft companies to further the application of this scheme :

- At Boeing, Chen, Yu, Rubbert and Jameson have developed variants for general nacelle configurations [44].
- More recently, preliminary calculations were done using tetrahedral cells in an unstructured mesh for the Douglas DC-10 airplane [45].

2.2.3 Upwind Schemes

The upwind schemes were constructed with the idea of introducing the physical properties of the flow equations in the discretization formulation. Two main variants of the method can be considered :

- Flux Vector Splitting (FVS) methods : here, only the information on the sign of the Jacobian eigen values is introduced in the discretization; the flux terms are split and discretized directionally according to the sign of the associated propagation speeds. (Steger and Warming [46], Van Leer [47]).
- Godunov-type methods : here, properties derived from the exact local solution of the Euler equations are introduced in the discretization. These schemes follow the pioneering work of Godunov [48]. In his method, the conservative variables are considered piecewise constant over the mesh cells at each time step; the time evolution is then determined by the exact solution of the Riemann (shock tube) problem at the interface between adjacent cells. In higher order extensions and variants of these methods, the local Riemann problem is only approximately solved through approximate Riemann solvers (Roe [49], Osher [50]). These methods are also known as Flux Difference Splitting (FDS) methods.

2.2.3.1 Basic Principles

The basic principle of the upwind schemes can be described by considering the following linearized convection equation :

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0 \tag{2.2.15}$$

For positive propagation speeds ($a > 0$), a stable first order explicit discretization of this equation with the Courant number σ is :

$$u_i^{n+1} = u_i^n - \sigma(u_i^n - u_{i-1}^n) \tag{2.2.16}$$

A von Neumann analysis indicates that this scheme is stable for values of the Courant number $0 \leq \sigma \leq 1$ but it is unstable for negative characteristic speeds. The analysis of the truncation error shows that the scheme is only first order accurate in space and time and that the equivalent equation has a dissipative term with a numerical viscosity coefficient equal to : $a\Delta x(1-\sigma)/2$. The remarkable property of the scheme is that this viscosity term vanishes for $a = 0$, i.e. when the characteristic eigenvalues pass through zero. This occurs in stagnation regions and at sonic transitions. This property of vanishing numerical viscosity is responsible for the sharp capture of discontinuities which is afforded by upwind schemes. In comparison, the numerical viscosity of a first order central scheme never vanishes and is much higher. Note that the property of vanishing dissipation at sonic transitions can also lead to non physical expansion shocks. The addition of an entropy condition to prevent this may then be necessary. The simple example above can also serve to illustrate the other nature of upwind schemes. Indeed, for negative propagation schemes ($a < 0$), a stable scheme is :

$$u_i^{n+1} = u_i^n - \sigma(u_{i+1}^n - u_i^n) \tag{2.2.17}$$

the scheme is stable for negative values of the Courant number $-1 \leq \sigma \leq 0$. This shows that an upwind scheme cannot be simultaneously stable for both positive and negative values. The so-called *upwind schemes*, illustrated above, apply a discretization that depends on the propagation direction of the wave or on the sign of the convection velocity a. The general form of a first order accurate upwind scheme for the linearized scalar form of the Euler equations can be written as :

$$u_i^{n+1} - u_i^n = -\tau[a^+(u_i^n - u_{i-1}^n) + a^-(u_{i+1}^n - u_i^n)] \tag{2.2.18}$$

with :

$$\tau = \Delta t/\Delta x \tag{2.2.19}$$

$$a^+ = max(a,0) = (a + |a|)/2 \tag{2.2.20}$$

$$a^- = min(a,0) = (a - |a|)/2 \tag{2.2.21}$$

and the stability limit is :

$$|\sigma| = \tau|a| \leq 1 \tag{2.2.22}$$

The origin of this upwind scheme can be traced back to Courant, Isaacson and Reeves [51]. The upwind schemes introduce a directionally biased space discretization that takes properly into account the propagation properties of a hyperbolic equation. In contrast, the central difference methods use central discretization without regard to the direction of propagation. The propagation properties of the equations are taken into account in these methods by the introduction of artificial viscosity terms.

2.2.3.2 Flux Vector Splitting Schemes

The idea of one-dimensional flux vector splitting was introduced by Steger and Warming [46]. This is an upwind method that defines the splitting of the fluxes according to the signs of the eigenvalues. The flux splitting methods developed in one dimension can be extended to multi-dimensional flows by applying one-dimensional splitting to each flux component separately, in accordance with the sign of the associated eigenvalues. Two such methods are those of Steger-Warming and Van Leer.

Steger and Warming flux splitting.

The flux splitting approach is equivalent to writing the flux $f(u)$ as the sum of a forward flux and a backward flux :

$$f(u) = f^+(u) + f^-(u) \tag{2.2.23}$$

where :

$$f^+ = A^+.u \qquad and \qquad f^- = A^-.u$$

A is the Jacobian of the flux vectors with respect to the conservative variables. Advantage is taken here of the fact that the flux vector $f(u)$ is a homogeneous function of degree one of u, and therefore :

$$f = A.u \tag{2.2.24}$$

The Jacobian of the split fluxes f_u^+ and f_u^- used by Steger and Warming are not continuously differentiable, since they have a discontinuous slope at sonic velocities. This may cause discontinuities in the slope of the computed solution at the sonic transition. The Steger-Warming flux formula does not provide any vanishing dissipation in any steady wave, which makes the capture of shock waves not as good as with Van Leer's formulation discussed next.

Van Leer flux splitting.

Van Leer introduced a different approach [47], with a certain number of conditions added on to the split fluxes. He obtains this way a differentiable flux-split formula to represent the numerical flux $f(u)$ of the convective terms of the Euler equations. In the relation :

$$f(u) = f^+(u) + f^-(u) \tag{2.2.25}$$

$f^+(u)$ and $f^-(u)$ and the associated Jacobians f_u^+ and f_u^- must be continuous functions of Mach number and are expressed as polynomials of the lowest possible order. It is required in addition that :

$$df^+(u)/du \geq 0 \tag{2.2.26}$$

$$df^-(u)/du \leq 0 \tag{2.2.27}$$

The numerical flux based on the left and right state can be written as :

$$f(u_L, u_R) = f^+(u_L) + f^-(u_R) \tag{2.2.28}$$

Van Leer's formula makes dissipation vanish in steady shocks, leading to capture of these shocks within one to two interior states. However, the scheme has numerical diffusion at contact discontinuities, with or without slip,which are less well captured. The CFL3D code used at NASA Langley by Anderson and Batina [52] is an implicit finite volume, upwind-differencing code. In this code, the spatial derivatives are split into forward and backward contributions using flux vector splitting so that type-dependent derivatives can be used. The flux-vector splitting is that of Van Leer [47]. The code incorporates also the flux-differencing approach of Roe [49] which will be described later. The program can be steady or unsteady and can be used either as an Euler or a Navier-Stokes code.

2.2.3.3 Godunov-type Schemes

The various numerical flux formulas used to represent the convective terms of the Euler and Navier-Stokes equations are discussed in a paper by Van Leer, Thomas, Roe and Newsome

[53]. The accuracy of the various schemes is discussed with relation to representing steady non linear waves (shocks) and steady linear waves (entropy/shear waves). It has been shown that the correct value of the flux vector can be found by solving the Riemann problem governed by a one-dimensional inviscid flow equation at the interface :

$$u_t + f(u)_x = 0 \tag{2.2.29}$$

Solving the exact Riemann problem at every interface leads to the Godunov flux formula. Numerically, it is wasteful to do this at every interface. This is the reason why several other methods use approximate Riemann solvers instead. The simplest flux formula is obtained by averaging the flux values at the left and right side of the interface :

$$f(u_L, u_R) = \frac{1}{2}[f(u_L) + f(u_R)] \tag{2.2.30}$$

This leads to central difference schemes which are unstable and require the addition of artificial dissipation to the flux. The scheme can be written as :

$$f(u_L, u_R) = \frac{1}{2}[f(u_L) + f(u_R)] - \frac{1}{2}Q(u_L, u_R).(u_R - u_L) \tag{2.2.31}$$

Different numerical fluxes vary in the selection of the matrix coefficient $Q(u_L, u_R)$ of the dissipation term.

Roe's approximate Riemann Solver (FDS, Flux Difference Splitting).

In the method proposed by Roe [49],[54], the numerical flux is based on the solution of Riemann's problem with the initial values u_L for $x < 0$ and u_R for $x > 0$ for the linearized equation :

$$u_t + A(u_L, u_R)u_x = 0 \tag{2.2.32}$$

where $A(u_L, u_R)$ is the mean value of the Jacobian $A(u)$ of $F(u)$ required to have the following properties :

1- $A(u,u) = A(u)$.

2- $A(u_L, u_R)$ has a complete set of eigenvalues and eigenvectors.

3- $A(u_L, u_R).(u_R - u_L) = f(u_R) - f(u_L)$.

The numerical flux based on this "approximate Riemann solver" can be written as :

$$f(u_L, u_R) = \frac{1}{2}[f(u_L) + f(u_R)] - \frac{1}{2}|A(u_L, u_R)|.(u_R - u_L) \qquad (2.2.33)$$

where $|A|$ is the matrix with the same eigenvectors as A, whose eigenvalues are the absolute values of the eigenvalues of A (which represent the characteristic speeds normal to the interface). Because of the third property of A, if an eigenvalue of $A(u_L, u_R)$ vanishes, the corresponding eigenvalue of the dissipation matrix vanishes as well. The Roe scheme leads therefore to crisp representations of steady shocks and contact discontinuities if aligned with an interface (within one interior state). In order to avoid the appearance of expansion shocks, which is possible with Roe's formula, it is necessary to add an additional flux contribution when an expansion through a sonic point is detected. This contribution is added to the upstream point and subtracted from the downstream point to preserve conservation. Note that computer codes based on Roe's scheme can be expensive because of a large increase in the number of arithmetic operations required in the numerical approximation. In addition, they do not satisfy the condition of constant total enthalpy at the steady state because the dissipative terms entering the mass and energy equations are constructed independently. Some applications of Roe's scheme at École Polytechnique de Montréal can be found in the work of Trépanier, Reggio, Zhang and Camarero [55], as well as in the paper on unsteady aerodynamics by M. Paraschivoiu [56].

Osher's approximate Riemann Solver.

Osher's scheme [50] is another approximate Riemann solver with properties similar to those of the Roe's scheme. The flux formula of the Osher scheme is :

$$f(u_L, u_R) = \frac{1}{2}[f(u_L) + f(u_R)] - \frac{1}{2}\int_{u_L}^{u_R} |A(u)|.du \qquad (2.2.34)$$

where the integral is calculated along a path piecewise parallel to the eigenvectors of $A(u)$. Because of the differentiability of this flux, steady shock waves are captured within one to two interior states, instead of within zero to one interior states, as for Roe's scheme. In Osher's scheme, the detection of the sonic point is essential and in consequence, expansion shocks do not remain solutions of the scheme.

Other approximate Riemann Solvers.

Another application of upwind methods can be found in the paper by Eberle of the german aircraft manufacturer MBB [57]. The method used by Eberle is a one-step time dependent finite volume scheme, based on characteristic Taylor expansion of conservative flow variables. A characteristic upwinding of the fluxes, inspired by the ability of the Godunov first order scheme to resolve sharply discontinuities, is used. No added artificial viscosity is required. Good results are shown for the ONERA M6 wing and for a fighter type wing/body combination [57]. Other approximate Riemann solvers have been described, such as the scheme suggested by Harten and Lax [58]. These schemes will not be discussed here.

Comparison of approximate Riemann Solvers with other schemes.

A comparison of the results of the various upwind schemes for the Euler and Navier-Stokes equations is made in reference [53]. This paper also compares these results with those of the MacCormack and Jameson schemes. The MacCormack scheme is one of the most popular finite difference schemes for Euler and Navier-Stokes solvers, as seen previously. The scheme leads to non-linear instabilities and requires extra dissipation terms with a scalar coefficient proportional to the second pressure difference (similar to the one used in Jameson's method). The Jameson/Schmidt/Turkel method uses a flux formula not written in terms of interface values. A user-specified artificial viscosity coefficient is added. This term does not enter the flux formula. A non oscillatory interpolation is used to calculate the interface values from the discrete solutions. It is found that for the Euler equation, the Jameson and MacCormack schemes with properly tuned artificial dissipation coefficients, yield acceptable answers with shock resolved in three cells. The quality of the results may be worse for contact discontinuities and slip surfaces where there is no intrinsic mechanism counteracting the numerical diffusion. On the other hand, Roe's scheme (FDS, Flux Difference Splitting) is said to be the best for Navier-Stokes solutions where boundary layers must be resolved. It is found to be more accurate than Van Leer scheme (FVS, Flux Vector Splitting). The MacCormack scheme performs better than the FVS schemes in the boundary layer but renders the shock less well, requiring twice as many cells as the Roe scheme. Van Leer's formula requires four times as many as Roe. It is concluded that the Jameson type scheme is best suited for Euler code and restricted to applications in which contact/slip surfaces lie outside the region of interest. For Navier-Stokes solutions, it is found to be at best, with trimmed dissipation coefficients, in competition with the FVS.

2.2.4 Second Order Upwind and High Resolution Schemes

The replacement of first order upwind space differences by second order accurate formulas leads to the generation of oscillations around discontinuities. It was also shown that linear second order upwind schemes always generate oscillations. It was therefore necessary to find new ways to develop oscillation-free second order schemes which can provide physically acceptable solutions and represent accurately shocks and contact discontinuities. The study of the conditions required to satisfy these criteria was initiated by Godunov [48] who introduced the concept of *monotonicity.* For non linear equations, Harten [59] introduced the concept of *bounded total variation* of the solution as a more general criterion for oscillation-free numerical schemes. Many schemes were developed to satisfy these requirements. It was shown however that such schemes could only be first order accurate (Godunov theorem). It was found finally that the only way to circumvent this limitation was to introduce non linear discretizations. This concept was introduced by Van Leer [60] and by Boris and Book [61] under the form of limiters which control the gradients of the computed solutions. A complete discussion of these schemes is outside the scope of this presentation. We will only briefly introduce the notion of Total Variation Diminishing (TVD) condition. The concept originates from an important property of the scalar conservation law :

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x} = 0 \tag{2.2.35}$$

The *Total Variation* of a physically admissible solution of this equation is defined as :

$$TV = \int_{-\infty}^{+\infty} |u_x| dx \tag{2.2.36}$$

This total variation has the property that it does not increase, provided that any discontinuity appearing in the solution satisfies an entropy condition. The Total Variation

Diminishing (TVD) schemes introduced by Harten provide a unifying framework for the study of shock capturing methods. TVD schemes are schemes with the property that the total variation of the discrete solution cannot increase. This total variation is defined as :

$$TV = \sum_{-\infty}^{+\infty} |u_i - u_{i-1}| \tag{2.2.37}$$

Because TVD schemes preserve the monotonicity of an initial monotone profile, they prevent the formation of spurious oscillations. In fact, one of the most useful outcomes of the development of the TVD schemes is the bridge which was established between upwind and central discretizations. This allowed the central schemes to be formulated with an amount of adapted dissipation satisfying TVD requirements. It can be shown, for instance, that the use of a dissipative coefficient with a magnitude of at least half the wave speed produces a TVD scheme, while the minimum sufficient value produces the upwind scheme. It is difficult to generalize TVD schemes to more than one dimension. One way of generalizing the use of one sided differencing to a system of equations is the Flux Vector Splitting method of Steger and Warming . The fluxes can be split into components based on the characteristic speeds and the TVD construction can be applied separately on each component. Examples of 3-D TVD solutions of the Euler equations can be found in the work of Takakura, Ishiguro and Ogawa, of The Fujitsu and National Aerospace Laboratory of Japan [62]. They conducted studies of two TVD schemes (Yee-Harten and Chakravarthy-Osher) for 3-D Euler equations expressed in general curvilinear coordinates and showed accurate solutions for the ONERA M6 wing.

2.2.5 Spectral Methods

Among the areas likely to see concentration of future CFD research in the search for higher orders of accuracy, Spectral Methods can be cited. An example is the work was done at NASA Langley in this area [63],[64]. Spectral Methods are an extreme development of the class of discretization schemes known as "Method of Weighted Residuals (MWR)". These methods use trial functions (expansion or approximating functions) as the basis functions for a truncated series expansion of the solution which, when substituted into the differential equation, produces the residual. They also use test functions (weight functions) to enforce the minimization of the residual. The choice of trial functions is what distinguishes the Spectral Methods from the finite difference and finite element methods. It has been suggested that a spectral method can require far fewer grid points than the best finite difference method to achieve engineering accuracy of a full potential solution for a lifting supercritical airfoil. The cost of a spectral multigrid method is comparable to that of the finite difference method. However, on most practical problems, in a direct comparison, finite difference methods produce better and cheaper solutions. The accuracy of today's spectral shock-capturing techniques for the Euler equations tends to be quite low. Exponentially convergent solutions to the Euler equations can be obtained by resorting to shock-fitting techniques, thus placing additional constraints on the Spectral Methods.

2.2.6 Artificial Viscosity

Von Neumann & Richtmyer introduced in 1950 the concept of artificial viscosity or artificial dissipation in order to remove the unavoidable high-frequency oscillations of second-order central schemes around discontinuities [65]. These additional terms are supposed to simulate the effects of physical viscosity on the mesh scale locally around discontinuities and be negligible (order higher than truncation error) in smooth regions. The artificial viscosity also produces a means of avoiding expansion shocks. The Von-Neumann-Richtmyer artificial dissipation terms were applied to the Lax-Wendroff scheme, modifying the flux calculation in the following form :

$$f^*_{i+\frac{1}{2}} = \frac{f_{i+1} + f_i}{2} - \frac{1}{2}\tau A_{i+\frac{1}{2}}(f_{i+1} - f_i) - D(u_i, u_{i+1}).(u_{i+1} - u_i) \qquad (2.2.38)$$

Where D is any positive function of $(u_{i+1} - u_i)$ tending to zero at least linearly with $(u_{i+1} - u_i)$. The function D must have the dimensions of velocity, which means that $D\Delta x$ has the dimensions of viscosity. Lax and Wendroff called D the *artificial viscosity.* D must be at least proportional to Δx to maintain second-order accuracy. The Von Neumann and Richtmyer artificial viscosity has the form :

$$f^{*\ with\ viscosity} = f^* - \varepsilon.\Delta x^2 \psi |\frac{\partial u}{\partial x}| \frac{\partial u}{\partial x} \qquad (2.2.39)$$

It is therefore non linear and proportional to Δx^2. When applied to the MacCormack scheme, dissipation terms are added to both predictor and corrector schemes. In this scheme, it has the effect of preventing the appearance of expansion shocks, of improving the resolution of shock waves but also of smearing contact discontinuities. There have been several attemps of using higher order artificial viscosity formulations (Third order scheme of MacCormack and Baldwin, Steger scheme, Jameson's scheme [13]. these formulations will not be discussed here, with the exception of the Jameson's scheme, which will be introduced here and further discussed in the following chapter.

Jameson's artificial dissipation.

In his finite volume explicit scheme, Jameson uses a blend of second and fourth order differences with excellent shock capturing characteristics. The same formulation has been used by Pulliam and Steger [66] into Beam-Warming codes with very good results. The corrected flux is defined by :

$$f^{*\ with\ viscosity}_{i+\frac{1}{2}} = f^{*}_{i+\frac{1}{2}} - d_{i+\frac{1}{2}} \tag{2.2.40}$$

with :

$$d_{i+\frac{1}{2}} = \varepsilon^{(2)}_{i+\frac{1}{2}}(u_{i+1} - u_i) - \varepsilon^{(4)}_{i+\frac{1}{2}}(u_{i+2} - 3u_{i+1} + 3u_i - u_{i-1}) \tag{2.2.41}$$

$\varepsilon^{(2)}$ is the non linear adaptive coefficient for the second difference. $\varepsilon^{(2)}$ controls the dissipation near discontinuities and is the the coefficient of the dominant term around shocks. Its expresssion is given by :

$$\varepsilon^{(2)} = \kappa^{(2)} max(\nu_i, \nu_{i-1}) \tag{2.2.42}$$

where :

$$\nu_i = \left|\frac{p_{i+1} - 2p_i + p_{i-1}}{p_{i+1} + 2p_i + p_{i-1}}\right| \tag{2.2.43}$$

The second difference dissipation is not sufficient to remove all oscillations in smooth regions of the flow. Therefore, a background dissipation term is added using a fourth difference. This background dissipation term was found to produce oscillations near shocks. That is the reason why it is turned off in these regions, when the other term is dominant. The adaptive coefficient of the background dissipation is given by :

$$\varepsilon^{(4)} = max(0, (\kappa^{(4)} - \varepsilon^{(2)}) \tag{2.2.44}$$

$\kappa^{(2)}$ and $\kappa^{(4)}$ are constants which are specified by the user. Typical values are $\kappa^{(2)} = (1/4)$ and $\kappa^{(4)} = (1/128)$. $\varepsilon^{(2)}$ is of the order $O(\Delta x^2)$ and $\varepsilon^{(4)}$ is of the order $O(\Delta x^4)$ in smooth regions of the flow. In these parts of the flowfield, the dissipative fluxes are of third order in comparison with the convective fluxes. In the neighbourhood of a shock wave, $\varepsilon^{(2)}$ reduces to first order or becomes of the order of one. The scheme then behaves like a first order scheme. Caughey (Cornell University) and Turkel (University of Tel Aviv) have studied the effects of various forms of dissipative terms on solution accuracy, especially on non-smooth meshes [67]. They use the spatial discretization of Jameson, Schmidt and Turkel and a solution of the difference equation by either an explicit Runge-Kutta time stepping scheme or the diagonalized Alternating Direction Implicit (DADI) method of Caughey [68]. Multigrid acceleration is used in both cases. The dissipation is provided by an adaptive blend of second and fourth differences as described above. The study shows that considerable errors in surface values can be introduced by the averaging required to calculate derived quantities of interest and that spurious entropy gains can be reduced by modulating the second difference dissipation with the local Mach number (this is the term needed to control oscillations near shock waves. By multiplying this term by a function of Mach number so that the second difference viscosity is small in regions of low speed, it is possible to prevent the occurence of large second difference viscosities near trailing edges where ν_i can be large for high angles of attack). Additional problems occur in multidimensional problems when the mesh sizes are different in the various coordinate directions. In original schemes, the artificial dissipation terms are properly scaled by multiplying them by a term which is the sum of the spectral radii of the matrix in the various coordinate directions. In a paper with Swanson of NASA Langley, Turkel discussed, in 1987, the artificial dissipation model, including boundary treatment used by central difference schemes for Euler and Navier-Stokes equations. They examined modifications of the model such as eigenvalue scaling suggested by upwind differencing [69]. In 1988, Turkel [70] proposed as scaling factor in each direction the spectral radius in that direction, thus improving the solution on highly stretched meshes. This idea was put forward again in 1990. Turkel and Vatsa [71] indicated that using the spectral radius of the Jacobian of convective fluxes to obtain scalar coefficients of artificial viscosity could add too much viscosity to the slower waves. They proposed to use instead matrix-valued coefficients for artificial viscosity which would give appropriate viscosity for each wave component. Additional studies of Jameson's artificial viscosity coefficients for multi-dimensional flow were carried out by Raj, of Lockheed [38]. He suggested two variations of the Standard Adaptive Dissipation (SAD) which he called Modified Adaptive Dissipation (MAD) and Flux-limited Adaptive Dissipation (FAD). An improved adaptive dissipation method for meshes of high aspect ratio cells is used in the present work, as suggested by Martinelli [72]. In applying artificial viscosity in Navier-Stokes solutions, one must be careful to make sure that the numerical dissipation does not overwhelm the physical dissipation. In Euler solutions, there is no physical dissipation and therefore the problem is less critical.

2.2.7 Boundary Conditions for Inviscid Transonic Flow

In order to have a well posed problem, it is necessary to impose initial and boundary conditions. In an explicit scheme, the influence of the boundary values propagate numerically one space step at a time. In an implicit scheme, if the boundary conditions are treated in an implicit way, a change in the value at the boundary influences the values at all the node points at the next time step, through the solution of the matrix system. The implementation of the boundary conditions has therefore a stronger influence on the solution with an implicit scheme. The number and the type of boundary conditions to be specified in the solution of the hyperbolic Euler equations depends on the propagation properties of the system or, more specifically on the information which is propagated from the boundary to the inside of the computational domain. The one-dimensional propagation properties of waves in the direction normal to a boundary are given by the spectrum of the eigenvalues of the Jacobians associated with the normal to the boundary. The problem is well posed if the full information on the ingoing and outgoing characteristics can be obtained from the boundary conditions imposed on the variables.

Physical boundary conditions : The variables transported from the boundaries to the inside of the domain are the only ones that can be imposed at the boundary. The other variables must be part of the solution.

Numerical boundary conditions : from the numerical point of view, it is necessary to obtain information on all the variables at the boundary before proceeding to the next step. The additional boundary conditions required are called numerical boundary conditions and have to be consistent with the physical properties of the flow and to be compatible with the discretized equations. They should not, however, influence the physical boundary conditions.

Physical boundary conditions.

For one-dimensional inviscid flow the number of physical boundary conditions that can be imposed at the boundaries is shown in the table below, according to the nature of the flow (a one-dimensional Euler equation has three dependent variables which can be designated

u_1, u_2 and u_3).

Table 1 : Number of Boundary Conditions.

Boundary type	**Subsonic flow**	**Supersonic flow**
Inlet	Two conditions u_1 and u_2	Three conditions u_1, u_2 and u_3
Outlet	One condition u_3	Zero condition

The physical conditions to be imposed are the entropy and the values of the characteristics or Riemann variables. It is more practical, however, to impose the value of physical properties such as pressures or velocities, which can be set in experiments, and then calculate the characteristic information in an iterative or approximate way.

Numerical boundary conditions.

The numerical boundary conditions remaining to impose are the difference between the total number of the dependent variables and the number of physical boundary conditions imposed. The *characteristic boundary method* adds the Riemann invariants for outgoing characteristics to the imposed physical boundary conditions in order to obtain the missing information at the boundary. The choice of the numerical boundary conditions can have a large effect on the accuracy, stability and convergence rate of a numerical method.

Non-reflecting boundary conditions.

Imposing the value of a physical property at a boundary may allow a perturbation wave to be reflected at the boundary. Consider for instance the specification of a given pressure $p^{n+1} = p^*$ at a subsonic outlet boundary, which is equivalent to imposing :

$$\Delta p = p^{n+1} - p^n = p^* - p^n = 0 \tag{2.2.45}$$

Since the incoming characteristic is $\Delta u_3 = \Delta u - \Delta p/\rho c$, imposing $\Delta p = 0$ corresponds to generating an incoming wave of intensity $\Delta u_3 = \Delta u$ reflected from the exit boundary. In these relations c is the speed of sound, u the flow velocity and ρ the density. Non reflecting boundary conditions, as described by Engquist and Majda [73], express the physical boundary conditions in manner that local perturbations propagated along incoming characteristics vanish :

$$\frac{\partial u_k}{\partial t} = 0 \tag{2.2.46}$$

or, in a discretized form :

$$\Delta u_k = 0 \tag{2.2.47}$$

for all k values corresponding to incoming waves. This procedure improves the accuracy of the flow computations. For the subsonic outlet case mentioned, a condition for non-reflecting boundary condition would be :

$$\Delta u_3 = \Delta u - \frac{\Delta p}{\rho^n c^n} = 0 \tag{2.2.48}$$

Boundary treatment for multi-dimensional Euler codes.

A general multi-dimensional computational domain will typically contain free surface boundaries (far-field, inlet, outlet) and solid surface boundaries (body surfaces). In three dimensional flow, the number of physical boundary conditions to be applied at boundary surface depends on the sign of the eigenvalues of the Jacobian matrix in a direction normal to the boundary.

2.2.7.1 Far-field Boundary Conditions

Inlet and outlet boundaries can be set be far enough from the body for the disturbance not to affect the freestream values. Since the disturbances can only be damped over large distances, these boundaries must be located at a large distance from the body (for instance : 50 chords away from an airfoil for a 2-D calculation). Unless a very coarse mesh is used near these boundaries (reducing the solution accuracy), a very large number of mesh points is then required to fill the computational domain (increasing the computing cost). It is therefore advantageous to take the external boundary as close as possible to the body but in order to do that, the far-field boundary condition must be adapted to match the far-field flow to the interior flow field. An approximation of this far field is usually obtained by introducing a perturbation field to the uniform flow and expressing it as an asymptotic series in a perturbation parameter. In the case of airfoils, far-field expansions are derived from thin airfoils with terms based on the circulation around the airfoil, calculated from the lift coefficient. Several such formulations have been derived, leading to a correction of the physical boundary conditions to account for the fact that the boundaries are located at a finite distance from the perturbed flow area. Research into effective far-field boundary condition can be found in an early paper by Bayliss and Turkel [74]. A radiation boundary condition is derived for the Euler equations linearized about a constant steady state with a mean flow. The formulation derived is also applicable to Navier-Stokes solutions since non linearities and viscosity are unimportant in the far-field. The formulation led to acceleration to steady state of the explicit calculations and to a reduction of size of physical domain. In a report for an AGARD workshop, Pulliam and Barton [33] show results indicating that the the implementation of characteristic boundary procedures using one dimensional Riemann invariants greatly reduce or nearly eliminate the dependence of the solution on the far-field boundary distance. Marcum and Hoffman of Purdue University developed a boundary condition procedure for Euler solvers which is a variation of the method of characteristics [75]. These boundary conditions were used in conjunction with a second-order accurate numerical procedure with a MacCormack explicit finite difference method for both 2-D and 3-D flow. Thomas and Salas of NASA Langley investigated two-dimensional transonic lifting flow and formulations of far field boundary conditions [76]. In this study, the airfoil was represented by a point vortex and an expansion of the linearized small disturbance equation in the far field was developed. The formulation was found effective in eliminating the dependence of the numerical results on the extent of the external boundary. An extension of the method to three-dimensional computations was also outlined. The application of Fourier series developments of the flow disturbances along the boundaries was also investigated by Giles, of MIT [77], who presented a unified theory for the construction of steady and unsteady non reflecting boundary conditions for the Euler equations. Giles investigated the mathematical basis of the boundary condition formulations and gave an interesting explanation of the relations to standard quasi-one-dimensional boundary conditions.

2.2.7.2 Wall Boundary Conditions

At a solid wall boundary, the normal velocity is zero and only one physical condition can be imposed : $v_n = 0$. The other variables must be determined from the interior of the domain. There is no mass or other convective flux penetrating the solid body and only the pressure term remains in the formulation of the fluxes at the wall. The wall pressure can be determined either by interpolation from the interior adjacent points or by solving the Euler equations with one-side discretization at the wall. The extrapolation of density and velocity to the surface from interior points can lead, in the absence of adequate grid resolution to surface boundary errors. The situation is normally improved by using a constant free stream enthalpy value at the surface and by setting boundary conditions on the velocity normal and tangential components : $V_n = 0$ and V_t extrapolated to the surface; (Pulliam and Barton [33]). A formulation of the finite volume method with node points directly on the surface has advantages in this respect since no extrapolations are required to obtain the values of the dependent variables on the body surface.

2.2.7.3 Kutta Condition

Inviscid flows around lifting bodies have an infinity of solutions depending on the circulation which is generated around the airfoil. The Kutta condition specifies that the closest approximation to the physical behavior of the flow is one with a value of the circulation such that the flow separates smoothly from the airfoil at the sharp trailing edge. In potential flow solutions, a cut simulating a vortex sheet is normally introduced downstream of the trailing edge and a potential jump equal to the circulation introduced. The value of the circulation is determined by imposing equal pressure or zero normal velocity at the trailing edge points above and below the cut. As mentioned by Rizzi and Viviand in 1981 [78], the solution of the time-dependent Euler equations does not require any Kutta condition to produce accurate results with the correct value of the circulation. This is due to the existence of a mechanism in the time-evolution of the computed Euler solutions reproducing an essential physical phenomena leading to the generation of circulation and lift. Initially, a lifting flow about an airfoil at incidence behaves in an irrotational manner and a stagnation point is located on the upper surface of the airfoil, leading to a turning of the flow around the sharp trailing edge. A very strong velocity gradient exists there in inviscid incompressible flow which tends to expand. This leads to the formation of an eddy which prevents infinite velocities, creating a surface of discontinuity (vortex sheet). The velocity induced by this eddy on the upper surface moves the stagnation point] towards the trailing edge point. After a while, the stagnation point reaches the trailing edge and the eddy is convected downstream of the airfoil, leaving a circulation around the airfoil of equal and opposite direction, in accordance with Kelvin's theorem. There are two possible mechanisms in the numerical solution of the Euler equations which can generate this vorticity at the trailing edge. A possible mechanism is the internal dissipation of the scheme needed for stability which, like physical viscosity, provides a mechanism for the numerical generation of vorticity and entropy (Pulliam and Barton [33]). A second possible mechanism, described by Rizzi [28] is the creation of an expansion fan, of shocks and of a contact discontinuity resulting from the large acceleration at the sharp trailing edge. The transient appearance of a shock wave would create vorticity and induce the vortex sheet surface to roll up into an eddy. Although no explicit Kutta condition is required in Euler solutions, the configuration of the mesh and the actual implementation of the surface boundary conditions in the vicinity of the trailing edge may have an effect on the accuracy of the prediction of lift. Pulliam and Barton [33] found that a C mesh is better than O mesh for 2-D airfoil calculations in transonic flow. This is because the trailing edge is a multi-valued point and is treated as two distinct points (upper surface and lower surface) A jump in velocity can therefore be captured at the trailing edge. In lifting transonic flow, the shock strengths differ on the upper and lower surface, leading to different total pressure jumps. The total pressure differs at the trailing edge and since the static pressure has to be the same, the velocities must differ.

2.3 APPLICATIONS OF EULER CODES

2.3.1 Nacelles, Inlets and Exhaust Nozzles

An example of application of an Euler code to the simulation of powered nacelles at Boeing can be found in reference [79]. The program used is a Jameson type cell-centered single-block solver tuned to the solution of a turbofan nacelle. Comparisons made with panel methods results at Mach 0.4 show good correlations. A model of a flow-through nacelle includes the fan cowl and the center body. For powered nacelles, the mass flux or the static pressure is prescribed at the fan face ; the total pressure, total temperature and the flow direction are specified at the fan and turbine exit planes. At high angles of attack, there is appearance of an artificial inviscid separation. This was found to be caused by an inaccuracy in the evaluation of the surface pressure boundary condition, leading to the formation of an artificial boundary layer (a thin layer of lower total pressure) adjacent to the surface. A nodal point-oriented scheme, such as the one used in the code developed here, requiring no extrapolation to compute surface pressure values, is believed to be an improvement. The National Aerospace Laboratory (Japan) and Nihon University use a finite volume explicit second-order MacCormack scheme to model inlet and exhaust flows [80]. The boundary conditions used are :

- Far-field : locally 1-D Riemann invariant boundary condition;

- Downstream boundary far behind the nacelle : zero order extrapolation in ξ direction;
- Body surface,symetric plane and axis : tangential slip flow condition - no mass flux on axis;
- Wake boundary : no condition;
- Fan inlet face boundary : one variable specified (Mass Flow Ratio) and others obtained from the solution (mass flux assumed uniform radially, velocity tangent to body of inlet and spinner and linearly varying in between. Density and circumferential velocity extrapolated. Pressure and energy given using adiabatic Bernoulli equation along a streamline.
- Jet exit boundary : total pressure ratio and total temperature ratio and MFR specified.

2.3.2 Propellers, Rotors and Propeller/Wing/Airframe Interference

Euler codes are very useful in simulating flow-fields in the wake of propellers . The automatic vortex capturing capability of the Euler solutions make unnecessary the modelling of sophisticated wakes as found in Full Potential or other panel models. An example of application of an Euler code to steady rotor flow can be found in reference [81]. In this model, the Euler equations are formulated in a rotating cylindrical coordinate system attached to the blade. Centrifugal and Coriolis forces are added, making the formulation fairly distinct from the one described here. Hall and Delaney of Allison applied an Euler analysis to the analysis of ducted propfan flowfields [82]. The equations were again cast in cylindrical coordinate systems in a rotating refererence frame. A comparison of the inviscid Euler results with experimental data indicates that the model is a very good approximation of the flow in the ducted propfan arrangement. The effect of a propeller on the aerodynamics of the

Fokker-50 aircraft was modelled by J.L. Kuijvenhoven of Fokker [83] using the multi-block Euler code developed by Amendola, Tognaccini, Boerstoel and Kassies [84]. The program is based on a the formulation of Jameson. Kuijvenhoven's paper shows very good comparisons between the computed pressure distributions and flight test data even on the portion of the wing inside the propeller streamtube.

2.3.3 High Angle of Attack Aerodynamics

The research in high-angle-of-attack aerodynamics has been dominated by two important problems : first, the occurrence of vortex breakdown and of the subsequent buffet sollicitation of the fins on military aircraft such as the F-18; the second problem is the control of airplanes at extreme angles of incidence, by controlling the forebody vortices. At the McDonnell-Douglas Corp., St. Louis, Euler and Navier-Stokes codes are being used to investigate numerically the vortex breakdown of delta wings at high angles of attack [85]. The fact that the vortex breakdown pattern of delta wings is only weakly dependent on Reynolds number suggests that it may be in essence an inviscid phenomenon (vortex instability) and therefore one that can be predicted by an Euler code. Another example of Euler solution can be found in the work of van den Berg, Hoeijmakers, Sytsma of NLR [86]. In this reference, Euler solutions are calculated at M=0.50 for a leading edge vortex flow about a 65 degrees sharp-edge cropped delta wing at incidences close to vortex breakdown. Using an O-O type body-fitted grid, it is shown that converged solutions can be obtained at M=0.5 up to an incidence of 21.25 degrees. A "solution breakdown" (no convergence or divergence) occurs at higher angles of incidence, which is said to be potentially indicative of vortex breakdown. The TEAM code (Three-dimensional Euler Aerodynamic Method) developed at Lockheed has also been used on several configurations at high angles of incidence [38], [39]. One of the references we used to validate our code for high angles of incidence flows over delta wings was that of Hilgenstock, from DLR, Gottingen, Germany [87] : the paper shows multi-block Navier-Stokes solutions for a 65 degrees sweep delta wing and shows comparisons with experimental data and previous Euler calculations by Rizzi and Muller [88].

2.3.4 Analysis of Arbitrary Aircraft Configurations

Three approaches to the treatment of complex aircraft configurations can be cited : an unstructured grid approach (Douglas), a Cartesian grid approach (AMI) and a structured multi-block approach. The Douglas Aircraft Company has experimented with a 3-D unstructured grid Euler code "AIRPLANE" developed by Jameson and Baker for complete airplane configurations [89]. The code uses a vertex centered finite volume method with the control volume made of all the tetrahedra meeting at a given node. Artificial dissipation is added and a Runge-Kutta time stepping algorithm is used. Results are shown for the ONERA M6 wing, in comparison with Jameson's FLO-67 results. A calculation was made also for the McDonnel Douglas Tri-Jet configuration, using 384,914 nodes and 2,332,022 tetrahedra. A converged solution was obtained in 300 iterations using a four-stage Runge-Kutta scheme. The solution took 29 hours and 50 minutes on a CONVEX C210 computer (similar to the one used in the present work) but only 1 hour and 47 minutes on a CRAY Y-MP. It is expected that using the 8 parallel processors of a CRAY Y-MP should permit solutions within 20 minutes. This example is cited to illustrate the type of large scale computations with Euler codes which are required in the development of aircraft configurations. The particular performance of this code will be compared with that of the code developed here in chapter 3. Israel Aircraft Industries and Analytical Methods Inc. have produced the MGAERO Euler code which makes multi-grid calculations on equally spaced Cartesian meshes with local refinement [90]. It removes the need for complex grid generation but introduces the need for sophisticated boundary condition implementation. Results have been shown for the ONERA M6, the supercritical wing research airplane and the Rockwell wing-body-canard. The code has also been tested by De Havilland and Shorts, subsidiaries of Bombardier but is believed to fall very short of the accuracy provided by the multi-block body fitted code developed in the present work. The next application of the Euler formulation to complex aircraft configurations is the work of Ranjit Collercandy of ONERA [91]. It illustrates the use of a multi-block structured body fitted grid. It is also one of the best examples we have seen of the use of upwind schemes for complex 3-D configurations. The method uses cell-centered upwind finite volume with an explicit three-stages Stetter's predictor-corrector algorithm for time stepping the Euler equations. The equations are cast in strong conservation form in Cartesian coordinate systems. Three stencils are used to discretize the fluxes in space : Van Leer's flux splitting approach, Roe's flux difference splitting approach and a higher-order accuracy Van Leer's MUSCL approach (Monotone Upwind-central Scheme for Conservation Laws). The latter scheme is obtained from a linear combination of one-sided and central interpolations. To avoid oscillations and ensure monotonicity, slope limiters are introduced. In order to prevent entropy violation in Roe's method, a fix suggested by Harten is used, limiting the modulus of the eigenvalues and using a problem dependent parameter. The applications shown in the above reference are :

- A NACA 0012 at M=0.85 and alpha=1 degree : excellent shock capturing from the upwind scheme, oscillation free.

- A 3-D ONERA M6 Wing : 2-blocks H-mesh topology : good comparison with experimental data at M=0.84 and alpha=3.06 degrees and sharp resolution of the lambda shock.
- A Transonic turbofan simulation and a model of the Hermes spaceplane configuration with winglets, alone and atop the Ariane launcher with boosters.

2.3.5 Application of Euler codes to the Design of Configurations

In a recent development by NASA [92], a practical wing and wing/body aerodynamic design tool was obtained by the coupling of a 3-D Euler code (Jameson's FLO-57 or Lockheed's TEAM) with a modified version of the QNMDIF Quasi- Newton numerical optimization method. The method was applied to the design of a cranked-Delta wing/body configuration and an arrow wing/body configuration in supersonic flow. A similar approach was used by Jurgen Schone of the German DLR to design wings at supersonic speeds [93]. In that case, the well known constrained optimization method of Vanderplaats (CONMIN) was used. With the Euler codes, CFD has joined the wind tunnel and flight test as a principal technology for aerodynamic design. An example of this is the "predictor-corrector" approach developed at NASA Langley which is used by Boeing to design transonic nacelles and winglets [94]. The predictor is the flow analyzer which in this particular case is a multi-block Euler code based on Jameson's techniques. The design driver or corrector, based on streamline curvature theory, converts the pressures computed by the flow analyzer into changes in surface curvature. A double integration of this curvature yields the new surface geometry.

2.4 CONCLUSIONS AND CHOICE OF A METHODOLOGY

As seen in the above discussion, there has been a remarkable development of techniques to solve the Euler equations in the past decade. This development, combined with the rapid improvement in computing speeds, has made possible the modelling of realistic three-dimensional inviscid flows around complete aircraft configurations. None of these methods described above has come out with undisputed advantages over the others and most of them are restricted to simple or particular cases. The methodology proposed in the present work is the result of a combination of judgement, experience and external circumstances such as the fulfillment of constraints particular to the Canadair application. The key elements in the choice of the method are discussed in the following paragraphs.

2.4.1 The Mesh : Structured Grid Versus Unstructured Grid

The advantage of *structured grids* is the simplicity and the straightforward treatment of the mesh in the solution formulation. The disadvantages are the mesh generation constraints for complex configurations. *Unstructured grids*, on the other hand, are very flexible in modelling difficult geometries. However, this is at the price of larger memory requirements and heavier bookkeeping of the node connectivities leading eventually to larger computing costs. Our experience with structured grid Euler solutions ranged from the use of 2-D codes with an O-mesh (FLO-52, FLO-82) and 3-D codes for isolated wings (FLO-67, FLO-87). On the other hand, we had experience with a 2-D unstructured mesh generation code (MESH2D) and an unstructured grid Euler solver for multi-element airfoils (FLO-72). Four reasons eventually motivated our choice of the structured grid formulation :

1. *The better confidence in structured grid results.* Although good results were produced with the unstructured grid codes, these results were found to depend heavily on the quality of the meshes generated around the wing. The structured grid results show also some grid dependence but it is one that is better understood at this time.
2. *The necessity to generate a 3-D grid generation package during a reasonable time scale.* The 3-D grid generation package was developed at the same time the solver was being written. Our experience with 2-D structured and unstructured mesh generation codes indicated that a 3-D structured grid generation code could be developed much faster and with better confidence than a 3-D unstructured grid code. We were confident that, using a block structured approach, we could develop grids accurate enough for the CL-601 aircraft configuration.
3. *The necessity to keep the memory size and computation times low.* The computations were to be carried out on a CONVEX C-220 computer with two 50 MFLOPS CPU's. The speed of this computer is about an order of magnitude less than that of CRAY machines which are typically used in the industry to produce 3-D Euler and Navier-Stokes solutions. Our experience with 2-D structured and unstructured grid Euler codes indicated that the difference in computational cost and turn-around time for a 3-D solution would be substantial, justifying the choice of the structured grid formulation.

2.4.2 The Numerical Method : Finite Volume Versus Finite Element

Finite element methods have been used successfully to evaluate the aerodynamics of aircraft, most notably at Dassault. The finite element method is at its full advantage when used with adaptive unstructured meshes. However, our experience with finite element methods for large scale computations was limited. We had more experience with *finite volume* methods [22] and *finite difference* methods which we used to solve the Transonic Small Disturbance equations for the complete CL-601 aircraft configuration [11]. In addition, the baseline Euler methods we had experience with were all formulated in finite volume. It was therefore a logical choice to use this method for the application to the complete aircraft configuration.

2.4.3 The Space Discretization : Central Versus Upwind Discretization

Central discretizations have the advantage of simplicity. The codes are simpler and easier to program. Their main disadvantage is that they require the introduction of artificial viscosity terms with parameters that must be adjusted by the user for each particular problem. A study quoted earlier indicated that with these parameters properly tuned, central difference Euler solutions can be made practically as accurate as upwind difference solutions. *Upwind discretizations* are theoretically more accurate than central difference discretizations. Their main advantage is the absence of parameters which must be adjusted by the user for a particular configuration. However, these schemes are more time-consuming to program and involve a larger number of arithmetic operations than the central schemes. Although some excellent 3-D results have been shown, the theoretical background for the multi-dimensional extension of the higher-order accurate upwind schemes still needs a better definition. In practice, the code such as the one developed here will serve in industry to model a limited number of configuration types (commercial aircraft configuration with fuselage-mounted engine and CF-18 fighter aircraft) for long periods of time. The effort spent to adjust the artificial viscosity coefficients of a central scheme for flow conditions of interest (cruise conditions and high angle of incidence condition; subsonic, transonic and supersonic speeds) is not prohibitive. A central discretization scheme was therefore selected.

2.4.4 Time Stepping : Explicit Versus Implicit Solutions

Explicit methods are simple and have minimal costs per time step. The disadvantage is that stability limitations on the step size can increase substancially the number of iterations required to reach a steady state solution. Since the time step is not constrained by stability limit, it is expected that *implicit methods* will yield convergence in a smaller number of time steps. This is an advantage only if the decrease in the number of time steps outweighs the increase in computational effort per time step. In practice, it is found that the explicit methods with appropriate acceleration techniques are more cost effective. In addition, the requirement for performing time-accurate unsteady calculations in the future is better fulfilled with an explicit formulation. In the present work, an explicit multi-stage time-integration is used. One advantage of the multistage scheme is that no startup procedure is required; the scheme can be taylored to give a desired stability region and it has proved extremely effective in practice.

2.4.5 Body-Fitted Grid Versus Rectangular Grid

the use of simple rectangular grids with its related advantages in turnaround time to obtain a solution was considered at first. The experience with the KTRAN Transonic Small Disturbance code [11] indicated that the possibility of setting up calculations very quickly was a key criteria in the usefulness of a CFD code in the aircraft design cycle. A formulation for the solution of Euler equations in rectangular coordinates is used in the MGAERO code [44]. However, it was finally decided that Transonic Small Disturbance and Full Potential codes would be used for rapid analyses and that the main use of the Euler code would be the production of the most accurate inviscid flow solution possible. For this reason, a formulation with body-fitted grids was selected.

2.4.6 Single Block Versus Multi-Block

The advantages of the multi-block approach are as follows :

- A simplified grid generation process : simple topology and grids optimized for each aircraft component.
- A reduction of the core memory required by the flow solver. Only data related to the block being solved for needs to be in memory at a time.
- Simpler logic for the flow solver.
- The possibility of zonal solutions with different types of equations in different blocks.
- The exploitation of the parallelism of the computer by solving concurrently for various blocks on parallel processors.

The disadvantages of the multi-block approach are the following :

- Possible loss of accuracy or conservation at block boundaries due to a lack of complete continuity of grid lines at block edges.
- The need for carrying a large data structure to identify block connectivity.

The multi-block approach that we propose here was designed to maximize the advantages of this formulation while minimizing the disadvantages. We will show, for instance, that our block boundary treatment did not result in loss of accuracy. The multi-block approach is the better choice in order to obtain accurate solutions on the complex configurations considered in this work.

2.4.7 Cell-Vertex Versus Cell Centered Schemes

The formulation finally selected was similar to that of Jameson [37]. The experience we had with several codes developed by Jameson was one of the main reasons for this choice. The code for complete aircraft configuration was to build on his formulation for 3-D Euler solutions on isolated wings. A choice was still to be made between using a cell-centered formulation (unknowns defined at the centers of the cells), as employed in the 2-D FLO-52 and the 3-D FLO-57, and the cell-vertex formulation used in the 3-D FLO-67 code (unknowns defined at the vertices of the cells). The cell-vertex formulation has numerous advantages, in comparison with the cell centered formulation [95] :

- In the cell-vertex formulation, the approximation of the flux through a cell side is the trapezoidal rule for integration and hence is second order accurate in cell size whatever the grid. In contrast, cell-centered fluxes are averages of the quantities at the centers of adjacent cells and are therefore second order accurate only on nearly uniform grids.
- No extrapolation is required to obtain values of the dependent variables on body surfaces leading to better predictions of the surface pressures and velocities.
- For a quadrilateral cell of side O(h), and flux balance O(h^2), the truncation error in

the flux balance is $O(h^3)$ or first order in the cell-vertex formulation, irrespective of the size of the adjacent cells. For a quadrilateral with opposing sides differing by $O(h^2)$, the truncation error is $O(h^4)$ or second order. In contrast, the truncation error of the cell-centered formulation depends on the sizes and shapes of adjoining cells (Roe [96])

- In 2-D, the cell-vertex scheme has one spurious mode (chequer-board mode), whereas the cell-centered scheme has three possible modes. It is expected therefore that the damping of spurious mode will be easier in a cell-vertex formulation.

For these reasons, we decided to use a cell-vertex formulation.

2.4.8 Formulation of the Proposed Euler Method

In summary, in order to fulfill the requirements for predicting inviscid transonic flow-fields around the Challenger CL-601 Business Jet, the following numerical solution methodology was implemented :

- Finite volume solution of time-dependent Euler equations cast in conservation form in Cartesian coordinates.
- Central space discretization with added adaptive artificial dissipation with a level controlled by the user.
- Explicit integration using a Runge-Kutta time stepping procedure.
- Multi-block body-fitted structured grid with unknowns stored at the cell vertices.

CHAPTER 3 : Geometry Modelling and Grid Generation

3.1 MODELLING OF THE AIRCRAFT GEOMETRY

The Euler code uses body-fitted grids and surface conforming boundary conditions and requires therefore the most accurate representation possible of the aircraft sur face. The geometry data of the aircraft is originally in the form of CADAM or CATIA models made of (x,y,z) points, 3-D splines or 3-D surfaces. The modelling of the geometry is done on a graphics terminal running the 3-D CADAM software. The surface of the aircraft is divided into patches. Each patch is described by lines or 3-D splines in sufficient number to define properly the geometry of the patch. A CADAM alphanumeric attribute is assigned to each one of these lines or splines, indicating to which patch it belongs. The attribute contains also the information controlling the generation of the patch surface grid and ultimately that of the 3-D grid in the external flowfield subregion associated with that patch. A CADAM half-model of the CL-601 aircraft is shown on figure 2. The patches on the wing, winglet and tailplane are described by splines defining wing sections. The fuselage and the various fairings are primarily defined by cross section cuts as sociated with longitudinal body lines outlining the patch edges. To preserve the accuracy of the geometry in the inlet lip area, the CF-34 turbofan engine is represented by longitudinal splines defined in successive radial planes located around the engine centerline, as shown more closely in figure 3. On CADAM, the user can verify the degree of continuity of the various curves and determine the intersection of curvilinear surfaces describing various aircraft components.

3.2 GRID GENERATION

The majority of single-block programs written for isolated wings have subroutines which generate, using conformal mapping or other procedures, a stacked C-grid around the wing. For more complex configurations, it was necessary to develop an independent grid generation capability. Programs were written to generate 2-D and 3-D grids using algebraic interpolations from arbitrary boundaries. Procedures were also developed to obtain smoother grids as solutions of elliptic partial differential equations. The generation of grids was validated first for single-block domains, then extended to domains decomposed into multiple blocks.

3.2.1 Algebraic Grid Generation

The development of the grid generation programs was done in a progressive manner. Starting with the basics, several unidirectional interpolation procedures were examined : Lagrange interpolation, Hermite interpolation and the use of tension splines. The AKIMA cubic spline was selected for further use in the grid generation programs. Functions other than polynomials were also examined. To control the variation of spacing along a grid line, the hyperbolic tangent function was selected since it gave the best overall point distribution. Multidirectional interpolation procedures were then evaluated. A two-directional transfinite interpolation scheme was programmed with various forms of polynomials as blending functions. The spline blended form was found to give the smoothest grid with continuous second derivatives. Using the methodology specified above, a 2-D algebraic grid generation code **GRID2D** was written to generate 2-D grids as well as curved surface grids in 3-D space. These programs can generate grids on a patch specified by 4 bounding curves or supported by a number of section cuts. A 3-D grid generator, **GRID3D**, was also written using 3-D transfinite interpolations [97]. The program can generate a grid inside a block bounded by six faces, given the geometry of 2 or 4 of the faces of the blocks. The missing faces are assembled from the boundaries of the input ones. If only two faces are specified, two of the remaining four faces are assumed to be straight.

3.2.2 Elliptic Grid Generation

Elliptic grid generation programs were also written, with a formulation based on a the work of Thompson [98], these programs rely on the inherent property of elliptic systems to produce non overlapping grids. The smoothing tendency of elliptic systems leads to high-quality grids with the advantage that boundary slope discontinuities are not propagated into the field. Additional desired grid characteristics, such as orthogonality and local grid concentration can also be applied very reliably with elliptic methods. A 2-D Elliptic grid generation program **EGRID2D** was written first. The program solves a Poisson equation, with the right hand side made of the functions designed to control the orthogonality and the stretching of the grid. These control functions are calculated from the point distribution on the boundaries. The equation is discretized using second order central differences for all derivatives and solved using successive line overrelaxation. The program was later extended to perform grid generation for curved surfaces in 3-D space [99]. The 3-D elliptic grid generation program **EGRID3D** was developed in a manner analogous to the 2-D program. The three control functions required to define the global stretching of the grid in this case are calculated from the point distributions on the six surfaces bounding the 3-D domain. linear transfinite interpolation is then used to calculate their values in the interior of the grid. To retain the desired stretching, these functions are calculated once at the beginning of the solution process and held fixed throughout all iteration levels. Local stretching in certain areas of the grid is obtained by adding a decaying exponential term on one or more of the three stretching functions. In addition to the stretching control functions, an inhomogeneous term is added to the elliptic equation in order to control the orthogonality and spacing at the boundaries [99]. An initial grid generated algebraically is used to start the calculations.

3.2.3 Multi-Block Grid Generation

A multi-block approach is used in the 3-D Euler code. In this approach, the space around the aircraft is divided into subregions or "blocks". Suitable grids are generated inside each one of these blocks and the flow equations are solved in each block in sequence. This approach has several advantages. First, it simplifies the process of grid generation since grids have to be generated inside blocks of relatively simple topology. Also, inside each block, the grid can be optimized for the particular component of the aircraft which supports the block. The amount of core memory required by the program can also be reduced since only the data relative to the block being solved for is required at a time. The logic of the flow solver can also be greatly simplified if the blocks are restricted to simple topologies with single boundary conditions on each block face. The multi-block approach was also selected with the long-term view of solving different flow equations in different blocks (zonal methods) and that of solving for different blocks in parallel on supercomputers with adequate architecture. The problems of the multi-block formulation are related to a possible loss of accuracy at the block interfaces and to reductions of the rate of convergence linked to the formulation of the interface boundary conditions. These problems are linked to the degree of continuity of the grid lines across block boundaries and to the smoothness of the grid in this area. The process of multi-block grid generation is made of the following steps :

3.2.3.1 Topological Block Decomposition

After the flow domain has been divided into blocks, the connectivity between the faces, edges and vertices of all the blocks must be properly defined. This is done using a conceptual model of the blocks defined around the aircraft configuration according to the following conventional rules :

- All blocks are conceptually similar to a cube, with six faces. Some of the faces may be collapsed onto a singular line or a singular point.
- The grid inside a block is an H-H type grid. Each block face corresponds therefore to a constant value of one of the grid indices i, j or k.
- The blocks are simply connected with no cuts or solid surfaces inside. Boundary conditions are applied only at the faces of the block and only one type of boundary condition can be specified on one face.
- There is no gap or overlap between adjacent blocks which must be perfectly matched at their common interface. A block face cannot be adjacent to more than one other block face. In axisymmetric configurations, the face of a block can be adjacent to the opposite face of the same block.

The faces, edges and vertices of each block are numbered in the order shown in figure 4 and given an attribute specifying their connectivity to the other blocks in the domain. The conceptual model used for the topological block decomposition of the Challenger wing/body/winglet configuration in 40 blocks is shown in figure 5(a).

3.2.3.2 Physical Block Decomposition On CADAM

In a second step, the space around the 3-D CADAM model of the aircraft surface is physically decomposed into blocks connected in the manner specified in the conceptual model (figure 5(b)). In the physical space, the block boundaries are curvilinear surfaces defined by edges or sectional cuts. These are lines or splines known on CADAM as "elements". The edges forming a face are grouped in "face sets" and those belonging to a block are grouped in "block sets". The connectivity attributes are then attached to each element, face set and block set. The number of grid points to be generated in the i, j and k direction and the desired grid point spacing are also coded as part of the attributes of each block set.

3.2.3.3 Automatic Geometry and Topology Extraction From CADAM

A program called MBLOCK was written to extract from CADAM the (x, y, z) coordinates of all the points on the block elements. The program checks for errors or inconsistencies that may have occurred in the coding of the topological information and outputs two data files : a topology file to be used by the grid generation program and the Euler flow solver and a second file with the geometry of the block elements written in a format corresponding to the defined topology.

3.2.3.4 Multi-Block Algebraic Grid Generation

In the next step, the **GRID3D** program is used to generate automatically the algebraic grid inside each block (figure 5(c)). The only constraint of the algebraic grid generation procedure is that blocks sharing the same face share the same grid points on that face, as specified in the topology data. The implementation of the slope continuity of the grid lines across block boundaries is left to the elliptic smoothing step.

3.2.3.5 Multi-Block Elliptic Grid Generation

The elliptic grid generation program **EGRID3D** was extended to read the multi-block topology file, the multi-block algebraic grid coordinates and produce multi-block grids with complete continuity at block interfaces and smooth continuous surface grids on the boundaries. To achieve this, the **EGRID3D** was allowed to smooth across the boundaries of contiguous blocks forming a simply-connected domain. The continuity of the grid lines across the outer boundaries of such groups of blocks is enforced by applying orthogonality conditions at these boundaries. The grouping is done on CADAM by attaching an group number to each block. At the end of the grid generation process, all groups are disassembled and the grid coordinates are output for each block independently. Several algorithms were added to the program to check the quality of the grid in a quantifiable manner. The orthogonality of the grid lines, the skewness of the grid cells, the relative step size of adjacent cells and the aspect ratios of the various cells are some of the characteristics that can be used to judge the quality of a grid. One effective measure of the grid quality is the normalized Jacobian of the transformation at every point in the grid. At a point where this Jacobian is equal to 1, the grid is fully orthogonal. A negative value of the Jacobian signals a negative cell volume or area. A positive value of the Jacobian close to 1 is therefore desired at every grid point. The grid generation program outputs the value of the average Jacobian in each block and the location and value of the minimum Jacobian in the block. The average and maximum cell expansion ratios and the maximum cell aspect ratio are also given for each block. These parameters point to the locations of the grid where improvements are needed. However, the final determination of the grid overall quality is done by visual inspection on a graphics workstation and through the analysis of the results of the flow solver using the grid.

3.2.4 Multi-Block Grid Generation Package

The body-fitted grids used in the applications presented here were generated using the CAD-based **MBGRID** package developed at Canadair over the past three years. The **MBGRID** package consists of three separate modules : the CAD-interface module **MBLOCK**, the 3D multi-block algebraic grid generator **GRID3D** and the 3D multi-block elliptic grid generator **EGRID3D**. The first program, **MBLOCK**, is a CAD interface which allows the construction of grids directly on the geometric model of the aircraft on the CAD system. There are two main advantages in generating grids on the CAD system: a) the retention of an accurate representation of the aircraft geometry, and b) the availability of a complete set of CAD tools to construct the domain decomposition around the aircraft model (i.e. generate surfaces, sectional cuts, block edges, etc.). This is considered an important asset of the **MBGRID** package, since the task of body-fitted grid generation involves a substantial amount of geometric modelling and manipulation. The **MBLOCK** program takes also advantage of the interactive features of the CAD system to permit the manipulation and visualization of different parts of the grid construction. A system was devised to allow individual blocks or block-faces to be viewed and processed in isolation. In addition, all the information necessary to define the grid can be specified at the CAD screen in the form of element or group attributes. In this way, the information associated with the grid construct is permanently stored in the CAD file and can be re-used to quickly modify the grid when necessary. The **MBLOCK** program allows the generation of multi-block grids with a general structure, including block-faces with multiple neighbours and/or multiple boundary condition types. The program has many built-in features designed to minimize the task of grid generation. For instance, the program assigns default values for most parameters defining the grid. It also deduces automatically much of the topological information associated with a grid, i.e. automatically generates block numbering, connectivity information, block-face-element association, sectional cut numbers, etc.. Other important features include :

- Auto-slaving of element spacings : if the spacing on one element is changed on CADAM, it is automatically changed on all the elements slaved to the first one in a group. This reduces the number of operations when modifying grids.
- Comprehensive error checking. The program checks the validity of the topology information coded on CADAM and flags any discrepancies between the information attached on interacting blocks.

Recent developments of the **MBLOCK** module [100] have allowed the generation of grids with a very large number of blocks. This was accomplished by defining a block "super-structure" consisting of block groups. Groups are defined as collections of blocks forming simply-connected, hexahedral domains. Typically, the number of groups is much smaller than the number of blocks. The new version of **MBLOCK** allows the generation of grids at the "group level", i.e. groups are constructed on the CAD system instead of individual blocks, where each group is defined by its six faces. Group "faces" can have partial or multiple neighbours and/or boundary conditions. Once the groups are defined on the CAD system, the program then automatically subdivides each group into blocks that individually satisfy the topological conventions of the **MBTEC** Euler flow solver. All connectivity information associated with the block sub-structure is automatically deduced by **MBLOCK** from the information specified at the group level. Once the grid construction has been defined on the CAD system, **MBLOCK** then generates an input file for the multi-block algebraic grid generator (**GRID3D**), and a topology file for both the elliptic grid generator (**EGRID3D**) and the **MBTEC** Euler flow solver. The algebraic grid generator, **GRID3D**, is the second module of the MBGRID package. The program reads the input file generated by **MBLOCK** and uses three-way transfinite interpolation to generate the grid. **GRID3D** can generate multi-block grids in arbitrary hexahedral domains that may include any number of degenerate boundaries. The program also allows the specification of block faces with segmented or discontinuous boundaries. The program is very efficient computationally. It can generate a 1-million-point grid in less than 30 seconds on the CONVEX C220. The grids generated with the **GRID3D** module are used as starting grids for the elliptic grid generator. The **EGRID3D** elliptic grid generator is the third module of the package. The program solves a system of elliptic equations to generate grids in arbitrary hexahedral domains that may also include any number of degenerate boundaries. This elliptic grid generator is very versatile, allowing the user to orthogonalize, cluster, uncluster, specify variable spacings on any boundary, merge blocks or split block-groups, and smooth any given algebraic grid, including the smoothing of arbitrary surfaces in 3D space. It also provides detailed grid quality diagnostics, including the Jacobian, the expansion ratios and aspect ratios of every cell. The code has been vectorized and paralellized for the 2-CPU CONVEX C220. The pogram can smooth a 1-million-point grid in approximately 40 minutes CPU on the CONVEX C220.

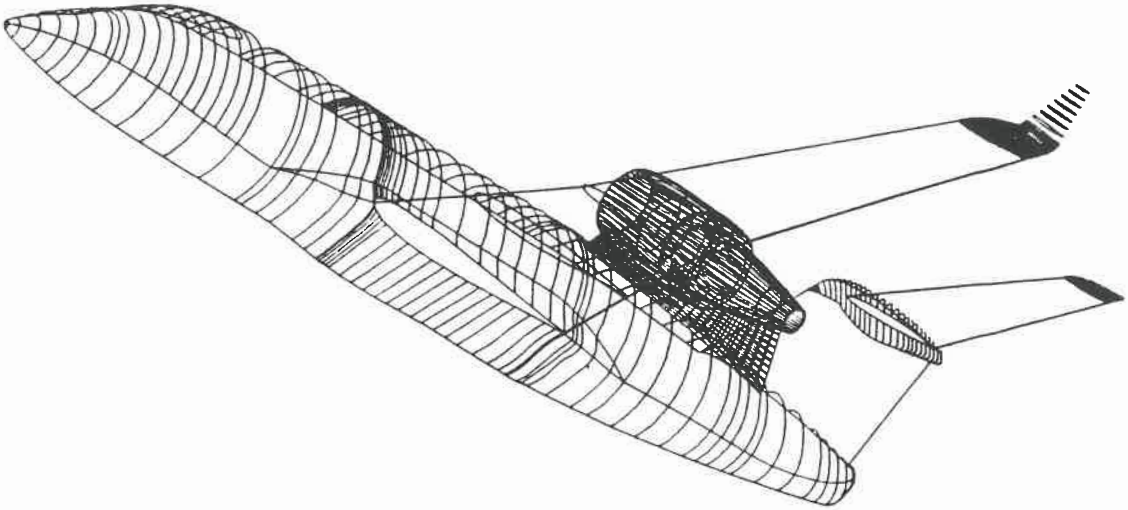


Figure 2 : CADAM model of the CL-601 Challenger aircraft.

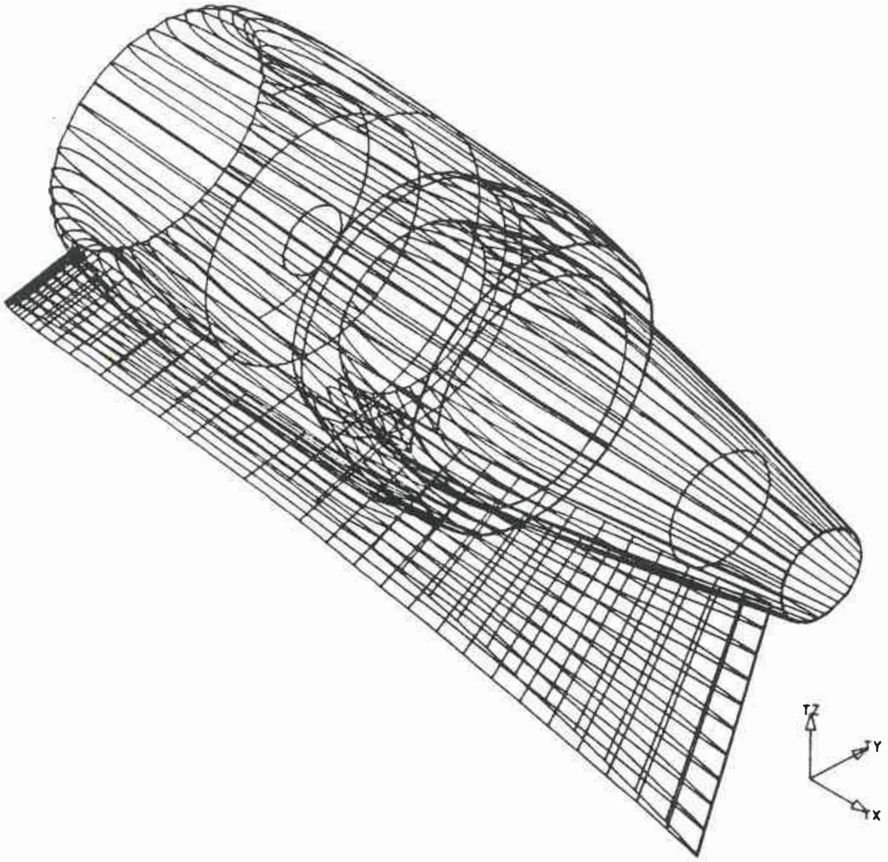


Figure 3 : CADAM model of the CF-34 turbofan nacelle.

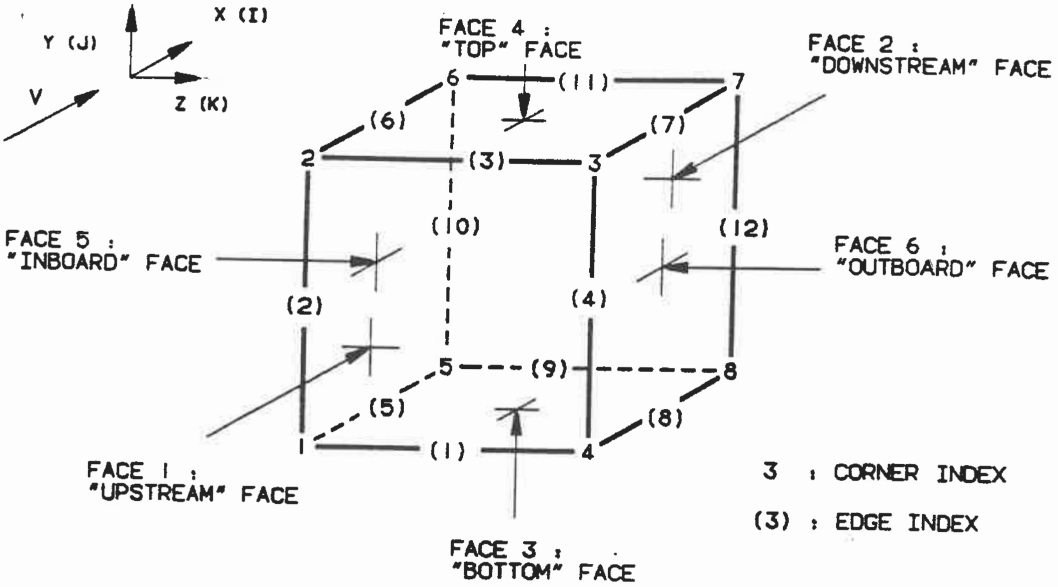


Figure 4 : Conventional numbering of block faces, edges and corners

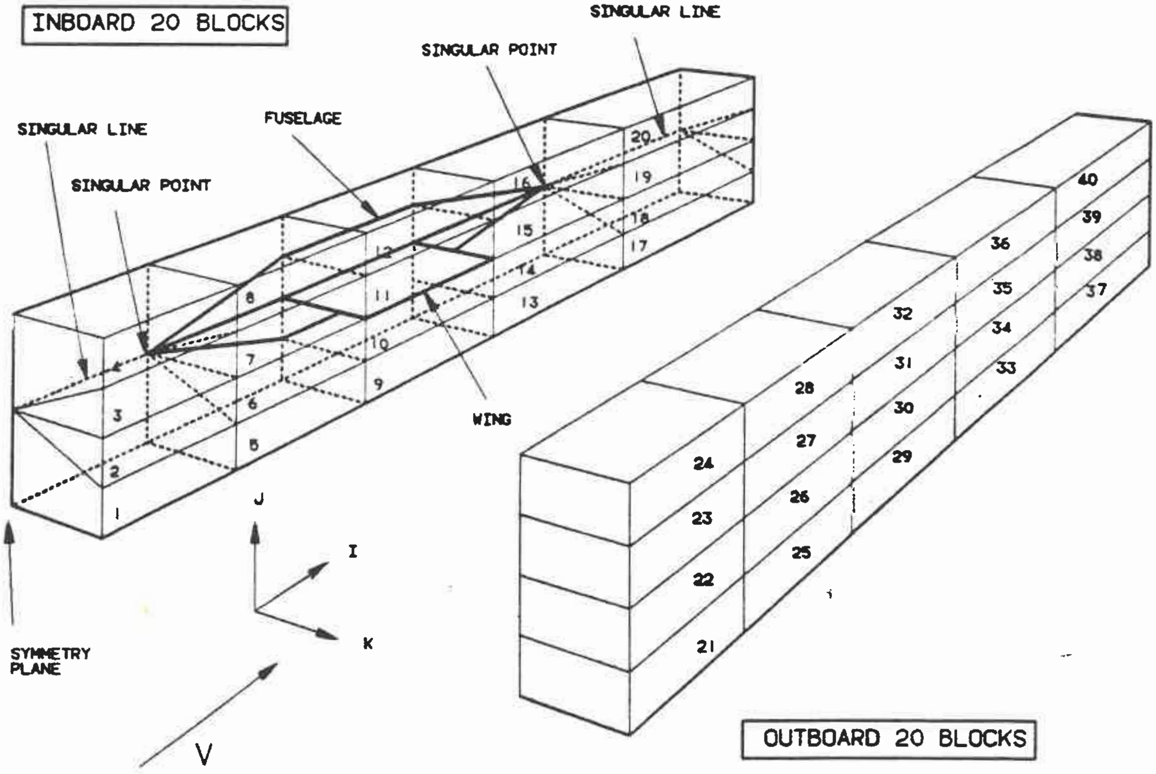


Figure 5(a) : Topological block decomposition for the Challenger wing/body/winglet

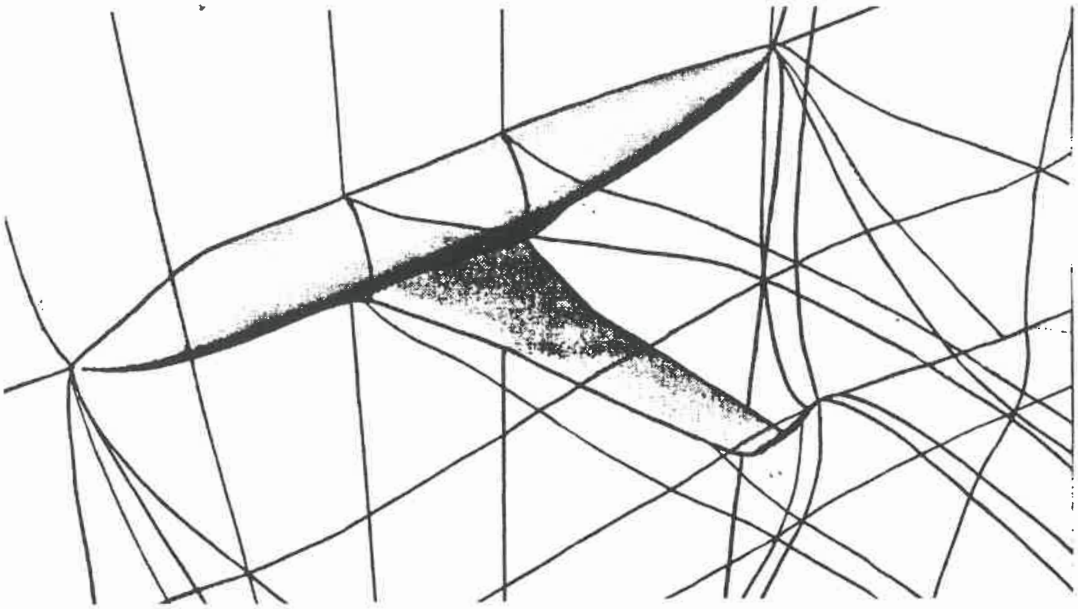


Figure 5(b) : Physical block decomposition for the Challenger wing/body/winglet

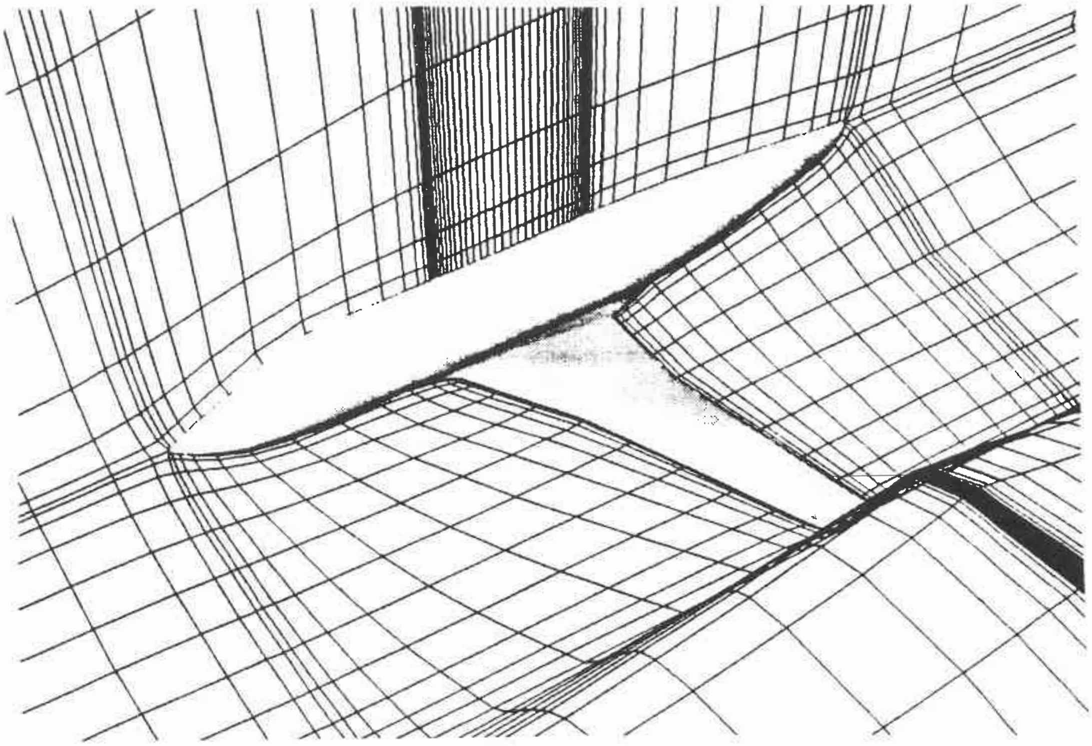


Figure 5(c) : View of the intermediate grid generated for the Challenger wing/body/winglet

CHAPTER 4 : Equations, Discretization, Time Integration

The discretization and solution methods used in the present work were derived from the baseline single-block numerical algorithm proposed by Jameson, Schmidt and Turkel [37]. The algorithm is presented in this chapter. The following chapter will be devoted to the development of the Multi-Block method using this algorithm which was the main object of our investigations.

4.1 FLOW EQUATIONS

In a Cartesian coordinate system of the three spatial dimensions (x, y, z), the time dependent Euler equations can be written :

$$\frac{\partial \vec{W}}{\partial t} + \frac{\partial \vec{F}}{\partial x} + \frac{\partial \vec{G}}{\partial y} + \frac{\partial \vec{H}}{\partial z} = 0 \qquad (4.1.1)$$

where $\vec{W}$ is the vector of dependent variables and $\vec{F}$, $\vec{G}$ and $\vec{H}$ are the convective flux vectors.

$$\vec{W} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix} , \quad \vec{F} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ \rho u w \\ \rho u H \end{pmatrix} , \quad \vec{G} = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ \rho v w \\ \rho v H \end{pmatrix} , \quad \vec{H} = \begin{pmatrix} \rho w \\ \rho u w \\ \rho v w \\ \rho w^2 + p \\ \rho w H \end{pmatrix}$$

p is the pressure, ρ the density and u, v and w are the velocity components in the x, y and z directions. E is the total energy and H the total enthalpy. For a perfect gas with a ratio of specific heats γ the pressure p and total enthalpy H are obtained through :

$$p = (\gamma - 1)\rho[E - \tfrac{1}{2}(u^2 + v^2 + w^2)] \qquad (4.1.2)$$

and :

$$H = E + \frac{p}{\rho} \tag{4.1.3}$$

The program uses a finite volume formulation. The system of Euler equations is integrated to represent the total flux balance in a control volume Ω enclosed by the surface $\partial\Omega$:

$$\frac{\partial}{\partial t}\int\int\int_{\Omega} W d\Omega + \int\int_{\partial\Omega} F dS = 0 \tag{4.1.4}$$

4.2 DISCRETIZATION OF THE EULER EQUATIONS

The solution is obtained by discretizing the flow domain into a large number of hexahedral cells and by applying the integration described above to each cell. The unknown variables are defined at the nodes of the grid. Each node (i,j,k) has a control volume consisting of the eight cells meeting at that node. At each node, the integrated equation can be written as :

$$\frac{\partial}{\partial t}(V_{i,j,k}W_{i,j,k}) + Q(W)_{i,j,k} = 0 \tag{4.2.1}$$

$V_{i,j,k}$ is the sum of the volumes of the eight cells meeting at node (i,j,k) and $Q_{i,j,k}$ is the net flux through the surface bounding the control volume made of the eight cells.

$$Q_{i,j,k} = \sum_{\ell=1}^{6} F_\ell . S_\ell \tag{4.2.2}$$

where F_ℓ is the mean flux vector across the face ℓ of the control volume and S_ℓ denotes the directed area of the ℓ^{th} face. The finite-volume spatial discretization reduces, for a Cartesian grid, to a central-difference scheme which is formally second order accurate. To suppress odd-even point oscillation modes and oscillations near discontinuities such as shocks, a dissipation term that goes to zero in the limit of zero mesh spacing is added.

The dissipation has a low background level to suppress odd-even point oscillations and it is increased near discontinuities in the flow field. This dissipation is constructed in a manner that preserves the conservation form of the equations : a dissipative flux $D_{i,j,k}$ which is the sum of three terms constructed for each of the computational coordinate directions, is subtracted from the $Q_{i,j,k}$.

$$\frac{\partial}{\partial t}(V_{i,j,k}W_{i,j,k}) + Q(W)_{i,j,k} - D(W)_{i,j,k} = 0 \tag{4.2.3}$$

With :

$$D(W)_{i,j,k} = D_x(W)_{i,j,k} + D_y(W)_{i,j,k} + D_z(W)_{i,j,k} \tag{4.2.4}$$

The contribution from the i-coordinate direction is :

$$D_x(W)_{i,j,k} = d(W)_{i+1,j,k} - d(W)_{i,j,k} \tag{4.2.5}$$

With :

$$d(W)_{i,j,k} = R_{i,j,k}.[\varepsilon^{(2)}_{i,j,k} - \varepsilon^{(4)}_{i,j,k}.\delta^2_x].[W_{i,j,k} - W_{i-1,j,k}] \tag{4.2.6}$$

In the above equation, δ^2_x is a second difference operator. The scaling factor $R_{i,j,k}$ is based on an estimate of the maximum local wave speed. The artificial dissipation scaling coefficient is redistributed between the three coordinate directions to avoid an imbalance of coefficients in these directions in meshes with high aspect ratio cells [72]. The coefficients $\varepsilon^{(2)}_{i,j,k}$and $\varepsilon^{(4)}_{i,j,k}$ are the second and fourth order adaptive dissipative coefficients. The dissipation near discontinuities is controlled by $\varepsilon^{(2)}_{i,j,k}$:

$$\varepsilon^{(2)}_{i,j,k} = \kappa^{(2)}.max(\nu_{i,j,k}, \nu_{i-1,j,k}) \tag{4.2.7}$$

$\nu_{i,j,k}$ is a sensor of the presence of a shock wave constructed by taking the second differences of the pressure :

$$\nu_{i,j,k} = \left|\frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}}\right| \tag{4.2.8}$$

and the background dissipation is controlled by $\varepsilon^{(4)}_{i,j,k}$:

$$\varepsilon^{(4)}_{i,j,k} = max(0, (\kappa^{(4)} - \varepsilon^{(2)}_{i,j,k}) \qquad (4.2.9)$$

The background dissipation is therefore switched off in regions where the second order dissipation is greater than $\kappa^{(4)}$. Typical values for the constants $\kappa^{(2)}$ and $\kappa^{(4)}$ are (1/4) and (1/128) respectively. Similar expressions are used for the dissipation in the j and k directions.

4.3 TIME INTEGRATION

In a fixed computational mesh, the control volume $V_{i,j,k}$ is independent of time . The integrated equation can therefore be written :

$$\frac{\partial}{\partial t}(W_{i,j,k}) + R(W) = 0 \qquad (4.3.1)$$

where R(W) is the residual, written as :

$$R(W) = \frac{1}{V_{i,j,k}}(Q_{i,j,k} - D_{i,j,k}) \qquad (4.3.2)$$

$Q_{i,j,k}$ and $D_{i,j,k}$ are the total convective flux and the total dissipative flux through the control volume $V_{i,j,k}$. The steady state solution is obtained by integrating the previous equation with a well proven multistage Runge-Kutta scheme [101]. For steady flows, a locally varying time step Δt is used. The maximum permissible time step is dictated by the Courant-Friedrichs-Lewy (CFL) condition and is calculated at every iteration. A five-stage scheme is presently used, computing each variable W with the following recurrence formula in which $\alpha_1 = 1/4$,$\alpha_2 = 1/6$,$\alpha_3 = 3/8$,$\alpha_4 = 1/2$, and $\alpha_5 = 1$:

$$W^{(0)} = W(t)$$

$$W^{(1)} = W(t) - \alpha_1 \Delta t R(W^{(0)})$$

$$W^{(2)} = W(t) - \alpha_2 \Delta t R(W^{(1)})$$

$$W^{(3)} = W(t) - \alpha_3 \Delta t R(W^{(2)})$$

$$W^{(4)} = W(t) - \alpha_4 \Delta t R(W^{(3)})$$

$$W^{(5)} = W(t) - \alpha_5 \Delta t R(W^{(4)})$$

$$W(t + \Delta t) = W^{(5)} \tag{4.3.3}$$

The artificial dissipation fluxes are calculated only at stages 1, 3 and 5 and frozen at their previous values at the other stages of the scheme. This was found to decrease computing time and increase the stability margin.

4.4 CONVERGENCE ACCELERATION TECHNIQUES

4.4.1 Residual Smoothing

The limitation of the maximum permissible time step imposed by the Courant number can be relaxed if each residual is replaced by an average of its neighbours. The greatest acceleration of the convergence is obtained if the residuals are averaged implicitly.

The following equation is solved locally :

$$(1 - \varepsilon_x \delta_x^2)(1 - \varepsilon_y \delta_y^2)(1 - \varepsilon_z \delta_z^2)\bar{R} = R \tag{4.4.1}$$

where R is the unsmoothed residual, δ_x^2, δ_y^2 and δ_z^2 the second difference operators and ε_x, ε_y and ε_z the smoothing coefficients given in input to the program.

4.4.2 Enthalpy Damping

Another convergence acceleration technique is to exploit the difference between the transient value of the locally computed enthalpy H and the steady state value which is constant and equal to its value at infinity H in homoenthalpic flow. Assuming that the rate of change of each variable is proportional to this difference, this information is used at the end of each time step to obtain an improved estimate of a variable W [102]. The enthalpy in the jet exhaust of a turbofan nacelle is different from that of the freestream. In this case therefore, the enthalpy damping scheme is switched off.

CHAPTER 5 : Multi-Block Boundary Conditions

5.1 FAR-FIELD BOUNDARY CONDITIONS

At the outer boundaries of the grid, appropriate inflow, outflow or side flow boundary conditions are imposed. To attenuate the spurious reflection of outgoing waves, the treatment of these boundary conditions is based on the introduction of the Riemann invariants for a one dimensional flow normal to the boundary. At a subsonic inflow boundary, four conditions are specified corresponding to incoming characteristics. At a subsonic outflow boundary, one condition is specified corresponding to the single incoming characteristic. The remaining conditions are determined by extrapolation from the interior. The program allows also supersonic boundary conditions in the far field. In this case, all the flow quantities are specified at an inflow boundary and they are all extrapolated from the interior at an outflow boundary. In the multi-block program, far field boundaries can be located on any one of the six faces of any block and are identified by the parameter **IBC = 0** (side boundary), **IBC = 1** or **2** (inflow or outflow boundary). The number of boundary conditions applied at the inflow and outflow boundaries depends on the nature of the flow at the boundary (subsonic or supersonic) and on the number of incoming and outgoing characteristics evaluated at the boundary. The formulation of the far field **inflow and outflow boundary conditions** in subroutine **BCFAR** is described here, with the subscript "e" indicating values extrapolated from inside the domain, the subscript "0" values calculated from the specified inflow or outflow conditions and the subscript "f" the final values calculated at node points on the boundary. p_e, ρ_e and q_e are the values of the pressure, density and velocity modulus calculated inside the domain, at node points adjacent to the boundary. A velocity normal to the boundary is first calculated using velocity components from inside the domain :

$$q_{n\ e} = \vec{u_e}.\vec{n} \tag{5.1.1}$$

where $\vec{n}$ is the normal to the boundary. The speed of sound calculated for the extrapolated local conditions is :

$$C_e = \gamma \frac{p_e}{\rho_e} \tag{5.1.2}$$

The values of p_0, q_0 and ρ_0 and of the speed of sound C_0 corresponding to the far field conditions are evaluated at the beginning of the calculations from the specified free stream conditions. A normal velocity component corresponding to the free stream conditions is defined as :

$$q_{n\ 0} = \vec{u_0}.\vec{n} \tag{5.1.3}$$

The Riemann invariant ER from interior values (i.e. outgoing wave) and the invariant FR from exterior specified conditions (i.e. incoming wave) are calculated as :

$$ER = q_{n\ e} - 2\frac{C_e}{(\gamma - 1)} \tag{5.1.4}$$

$$FR = q_{n\ 0} + 2\frac{C_0}{(\gamma - 1)} \tag{5.1.5}$$

The normal velocity at the boundary is then determined by the expression :

$$q_n = \frac{ER + FR}{2} \tag{5.1.6}$$

This velocity is used to determine whether there is inflow or outflow at the boundary. For a **subsonic inflow**, the three components of the velocity at the boundary are calculated as :

$$u_f = u_0 + (q_n - q_{n\ 0}).n_x \tag{5.1.7}$$

$$v_f = v_0 + (q_n - q_{n\ 0}).n_y \tag{5.1.8}$$

$$w_f = w_0 + (q_n - q_{n\ 0}).n_z \tag{5.1.9}$$

The entropy S_f at the boundary is set at its free stream value S_0. For a **subsonic outflow**, the velocity components are calculated using the expressions :

$$u_f = u_e + (q_n - q_{n\ e}).n_x \tag{5.1.10}$$

$$v_f = v_e + (q_n - q_{n\ e}).n_y \tag{5.1.11}$$

$$w_f = w_e + (q_n - q_{n\ e}).n_z \tag{5.1.12}$$

The entropy S_f at the boundary is calculated in this case from values extrapolated from the interior :

$$S_f = \frac{\rho_e^\gamma}{p_e} \tag{5.1.13}$$

The final value of the velocity modulus at the boundary is :

$$q_f = \sqrt{u_f^2 + v_f^2 + w_f^2} \tag{5.1.14}$$

The final values of the density, pressure and of total energy are eventually calculated using the following expressions :

$$(CC) = \frac{(\gamma - 1)}{\gamma}(H_0 - \frac{q_f^2}{2}) \tag{5.1.15}$$

$$\rho_f = (S_f(CC))^{\frac{1}{\gamma-1}} \tag{5.1.16}$$

$$p_f = \rho_f(CC) \tag{5.1.17}$$

$$\rho E_f = \frac{p_f}{\gamma - 1} + \rho_f \frac{q_f^2}{2} \tag{5.1.18}$$

On a **supersonic inflow** boundary, all five conservative variables are given their free stream values. On a **supersonic outflow** boundary, the five conservative variables are extrapolated from the interior. On a **side boundary**, the variables are simply equated to their values far upstream.

5.2 SOLID SURFACE AND SYMMETRY BOUNDARY CONDITIONS

At node points lying on a solid boundary, flow tangency is enforced at the end of each time step by setting to zero the normal component of velocity. The fluxes through the cell faces lying on such a surface are set to reflect this condition. In addition, when only half of the flow field is being considered, symmetry is enforced on the plane of symmetry of the configuration. In the multi-block code, any one of the six faces of a block can be a solid surface or part of a plane of symmetry. The program checks at every iteration the boundary condition parameter **IBC** of each face and identifies the faces which are either on a solid surface ($\mathbf{IBC} = 4$) or on a plane of symmetry ($\mathbf{IBC} = \mathbf{3}$). The velocity components on the cells belonging to these faces are adjusted to be tangent to the boundary. Subroutine **BCIMP** forces also the velocity at the junction between two such boundary planes to be tangent to the direction of the intersection. This is the case at the wing-body junction and at the junction between the fuselage and the plane of symmetry. The H-H topology of the multi-block program results in blocks having a single grid line or a single grid point located on a solid surface. These are blocks which faces are not labeled with an **IBC** parameter corresponding to **3** or **4**, but with a parameter equal to **6** for block interface. However, in order to preserve consistency with the calculations in the neighbouring blocks, the program has to apply the solid surface boundary conditions at those points. It was therefore necessary to write a procedure to interrogate all the face, edge and corner neighbours of a block to determine whether any one of them has a face on a solid surface. The program then determines whether a grid line or a grid point of the current block is also located on that surface and raises a flag. Subroutine **BCIMP** reviews all the flags and applies boundary conditions at the appropriate points. Our experience is that no satisfactory convergence of the calculations can be obtained unless these boundary conditions are properly applied in all blocks. In summary, there cannot be node points where solid surface boundary conditions are applied in some of the blocks to which they belong and not applied in others. As discussed in chapter 2, **no Kutta condition is needed in the Euler code at the trailing edges of lifting surfaces.**

5.3 ENGINE INLET BOUNDARY CONDITIONS

One of the most important objectives of the present development was the modelling of powered engine nacelle flow fields. This required the implementation in the code of new types of boundary conditions for the engine inlet and exhaust. The definition of accurate and stable boundary conditions required a fair amount of experimentation. This experimentation was carried out using single-block grids around various configurations of an isolated nacelle. These test cases provided substantial topological challenges requiring adjustments in the program being developed. The results of these tests were subsequently incorporated to the multi-block code. The development of the nacelle boundary conditions was carried out in two phases :

- **Phase I** : Calculation of flows near an engine inlet.
- **phase II** : Modelling of both engine inlet and engine exhaust boundary conditions.

A subsonic engine inlet boundary condition is used in the code. At such boundary, where the flow exits the computational domain, only one boundary condition can be specified. The engine inlet **Mass Flow Ratio** (MFR), also known as the inlet Streamtube Area Ratio, is input to the program and is used to calculate and impose an average mass flow through the engine inlet face :

$$Q_n = (\rho q_{n\ fan}) = (\rho q)_\infty.(MFR).\frac{A_{hilite}}{A_{fan}} \tag{5.3.1}$$

Where A_{hilite} is the cross sectional area of the inlet highlight and A_{fan} is the cross section area of the fan face. The remaining parameters are extrapolated from the inside of the domain. The development and testing of this inlet boundary condition were carried out using a single-block model of a nacelle inlet. This approach was selected in order to discriminate the effect of the boundary condition on the code stability from the effects of the multi-block formulation.

5.3.1 Nacelle Inlet Grid

The geometry of the nacelle inlet configuration was defined as shown in figure 6 with the outer cowling continued as a solid wall to the far field downstream. This topology is quite different from one that would be used typically for an isolated wing, as shown in figure 7. Figure 6 shows the grid patching and will help to identify the key points of the grid structure. To simplify the generation of the grid, an axisymmetric nacelle geometry was used. The grid was generated in a two-dimensional mode around a cut through the nacelle and the three-dimensional grid was simply obtained by rotating the two-dimensional grid around the axis of symmetry of the nacelle. The grid was made to cover the right half of the nacelle. It was thus possible to retain the formulation of the original program with a vertical plane of symmetry. The grid was constructed using the algebraic and elliptic grid generators also developed at Canadair. For testing purposes, a relatively coarse grid was specified, with only 64 × 16 × 16 cells. The resulting grid is shown in figure 8, with a closer view of the nacelle lips shown in figure 9. The characteristics of the nacelle inlet grid are as follows (figure 6) :

- The engine inlet plane is represented by the grid plane **i = 1**.
- The cowling of the nacelle is represented by the grid plane **j = 1**. In this plane, the **i** index varies from **i = 1** at the engine inlet face to the maximum **i = IL** at the far-field boundary downstream. The leading edge of the nacelle is located at **i = NX/4)+1** where **NX** is the total number of grid cells in the **i** direction (**NX = IL-1**).
- All the points on the **j = JL** grid plane, between **i = 1** and **i = NX/2)+1** are collapsed onto the axis of symmetry of the engine. All the grid cells with one face on the axis of symmetry have one face degenerated into a single line. The grid points on the **j = JL** grid plane between **i = NX/2)+1** and the maximum **i = IL** constitute the outer boundary on which an inflow boundary condition is specified.
- The far-field outflow boundary is represented by the **i = IL** grid plane.
- In the new grid, the plane of symmetry is made of two grid planes : the upper half is grid plane **k = 1** and the lower half grid plane **k = KL**. In an isolated wing grid case, the plane of symmetry is normally entirely covered by the **k = 1** grid plane.

Several developments had to be incorporated to the starting isolated wing program in order to perform calculations in the computational domain defined by this nacelle inlet grid. These developments included :

- The implementation of symmetry boundary conditions at the **k = KL** grid plane, instead of the usual side boundary conditions.
- The specification of adequate boundary conditions at the engine inlet plane.

- The implementation of the solution for the singular line at the axis of symmetry.
- The implementation of inflow boundary conditions on only part of the **j = JL** grid plane.

5.3.2 Symmetry Boundary Conditions at k = KL

In an isolated wing topology as shown in figure 7, the **k = KL** plane is the outboard far-field side boundary of the domain. The variables at the node points in this plane are simply set equal to their upstream values. In the isolated nacelle inlet case, the **k = KL** grid plane is part of the plane of symmetry. The values of the variables in the plane of symmetry are calculated through the same time-integration procedure applied at node points inside the domain. The symmetry boundary conditions are applied by specifying a zero mass flux through this boundary and limiting the momentum fluxes to their component due to the pressure acting on the symmetry plane. An additional condition is applied at the junction between the symmetry plane and the nacelle cowling. The flow there is forced to follow the direction of the intersection between these two surfaces. These conditions were implemented at both the **k = KL** and the **k = 1** grid planes. The specification of symmetry boundary conditions at the **k = KL** plane has further repercussions in other parts of the code. The limits of the flow calculation had to be extended in several subroutines to **k = KL**, instead of the limit **k = KL-1** used for isolated wings, for which the variables at the side plane **k = KL** are set equal to the upstream values. This required modifications to subroutines **EFLUX** (Euler flux calculations), **DFLUX** and **DFLUXC** (calculation of artificial viscosity fluxes), **PSMOO** (residual averaging) and **EULER** (time integration, enthalpy damping).

5.3.3 Inlet Boundary Conditions

A new subroutine, **BCINLET**, was written to impose the engine inlet boundary conditions on the grid points located in the i = 1 grid plane. It was assumed, during the course of this exercise, that the flow velocity in this plane is subsonic. At this boundary, the flow exits the computational domain. On such a subsonic outflow boundary, only one boundary condition can be imposed. The program solves for five basic unknowns : the density ρ, the velocity components u, v and w and the total energy E. As only one boundary condition can be specified in the inlet plane, the information on the four remaining parameters must be extrapolated from the inside of the domain. The operating condition specified for an engine inlet is the Mass Flow Ratio (MFR) at the nacelle highlight plane (leading edge plane). The value MFR must be given in the input data. The boundary condition is imposed at the engine fan face, located further aft on the nacelle than the highlight plane. The condition is imposed in the form of an average mass flow $Q_n.A_{fan}$ at the fan face :

$$Q_n = (\rho q_{n\ fan}) = (\rho q)_\infty.(MFR).\frac{A_{hilite}}{A_{fan}} \tag{5.3.2}$$

Where A_{hilite} is the cross sectional area of the inlet highlight and A_{fan} is the cross section area of the fan face. Q_n is specified at every node point on the inlet plane, assuming a constant distribution over the fan face. Q_n is also given by the expression :

$$Q_n = \rho u n_x + \rho v n_y + \rho w n_z \tag{5.3.3}$$

Where n_x, n_y and n_z are the components of the unit normal to the inlet face. More information is therefore required from the inside of the domain in order to define each variable. The fan face is defined in the computations by the index $i = 1$. The next grid plane in the domain is the $i = 2$ plane where velocity ratios can be calculated as :

$$r_{vu} = \frac{v_2}{u_2} \tag{5.3.4}$$

and

$$r_{wu} = \frac{w_2}{u_2} \tag{5.3.5}$$

These ratios are extrapolated to the inlet plane where Q_n can therefore be rewritten as :

$$Q_n = \rho_1 u_1.(n_x + r_{vu} n_y + r_{wu} n_z) \tag{5.3.6}$$

$\rho u, \rho v$ and ρw can therefore be calculated at the inlet face as :

$$\rho_1 u_1 = \frac{Q_n}{(n_x + r_{vu} n_y + r_{wu} n_z)} \tag{5.3.7}$$

$$\rho_1 v_1 = \rho_1 u_1.r_{vu} \tag{5.3.8}$$

$$\rho_1 w_1 = \rho_1 u_1.r_{wu} \tag{5.3.9}$$

The value of the density ρ is also extrapolated from the inside of the domain :

$$\rho_1 = \rho_2 \tag{5.3.10}$$

The pressure at the inlet plane is calculated assuming that the total enthalpy H is constant and equal to its upstream value H_∞ :

$$P_1 = \rho_1.\frac{\gamma - 1}{\gamma}.(H_\infty - \frac{q_1^2}{2}) \tag{5.3.11}$$

where $q_1 = \sqrt{u_1^2 + v_1^2 + w_1^2}$ is the local velocity at the inlet plane. Finally, the last variable ρE is calculated with the expression :

$$\rho_1 E_1 = \frac{p_1}{\gamma - 1} + \frac{\rho_1 q_1^2}{2} \tag{5.3.12}$$

This procedure was tested and led to stable calculations of the inlet flow field .

5.3.4 Singular Line on the Axis of Symmetry

The axisymmetric grid generated for the nacelle inlet has a singular line on the axis of symmetry (**j = JL**) where all the cell faces degenerate into a single line. The area of the cell faces on the axis is zero. It was therefore necessary to by-pass the calculation of projected cell areas and normals on these faces which could have produced divisions by zero and ensuing overflows of the program. Two procedures were tested for evaluating the flow on the axis of symmetry :

- Initially, the values of the flow variables on this axis (**j = JL, i = 1** to **(NX/2)+1**) were taken to be equal to the average of the values on the surrounding node points at **j = JL-1**. The best solution was obtained when taking values interpolated three-dimensionally from those of the **j = JL** node points rather than by doing interpolations in **i = constant** grid planes.
- A more exact treatment was then implemented by which the flow variables on the axis of symmetry are computed by discretizing the equation there. This required modifications of loops in the subroutine **EULER** performing the Runge-Kutta time integration and further modifications to the calculations of the fluxes in subroutines **EFLUX**, **DFLUX** and **DFLUXC**. On the axis of symmetry, **j = JL**, the node points **k = 1** to **KL** are collapsed onto the same location in every **i = constant** grid plane. The program however calculates flow variables at each of these node points independently. A modification was made to replace the variables at these points sharing the same physical location in space by their average. This procedure ensured the uniqueness of the solution on the axis of symmetry throughout the computations. The calculation of the second and fourth order derivatives used for the artificial viscosity fluxes was done on this axis using the symmetry property of the grid.

5.3.5 Domain Inflow Boundary Conditions

In the isolated wing topology, inflow boundary conditions are imposed on the entire $\mathbf{j} = \mathbf{JL}$ grid plane. With the inlet grid topology, the inflow boundary on the $\mathbf{j} = \mathbf{JL}$ grid planes extends only from $\mathbf{i} = \mathbf{NX/2)+1}$ to $\mathbf{i} = \mathbf{IL}$. A modification was made in subroutine **BCFAR** to reflect this difference.

5.3.6 Testing of Inlet Model

The inlet models were tested by running the code for 100 iterations on the grid described earlier, using three levels of multi-grid cycles. The first test case was run at Mach 0.75 and $\alpha = 0^\circ$ with a mass flow ratio MFR = 0.75, using a full discretization of the equations on the nacelle axis. Figure 10 shows the pressure distribution calculated in the plane of symmetry which can be seen to be completely symmetrical with respect to the axis of the nacelle. Flow stagnation regions form at the leading edge of the nacelle lip with subsequent acceleration as the flow negotiates the entrance to the fan. The flow is then decelerated in the diffusion region ending at the fan inlet plane. A solution was then calculated for the same case, but with the variables on the axis of symmetry obtained as averages of the values at the surrounding node points. The solution was almost identical to that of figure 10, indicating that the averaging process was fairly accurate in this case. There was less than 1field with the two approaches. However, the accuracy of the results with an averaging procedure depends on the fineness of the grid and on the extent of the flow gradients present in the field. It was therefore assumed that they could be less accurate in some other cases and the rest of the calculations for the inlet was performed using the full discretization on the singular line. Calculations were then made, for a Mach number

0.65 and an angle of incidence $\alpha = 7^\circ$, with a mass flow ratio MFR = 0.85. As shown on figure 4.7, the flow is asymmetrical with respect to the nacelle axis. The higher mass flow ratio produces a higher acceleration of the flow in the intake and higher throat Mach numbers. Additional calculations were made to visualize the effect of the mass flow ratio on the diameter of the intake stream tube and on the stability of the calculations. A low mass flow ratio MFR = 0.25 was used on the results shown in figure 11, at a Mach number M = 0.6 and $\alpha = 0^\circ$. The large diffusion that takes place in the stream tube is clearly visible on this figure. On the other hand, a higher mass flow ratio MFR = 2.0, at Mach M = 0.3 and $\alpha = 0^\circ$ produces a noticeable contraction of the streamtube in front of the nacelle intake. This is shown in figure 12. The code was found to calculate well the inviscid flow field around a nacelle intake for various Mach numbers, angles of incidence, and mass flow ratios. It was noticed that the blockage due to the specification of very small mass flow ratios (smaller than 0.25) affected the convergence characteristics of the computations, particularly at the beginning of the calculations. This was thought to be partly due to the initial conditions used in the code (uniform flow everywhere, with the specified upstream values). The specification of initial velocities in the intake duct more compatible with the inlet mass flow ratio improve the impulsively started calculations.

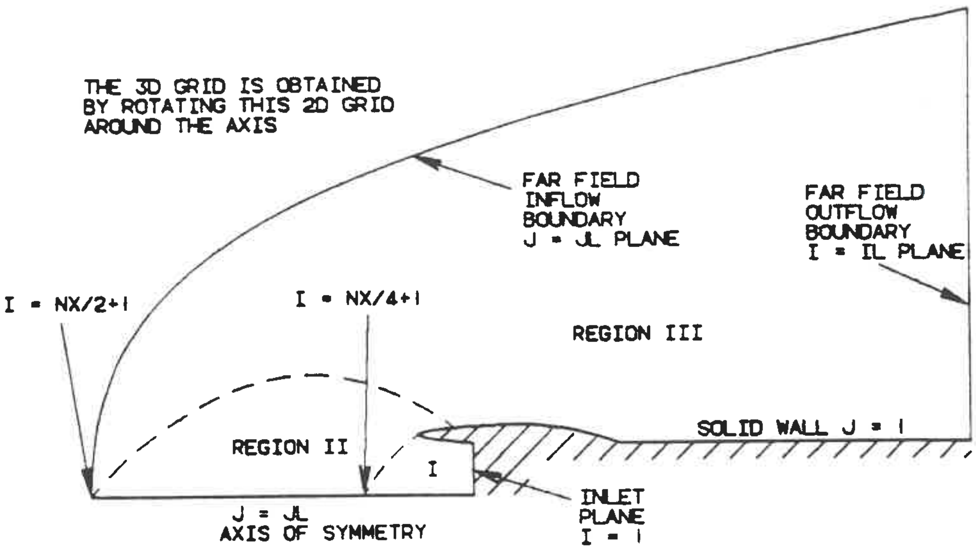


Figure 6 : Topology of the grid used to validate nacelle inlet conditions.

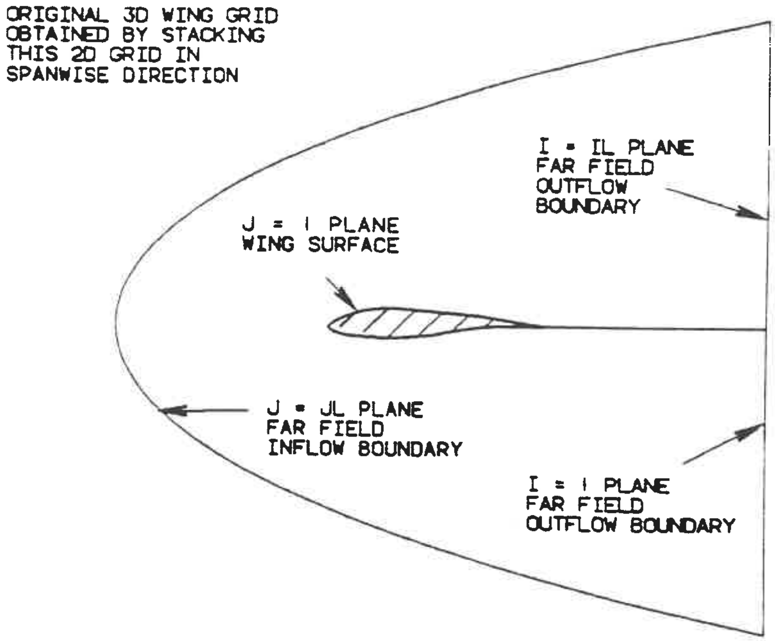


Figure 7 : Topology of the C-H grid used for isolated wings.

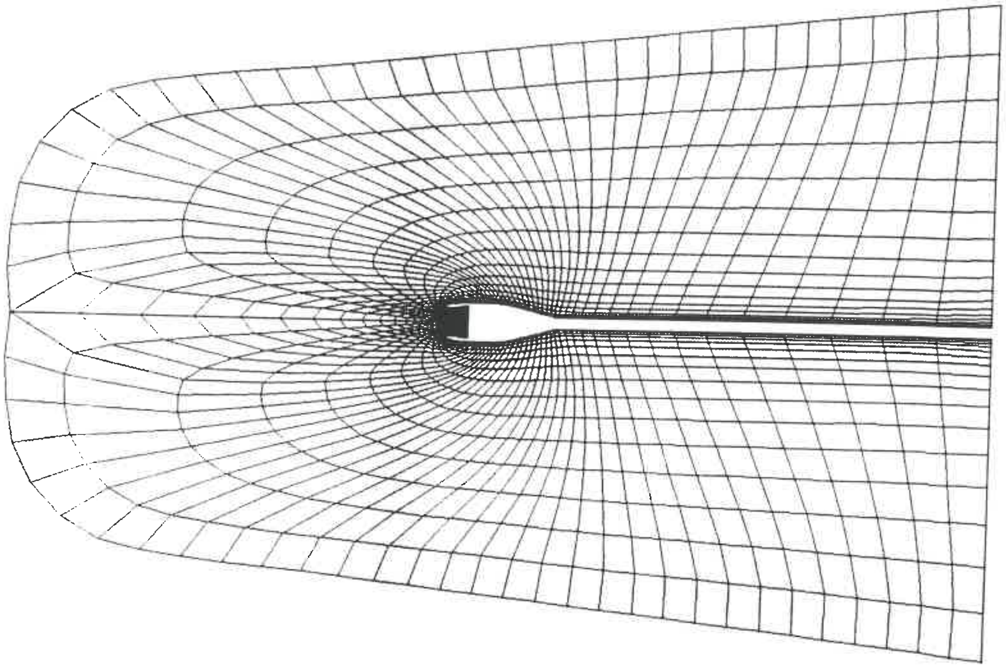


Figure 8 : Overall view of the grid generated for nacelle inlet calculations.

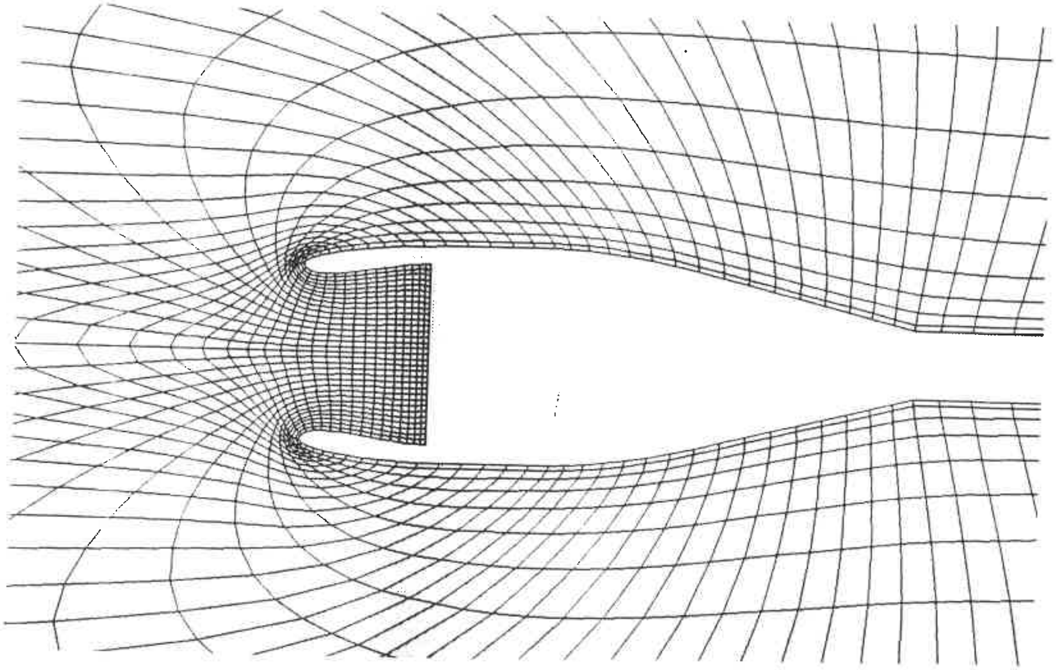


Figure 9 : Close view of the grid generated for nacelle inlet calculations.

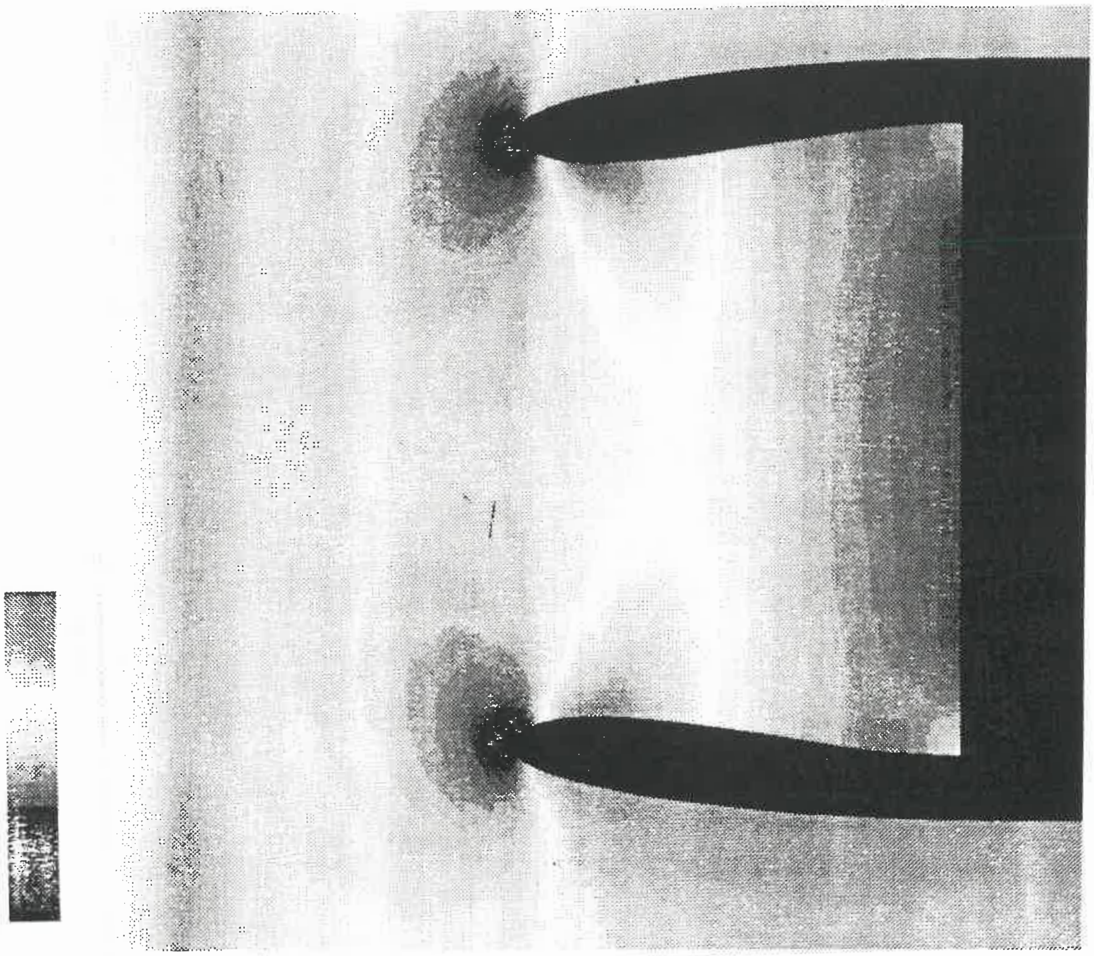


Figure 10 : Pressure distribution in the plane of symmetry of the nacelle inlet at M = 0.75, $\alpha = 0^\circ$ and $MFR = 0.75$
(Full discretization of the equations on the axis of symmetry.

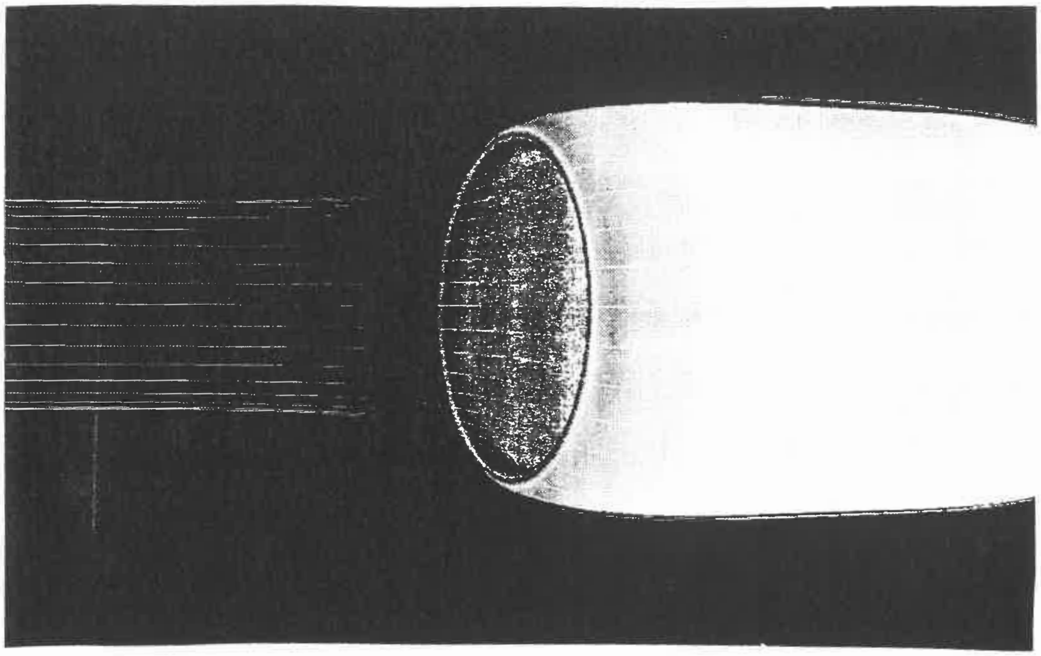


Figure 11 : Streamtube calculated in front of the nacelle inlet at $M = 0.60$, $\alpha = 0^\circ$ and $MFR = 0.25$.

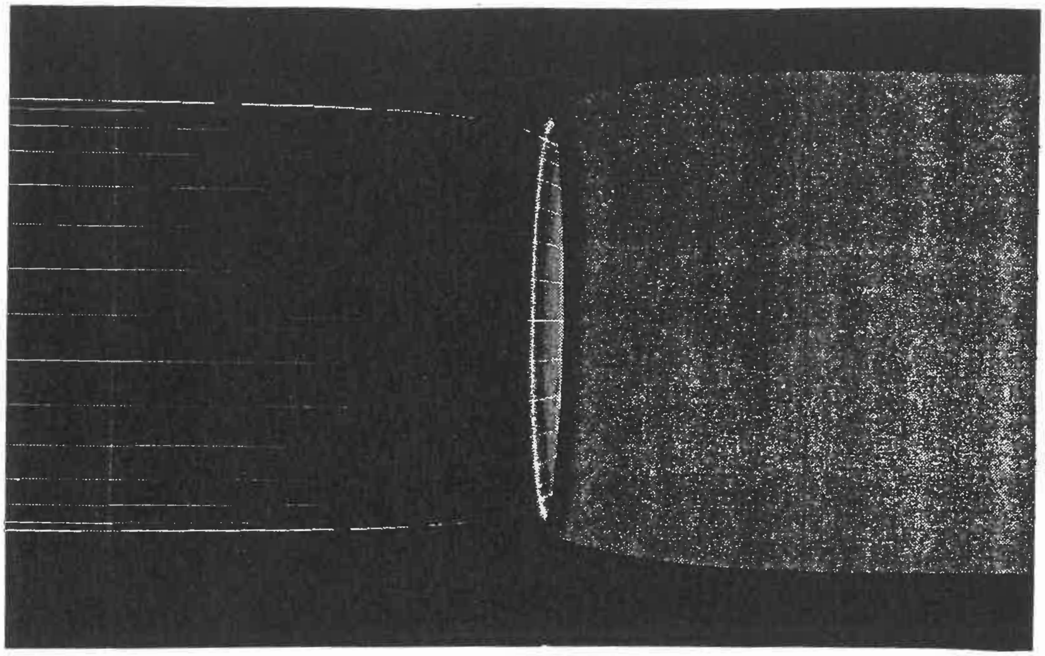


Figure 12 : Streamtube calculated in front of the nacelle inlet at $M = 0.30$, $\alpha = 0^\circ$ and $MFR = 2.0$.

5.4 ENGINE EXHAUST BOUNDARY CONDITIONS

5.4.1 Complete Nacelle Grid

The grid geometry defined for the complete nacelle is outlined on figure 13. The constraint of a single-block grid prevented the modelling of the flow inside the nozzle. The grid would have had to negotiate a very sharp turn at the exhaust plane. The engine exhaust boundary conditions were therefore tested using conditions specified on the back face of the nacelle. The characteristics of the grid on the inlet side are the same as those described previously. In the complete nacelle grid, the **j = JL** plane closes behind the nacelle and is no longer solely an inflow boundary. The aft part of the grid plane has flow leaving the domain. The **i = 1** grid plane is still the inlet plane but the **i = IL** grid plane is now collapsed onto the axis of symmetry behind the nacelle. The exhaust plane is the aft part of the **j = 1** grid plane, between **i = 5/6)*NX+2** and **i = IL**. A grid was generated with (96×32×24) node points. The grid is shown in figure 14, with a view closer to the nacelle shown in figure 15. The program for complete nacelle flow fields was produced as a further development of the nacelle inlet code. The modifications required to perform the calculations for the new topology included :

- The modification of the far-field boundary conditions.
- The treatment of the singular line downstream of the nacelle.
- The implementation of the engine exhaust boundary conditions.

5.4.2 Far Field Boundary Conditions

The far-field boundary in the new grid topology is the part of the **j = JL** grid plane extending from **i = NX/2)+1** to **i = IL**. The flow is no longer necessarily an inflow into the domain at this boundary. The subroutine **BCFAR** was modified in order to calculate the flow normal to the boundary and apply inflow or outflow boundary conditions according to the direction of the flow.

5.4.3 Singular Line Downstream of the Nacelle

In the new grid topology, the $\mathbf{i} = \mathbf{IL}$ grid plane is no longer an outflow boundary and is collapsed onto a singular line on the nacelle axis of symmetry. Modifications were made to subroutine **BCFAR** not to impose far-field boundary conditions on these grid points and a singular line treatment similar to the one discussed for the inlet was applied to these node points. Since the main purpose of the exercise was the validation of the engine exhaust boundary conditions, a simple averaging procedure was used to calculate the flow variables on the $\mathbf{i} = \mathbf{IL}$ singular line. This procedure was shown in the previous section to be fairly accurate for the grids used here.

5.4.4 Engine Exhaust Boundary Conditions

The modelling of an engine exhaust plume with an Euler code is an approximation, since it does not include any of the viscous and turbulence effects which are important in real flow. However, the model is judged to be more accurate than those used in potential models where the jet discontinuity is modelled using wake planes. The model is also accurate insofar as it produces a good description of the pressure distribution that develops on all solid surfaces adjacent to the exhaust plume. The Euler calculations can also give a description of the wave structure inside a supersonic plume but their are not meant to be a completely accurate model of the exhaust. The engine exhaust plane is a boundary with fluid flowing into the computational domain. The program was coded to accept both subsonic and supersonic exhaust flow conditions. If the exhaust is subsonic, four parameters must be specified and the fifth calculated from conditions inside the flow domain. If the exhaust is supersonic, all five unknowns of the three dimensional Euler equations must be specified at the exhaust boundary plane. A new subroutine **BCNOZLE** was written to apply this boundary condition. The inputs to **BCNOZLE** are :

- $\mathbf{M_{jet}}$: Mach number of the exhausting jet.
- **TPR** : Exhaust plane total pressure ratio.
- **TTR** : Exhaust plane total temperature ratio.
- δ : Swirl angle at the exhaust plane.

The ratios are with respect to the total normalized fluid characteristics in the far field upstream :

$$(TTR) = \frac{T_{0\ jet}}{T_{0\ \infty}} \tag{5.4.1}$$

$$(TPR) = \frac{P_{0\ jet}}{P_{0\ \infty}} \tag{5.4.2}$$

In the absence of a proper model of the inside of the nozzle, the jet at the exhaust plane was assumed to be parallel to the axis of the engine at the center of the nozzle and parallel to the edge of the engine cowling on the fringes of the exhaust plane. A local radial flow angle, θ, interpolated between the directions at the center and on the circumference, was used to define locally the angle of the jet at other points in the exhaust plane. Since the present calculations were carried out in a domain defined with a plane of symmetry, the swirl angle δ at the engine exhaust was specified as zero for all cases discussed here. If the exhaust is supersonic, all the flow quantities are specified at the exhaust plane. These values are calculated from the total pressure, total temperature and the exhaust Mach number which must also be specified :

5.4.4.1 Supersonic Exhaust Boundary Conditions

When the value of M_{jet} is greater or equal to 1.0, the value of the five conservative variables used in the program $(\rho, \rho u, \rho v, \rho w, \rho E)$ must be specified at the node points on the exhaust plane. The program calculates these parameters from the information given as input to the subroutine **BCNOZLE**. The local radial flow angle at every node (i,j,k) on the exhaust plane is calculated first, using an interpolation between the direction θ_{cowl} at the radius of the exhaust and that of the nacelle axis at the center :

$$\theta = \theta_{cowl} \frac{(y_{i,j,k} + z_{i,j,k})}{R_{exhaust}} \tag{5.4.3}$$

The total temperature and total pressure at the exhaust plane are then calculated from the input ratios TTR and TPR and the normalized upstream value $T_{0\ \infty}$ used in the code :

$$T_{0\ jet} = (TTR).T_{0\ \infty} \tag{5.4.4}$$

$$P_{0\ jet} = (TPR).P_{0\ \infty} \tag{5.4.5}$$

The non-dimensional jet velocity is calculated using the exhaust Mach number and total temperature :

$$q_{jet}^2 = \frac{\gamma M_{jet}^2 T_{0\ jet}}{1 + \frac{\gamma - 1}{2} M_{jet}^2} \tag{5.4.6}$$

The three components of the jet velocity are then calculated using the radial jet flow angle, the swirl angle and the circumferential location of the node point on the exhaust plane :

$$u_{jet} = q.cos\theta \tag{5.4.7}$$

$$U_r = -q.sin\theta \tag{5.4.8}$$

$$U_t = U_r.tan\delta \tag{5.4.9}$$

$$v_{jet} = U_r.cos\phi + U_t.sin\phi \tag{5.4.10}$$

$$w_{jet} = U_r.sin\phi + U_t.cos\phi \tag{5.4.11}$$

Where :

$$\phi = \pi.\frac{(K-1)}{NZ} \tag{5.4.12}$$

NZ is the maximum number of computational cells around the axis, from the top to the bottom of the axis of symmetry (**NZ** = **KL-1**). The pressure at the exhaust plane is calculated as :

$$p_{jet} = \frac{P_{0\ jet}}{[1 + \frac{\gamma-1}{2} M_{jet}^2]^{\frac{\gamma}{\gamma-1}}} \tag{5.4.13}$$

The density is obtained as :

$$\rho_{jet} = \frac{P_{0\ jet}}{T_{0\ jet}} \frac{1}{[1 + \frac{\gamma-1}{2} M_{jet}^2]^{\frac{1}{\gamma-1}}} \tag{5.4.14}$$

The variable containing the total energy is calculated as :

$$\rho E_{jet} = \frac{p_{jet}}{\gamma - 1} + \rho_{jet} \frac{q_{jet}^2}{2} \tag{5.4.15}$$

5.4.4.2 Subsonic Exhaust Boundary Conditions

If the exhaust is subsonic, only four conditions can be specified at the exhaust plane. The fifth parameter must be calculated using information from inside the flow domain. Three different formulations of these boundary conditions were investigated :

- The first formulation was used by Samant and Yu of NASA (reference [103]) to model propfan flows. This formulation was also used by Amendola, Tognacini, Boerstoel and Kassies to model propeller slipstream flows (reference [84]) This formulation will be referred to in the text as the *"NASA" formulation.*

- A second formulation tested was inspired by a paper from Hirose, Asai, Ikawa and Kawamura (reference [80]) of the Japan NAL (National Aeronautical laboratory). This formulation is designated as the *"Japan NAL" formulation* in the text.
- Finally, a third formulation was devised at Canadair, to be consistent with the boundary conditions used on the outer boundaries in the code. This formulation will be referred to as the *"Canadair" formulation* in the text.

a - *"NASA" Formulation*

This boundary condition was originally used for calculating propeller slipstreams using an actuator disk model. The angles θ and ϕ and the total pressure and total temperature at the exhaust, P_0 and T_0 are evaluated using the relations described earlier. The modulus q of the local flow velocity is extrapolated from the inside of the flow field and the velocity components evaluated using the relations given earlier. In the case of a subsonic exhaust, the value of M_{jet} given in input to the program is not used for any other purpose than to indicate that the boundary is subsonic. In this case, the Mach number at the exhaust plane is then evaluated using the expression :

$$M^2 = \frac{q^2}{(\gamma T_{0\ jet}^2 - (\gamma - 1)\frac{q^2}{2})} \tag{5.4.16}$$

Finally, the values of the local pressure, density and total Energy are calculated using the expressions :

$$p_{jet} = \frac{P_{0\ jet}}{[1 + \frac{\gamma-1}{2} M_{jet}^2]^{\frac{\gamma}{\gamma-1}}} \tag{5.4.17}$$

$$\rho_{jet} = \frac{P_{0\ jet}}{T_{0\ jet}} \frac{1}{[1 + \frac{\gamma-1}{2} M_{jet}^2]^{\frac{1}{\gamma-1}}} \tag{5.4.18}$$

$$\rho E_{jet} = \frac{p_{jet}}{\gamma - 1} + \rho_{jet}\frac{q_{jet}^2}{2} \qquad (5.4.19)$$

This formulation of the boundary condition works well for propeller flows in which the propeller is modelled using an actuator disk. In the models using this formulation, the mass flow is imposed on the upstream face of the disk. This mass flow is the same coming out on the downstream side of the disk. This is equivalent to setting an additional constraint for the flow. In the present model of the jet engine, there is no direct correlation in the program between the conditions at the inlet face and the conditions at the exhaust plane. Without this additional constraint, this formulation of the boundary conditions, when tested, did not lead to stable and converged results.

b - *"JAPAN NAL" Formulation.*

This formulation is very similar to the "NASA" formulation except that the mass flow at the exhaust plane is now explicitly imposed. Another difference is that the value extrapolated from the inside of the computational domain is the density ρ instead of the velocity modulus q. Once again, the value of M_{jet} given in input to indicate that the exhaust is subsonic but is not used to set the values of the variables at the exhaust plane. θ, ϕ, $T_{0\ jet}$ and $P_{0\ jet}$ are again calculated using the formulae given in the section on supersonic exhaust. The density, as already mentioned, is extrapolated from its value at the next grid plane in the domain (j = **2** grid plane) :

$$\rho_{jet} = \rho_2 \qquad (5.4.20)$$

The specified mass flow ratio at the exhaust is used to calculate ρu_{jet} :

$$\rho u_{jet} = \rho_\infty q_\infty MFR_{exhaust} \qquad (5.4.21)$$

The x-component of velocity of the jet, normal to the exhaust plane is then :

$$u_{jet} = \frac{\rho u_{jet}}{\rho_{jet}} \tag{5.4.22}$$

The tangential and radial components of the velocity are then calculated in the manner described earlier :

$$U_r = -u_{jet}.tan\theta \tag{5.4.23}$$

$$U_t = U_r.tan\delta \tag{5.4.24}$$

The y and z components of the velocity in the exhaust plane are then given by :

$$v_{jet} = U_r.cos\phi + U_t.sin\phi \tag{5.4.25}$$

$$w_{jet} = U_r.sin\phi + U_t.cos\phi \tag{5.4.26}$$

The modulus of the velocity and the Mach number are then evaluated as :

$$q^2 = (u_{jet}^2 + U_r^2 + U_t^2) \tag{5.4.27}$$

$$M^2 = \frac{q^2}{(\gamma T_{0\ jet}^2 - (\gamma - 1)\frac{q^2}{2})} \tag{5.4.28}$$

The pressure and the total energy component of the Euler flux term can finally be calculated as :

$$p_{jet} = \frac{P_{0\ jet}}{[1 + \frac{\gamma-1}{2} M_{jet}^2]^{\frac{\gamma}{\gamma-1}}} \tag{5.4.29}$$

$$\rho E_{jet} = \frac{p_{jet}}{\gamma - 1} + \rho_{jet}\frac{q_{jet}^2}{2} \tag{5.4.30}$$

The use of this formulation leads to stable calculations.

c - *" CANADAIR" Formulation.*

During the development of the code, it was felt that a formulation of the boundary conditions consistent with the one used for the other inflow boundaries in the code should be tested. This formulation, designed to absorb waves propagating in a direction normal to the boundary, uses the Riemann invariants at the boundary to convect information on the flow variables. The formulation is described here, with the subscript "*e*" indicating values extrapolated from inside the domain, the subscript "*j*" values calculated from the specified exhaust jet conditions and the subscript "*f*" the final values calculated at node points on the exhaust plane. p_e, ρ_e and q_e are the values of the pressure, density and velocity modulus calculated inside the domain, at node points adjacent to the exhaust plane. The x-component of the velocity, which is also normal to the exhaust plane, is first extrapolated from inside the domain :

$$q_{n\ e} = u_e \tag{5.4.31}$$

The speed of sound calculated for these extrapolated local conditions is :

$$C_e = \gamma \frac{p_e}{\rho e} \tag{5.4.32}$$

Similarly, the values of p_j, q_j and ρ_j corresponding to the specified exhaust jet conditions are evaluated, using the expressions given in the section on supersonic exhaust. A normal velocity component and a local speed of sound corresponding to the specified jet conditions can be defined as :

$$q_{n\ j} = u_j \tag{5.4.33}$$

$$C_j = \gamma \frac{p_j}{\rho_j} \tag{5.4.34}$$

The remaining components v_j and w_j of $q_{n\ j}$ are calculated using the local flow angle θ and the swirl angle δ as indicated previously. The Riemann invariant ER from interior values (i.e. outgoing wave) and the invariant FR from exterior or jet condition values (i.e. incoming wave) are calculated as :

$$ER = q_{n\ e} - 2\frac{C_e}{(\gamma - 1)} \tag{5.4.35}$$

$$FR = q_{n\ j} + 2\frac{C_j}{(\gamma - 1)} \tag{5.4.36}$$

At the boundary, the normal velocity is then determined by :

$$q_n = \frac{ER + FR}{2} \tag{5.4.37}$$

From this expression of q_n, the final values of the three components of velocity at the exhaust plane are calculated as :

$$u_f = u_j + (q_n - q_{n\ j}) \tag{5.4.38}$$

$$v_f = vj \tag{5.4.39}$$

$$w_f = wj \tag{5.4.40}$$

And the final value of the velocity modulus at the exhaust plane is :

$$q_f = \sqrt{u_f^2 + v_f^2 + w_f^2} \tag{5.4.41}$$

A measure of the total entropy of the exhaust flow is then calculated as :

$$S_j = \frac{\rho_j^\gamma}{p_j} \tag{5.4.42}$$

The total enthalpy of the jet is given as :

$$H_j = \frac{\gamma}{(\gamma - 1)}\frac{p_j}{\rho_j} + \frac{q_j^2}{2} \tag{5.4.43}$$

The final values of the density, pressure and of total energy are eventually calculated using the following expressions :

$$(CC) = \frac{(\gamma - 1)}{\gamma}(H_j - \frac{q_j^2}{2}) \tag{5.4.44}$$

$$\rho_f = (S_j(CC))^{\frac{1}{\gamma-1}} \tag{5.4.45}$$

$$p_f = \rho_f(CC) \tag{5.4.46}$$

$$\rho E_f = \frac{p_f}{\gamma - 1} + \rho_f \frac{q_f^2}{2} \tag{5.4.47}$$

This formulation of the exhaust boundary conditions leads to very stable computations with good convergence characteristics. It was selected as the permanent formulation of the newly developed program since it was considered to be more consistent with the formulation of the remaining boundary conditions in the code.

5.4.5 Testing of the Complete Nacelle Model

The complete nacelle model was tested by running the program on the grid previously described for 100 iterations and 3 levels of multi-grid cycles. The first result, shown in figure 16 is for an upstream Mach number 0.85, an angle of incidence $\alpha = 0^\circ$, an inlet mass flow ratio $MFR = 0.75$ and a supersonic exhaust with a Mach number $M_{jet} = 1.15$. The specified exhaust total pressure ratio is $TPR = 1.3$ and the total temperature ratio is $TTR = 2.0$. The figure shows the static pressure distribution in the plane of symmetry, which is seen to be perfectly symmetrical . The total pressure plot in figure 17 outlines the jet with its higher total pressure. There is no shock wave outside of the jet in the flow field and the total pressure is constant. Figure 18 gives a view of the velocity vectors calculated in and around the exhaust jet. The transition between the jet and the outside field appears to be smooth. A minor difficulty arose at the perimeter of the engine because of the grid configuration and the lack of modelling of the flow inside the nozzle. Originally, exhaust boundary conditions were applied at all the grid points on the exhaust plane, including the ones at the perimeter, at the junction with the engine core cowl. It was found that this was causing some unphysical expansion of the exhaust into the field at the corner since the flow could not be constrained to be tangent to the core cowl there. The scheme was changed to apply exhaust boundary conditions on the exhaust plane only to the next to last point near the perimeter of the exhaust plane. The results were improved but there was a residual acceleration of the flow at the corner which the flow is trying to turn. The only satisfactory treatment which was later implemented on multi-block models is to model the flow inside the nozzle and obtain the correct direction of the flow at the exhaust perimeter. The present model gave nevertheless good results everywhere except in this very local area. The Euler program uses enthalpy damping as a means of accelerating the convergence of the calculations. This procedure assumes that in the steady state, the total enthalpy of the fluid is constant and equal to its upstream value. This is no longer true with an engine exhaust with fluid introduced into the domain at a total temperature and total pressure different from the upstream values. In order to avoid steering the solution towards wrong results, the enthalpy damping is not used in large parts of the flow field inside and around the exhaust jet. During the calculations, it was found that the solution had difficulties in converging at the beginning for high total temperature and total pressure ratios. This was due to the initial flow field originally calculated by the program (impulsive start in uniform flow with upstream values) which was a very poor approximation in the exhaust jet area. This problem was encountered in the calculations at the engine inlet but appeared to be worse at the exhaust, due to the very different flow conditions specified at that boundary. It was possible to improve the initial convergence characteristics by substituting p_{jet} and ρ_{jet} to the upstream uniform flow values as initial flow field in the jet area. However, on the Challenger multi-block model developed later, it was found that the program did converge from an initial uniform flowfield for normal operating total pressure and temperature ratios. Unless otherwise specified, the results with subsonic exhaust flows were obtained using the "Canadair" formulation of the boundary conditions. The results shown in figure 19 are for an upstream Mach number 0.74, an angle of incidence $\alpha = 0^{\circ}$, an inlet mass flow ratio $MFR = 0.753$ and a subsonic exhaust flow with a nominal Mach number $M_{jet} = 0.915$. The specified exhaust total pressure ratio is $TPR = 1.05$ and the total temperature ratio is $TTR = 1.18$. These values were selected to be consistent with the operating conditions of a typical engine, as found in the literature (reference [80]). Figure 19 shows the static pressure distribution obtained in the plane of symmetry and figure 20 the plot of the local static temperature in the same plane. Figure 21 shows the iso-Mach lines calculated on the plane of symmetry and on the nacelle cowling. On figure 22, the pressure distributions along the top line of the nacelle and on the intake wall are shown, indicating the diffusion area upstream of the fan face. The pressure distribution on the outer cowling is similar to distributions calculated on nacelles of the same type in reference [80] with the exception of the acceleration at the end of the core cowl which is due to the treatment at the corner of the cowl and the exhaust plane mentioned previously. This acceleration disappear on multi-block models with proper grid topology and a proper calculation of the nozzle flow, as will be shown later. The last test result presented here is for an upstream Mach number M = 0.75, an angle of incidence $\alpha = 8^{\circ}$, a low mass flow ratio $MFR = 0.375$ and a supersonic exhaust with $M_{jet} = 1.15$. The total pressure ratio at the exhaust is $TPR = 1.3$ and the total temperature ratio $TTR = 2.0$. This case was interesting because it associated a high angle of incidence with a very low mass flow ratio and fairly high total temperature and total pressure ratios at the exhaust plane. After an order of magnitude of convergence, the solution started oscillating. Upon visualization on a graphics workstation, it was realized that vortices had formed on the upper surface of the nacelle. Figure 23 shows a plot of the Mach number obtained in the plane of symmetry in such vortical conditions. Superimposed to this plot are the trajectories of particles involved in the vortical flow. Figure 24 shows pressure distributions at the top line of the nacelle and on the intake surface. The combination of a high angle of incidence and of a low inlet mass flow ratio leads to high velocity peaks on the outer surface of the nacelle lip. This pattern of flow induced at the leading edge is probably responsible for the appearance of vortices. This behavior has been reported in other Euler code applications where it was thought that the use of artificial viscosity, associated with velocity peaks and entropy fluctuations, were responsible for generating this separated flow pattern. The density of the grid is probably insufficient to faithfully represent the flow in this area and may be the reason for the lack of adequate convergence of the results. This behavior, not reproduced in any of the other applications discussed in the present work, is an interesting subject for further investigation with various solvers. A more refined grid around the engine is also necessary to capture correctly the shock pattern of expansion and compression waves in the exhaust plume. On some of the test runs, these waves were noticeable near the exhaust plane where the grid is the finest but the pattern was more diffused as the size of the mesh increased downstream (figure 25). Another observation made on the results of this test case was the existence of a vorticity layer between the exhaust jet and the surrounding flow field. The presence of such a layer is also mentioned by Hirose, Asai, Ikawa and Kawamura (reference [103]). It is mostly visible on total pressure plots displaying a zone of low total pressure around the jet.

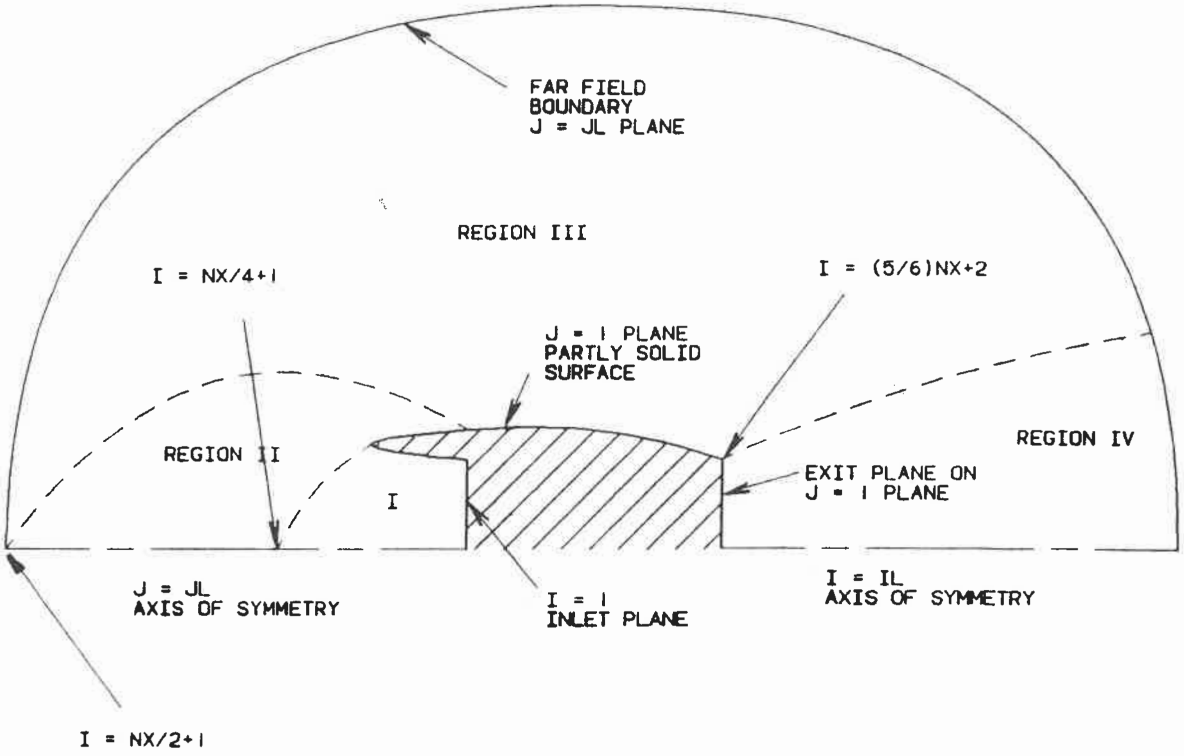


Figure 13 : Topology of the grid used for the complete nacelle calculations.

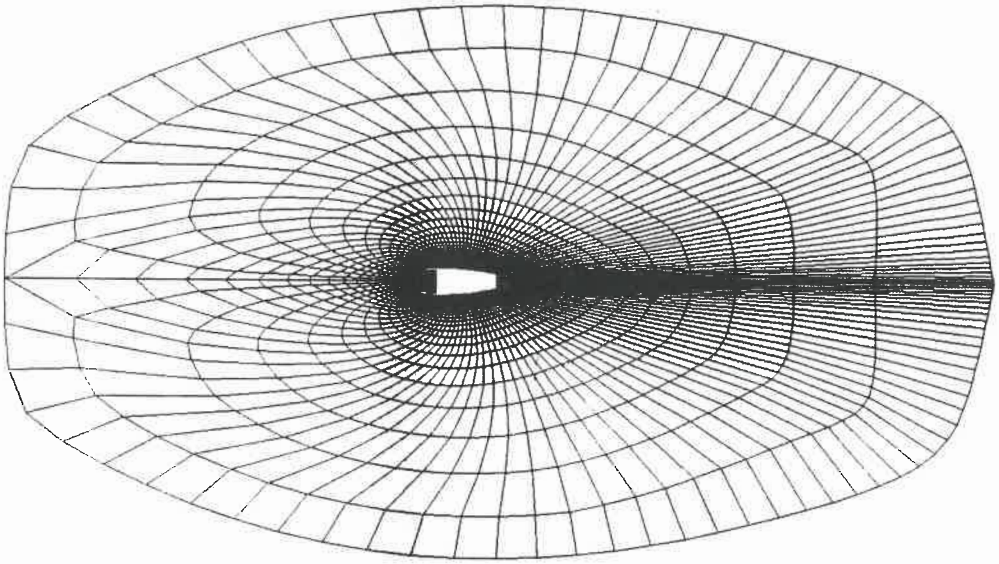


Figure 14 : Overall view of the grid generated for complete nacelle calculations.

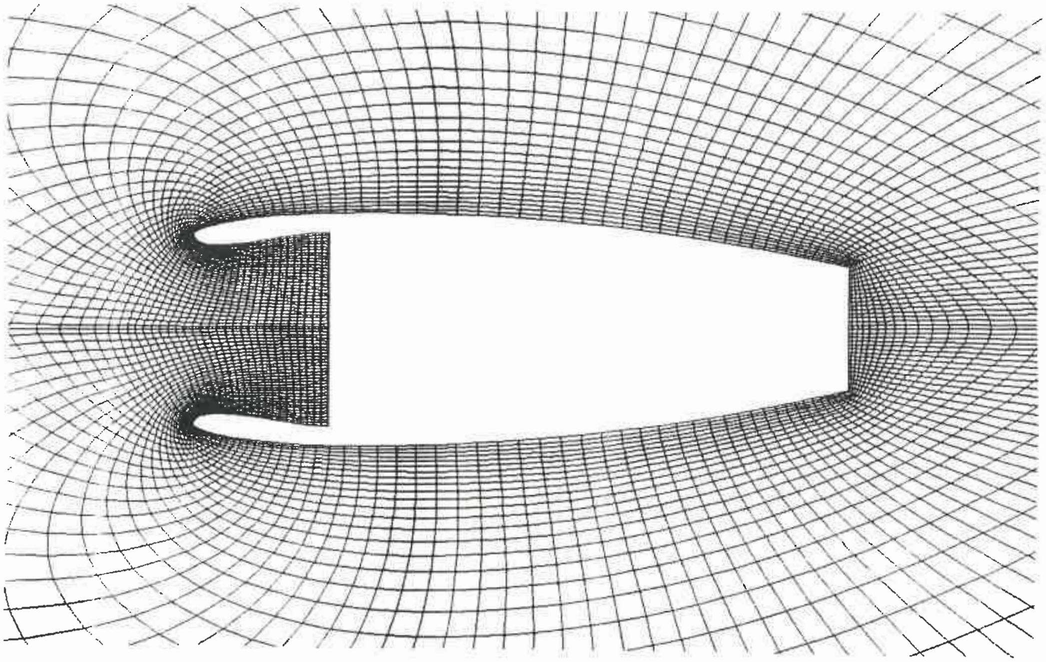


Figure 15 : Close view of the grid generated for complete nacelle calculations.

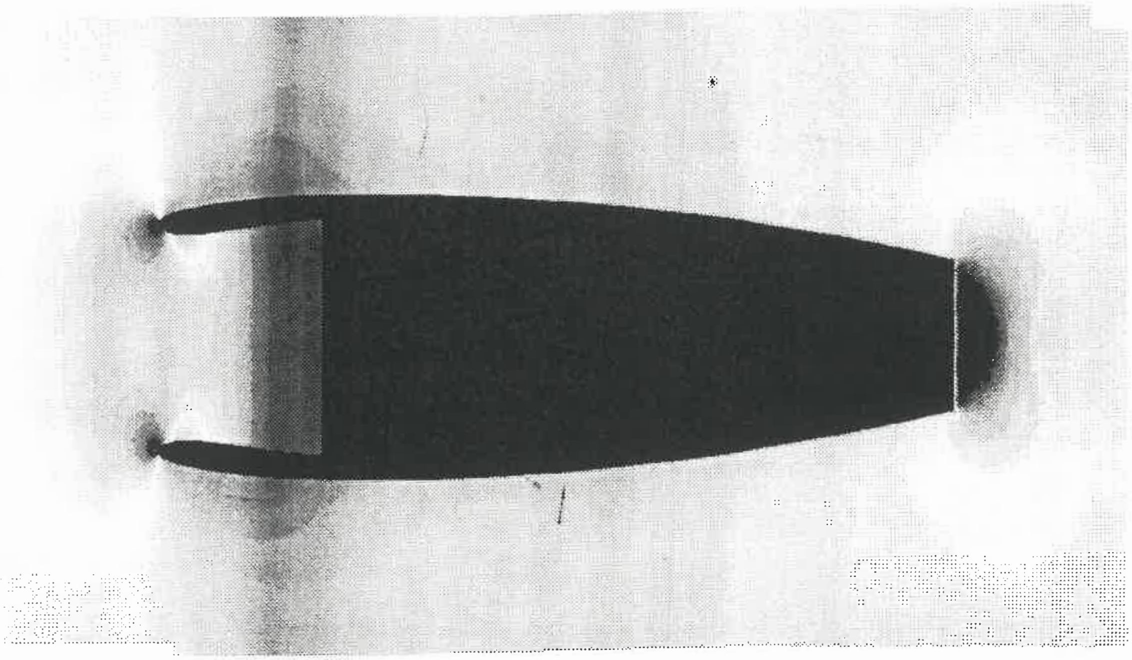


Figure 16 :

Pressure distribution in the plane of symmetry of the complete nacelle at M = 0.85, $\alpha = 0^\circ$ and $MFR = 0.75$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.3$ and $TTR = 2.0$.

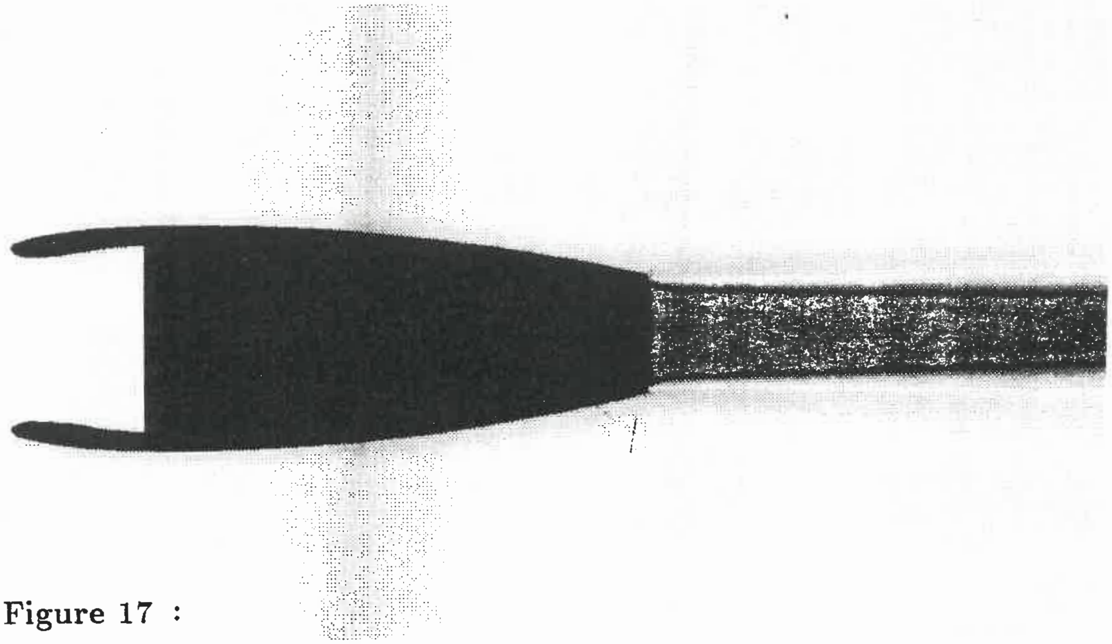


Figure 17 :

Total pressure distribution in the plane of symmetry of the complete nacelle at M = 0.85, $\alpha = 0^\circ$ and $MFR = 0.75$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.3$ and $TTR = 2.0$.

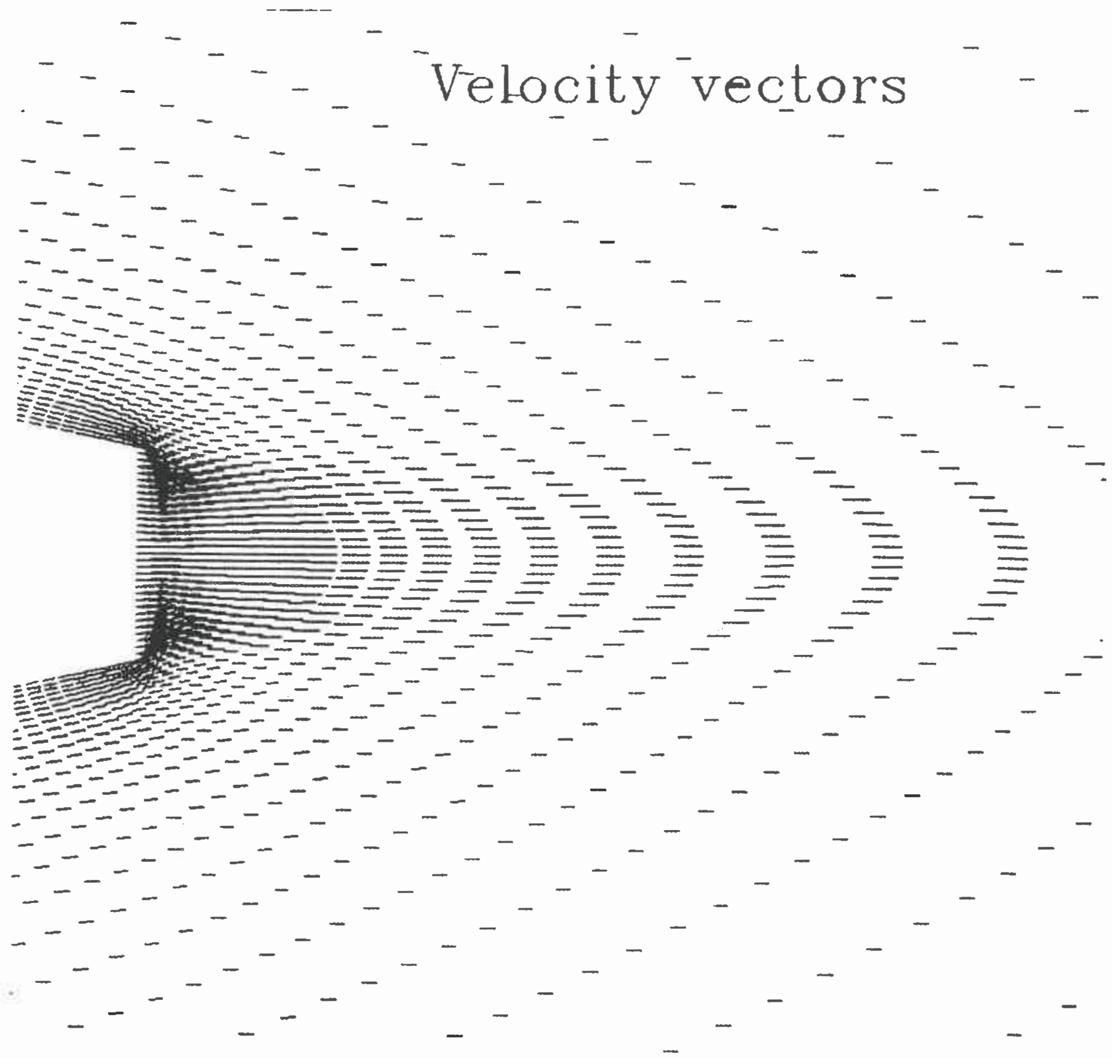


Figure 18 :

Velocity vectors calculated in the plane of symmetry of the complete nacelle at M = 0.85, $\alpha = 0^\circ$ and $MFR = 0.75$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.3$ and $TTR = 2.0$.

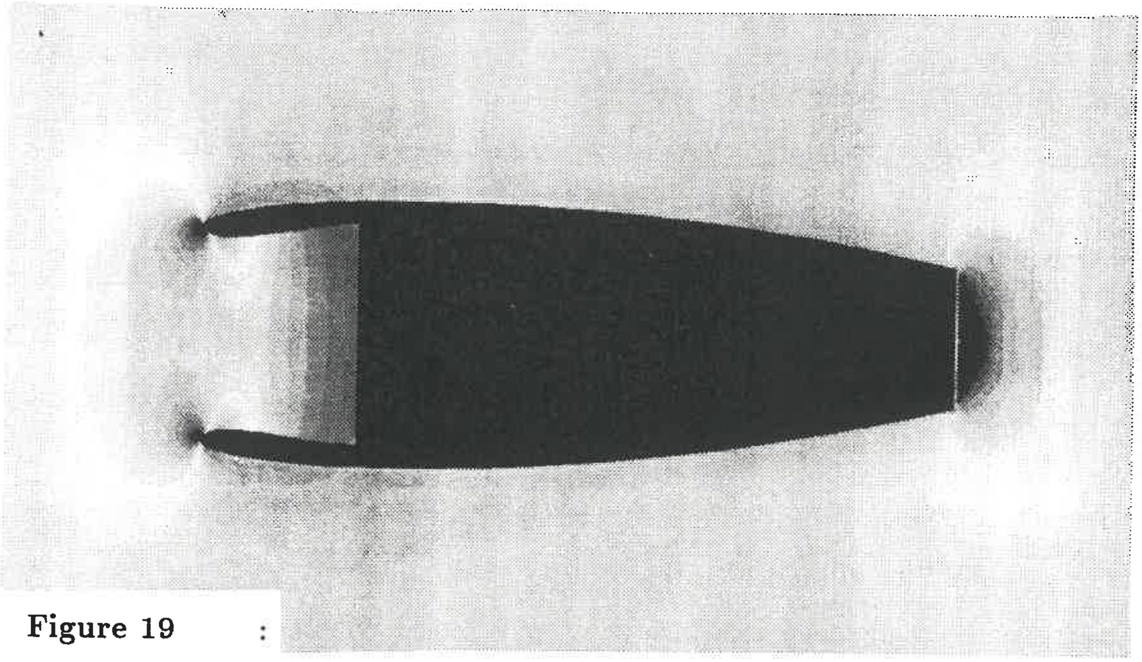


Figure 19 :

Pressure distribution in the plane of symmetry of the complete nacelle at M = 0.74, $\alpha = 0^\circ$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

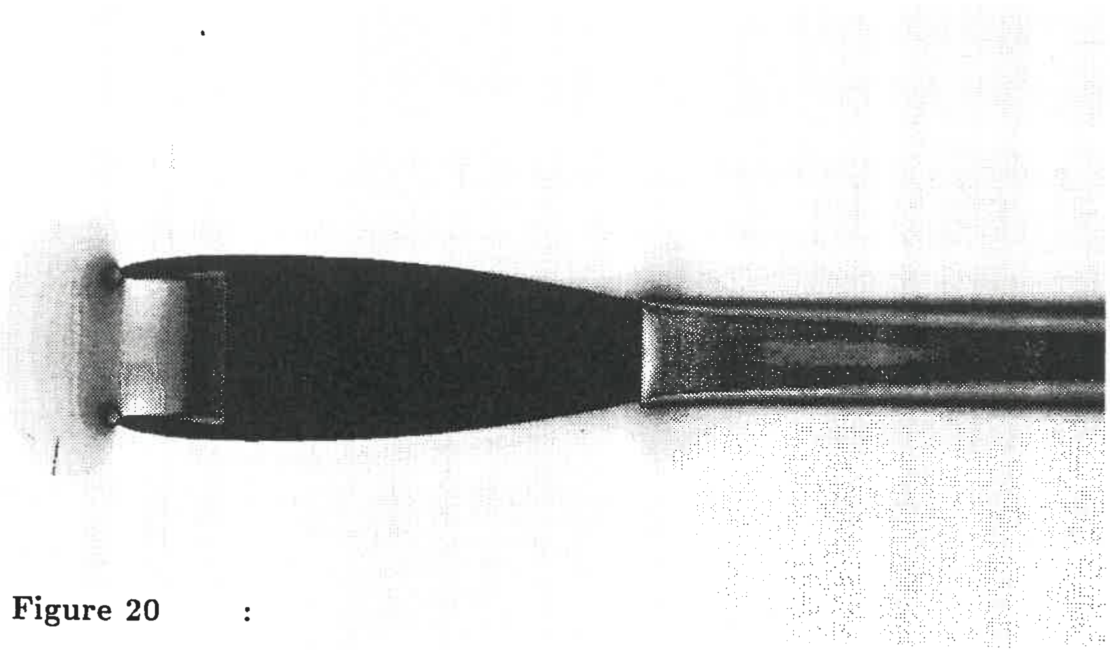


Figure 20 :

Temperature distribution in the plane of symmetry of the complete nacelle at M = 0.74, $\alpha = 0^\circ$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

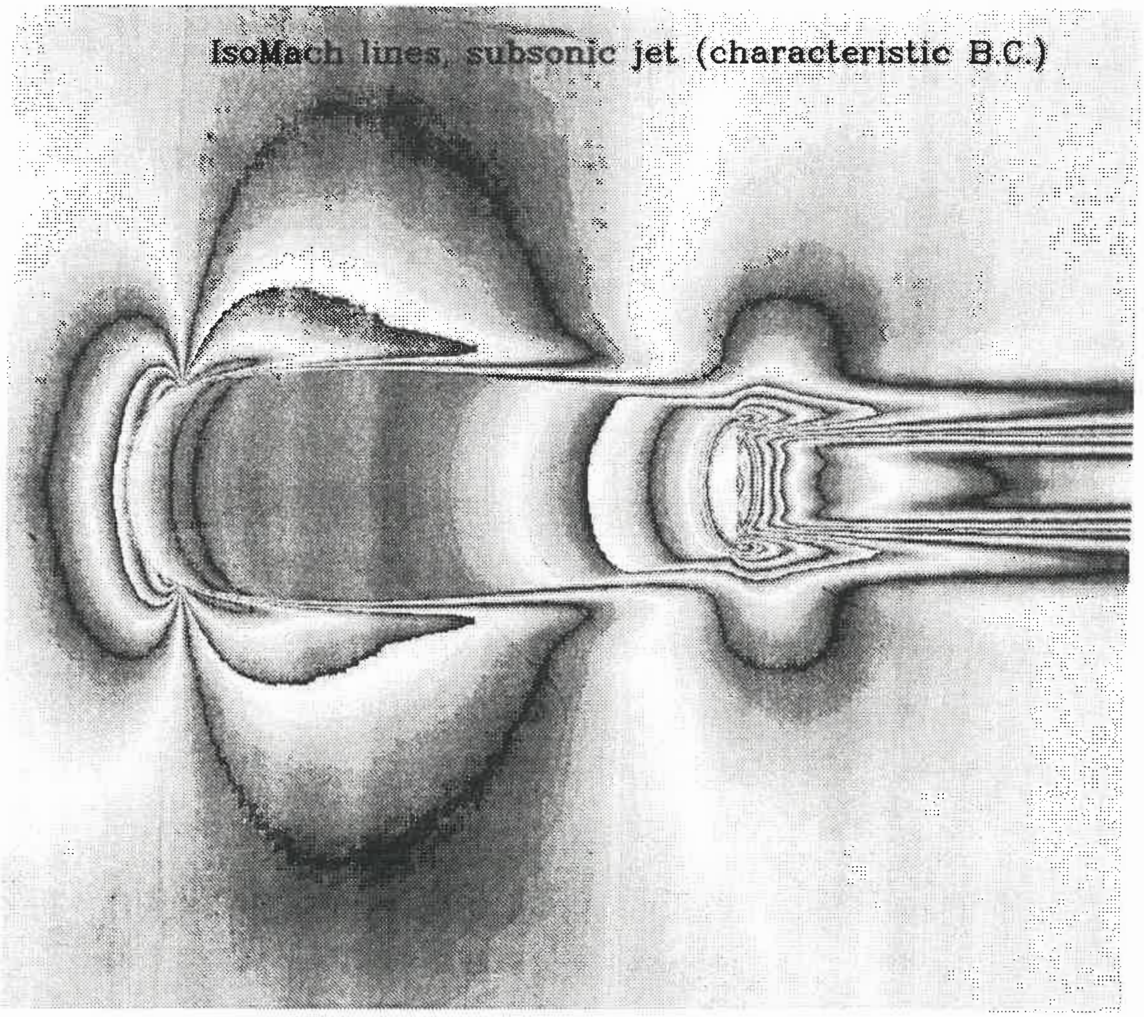


Figure 21 :

Iso-Mach lines calculated in the plane of symmetry and on the cowl of the complete nacelle at M = 0.74, $\alpha = 0^\circ$ and $MFR = 0.753$ for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

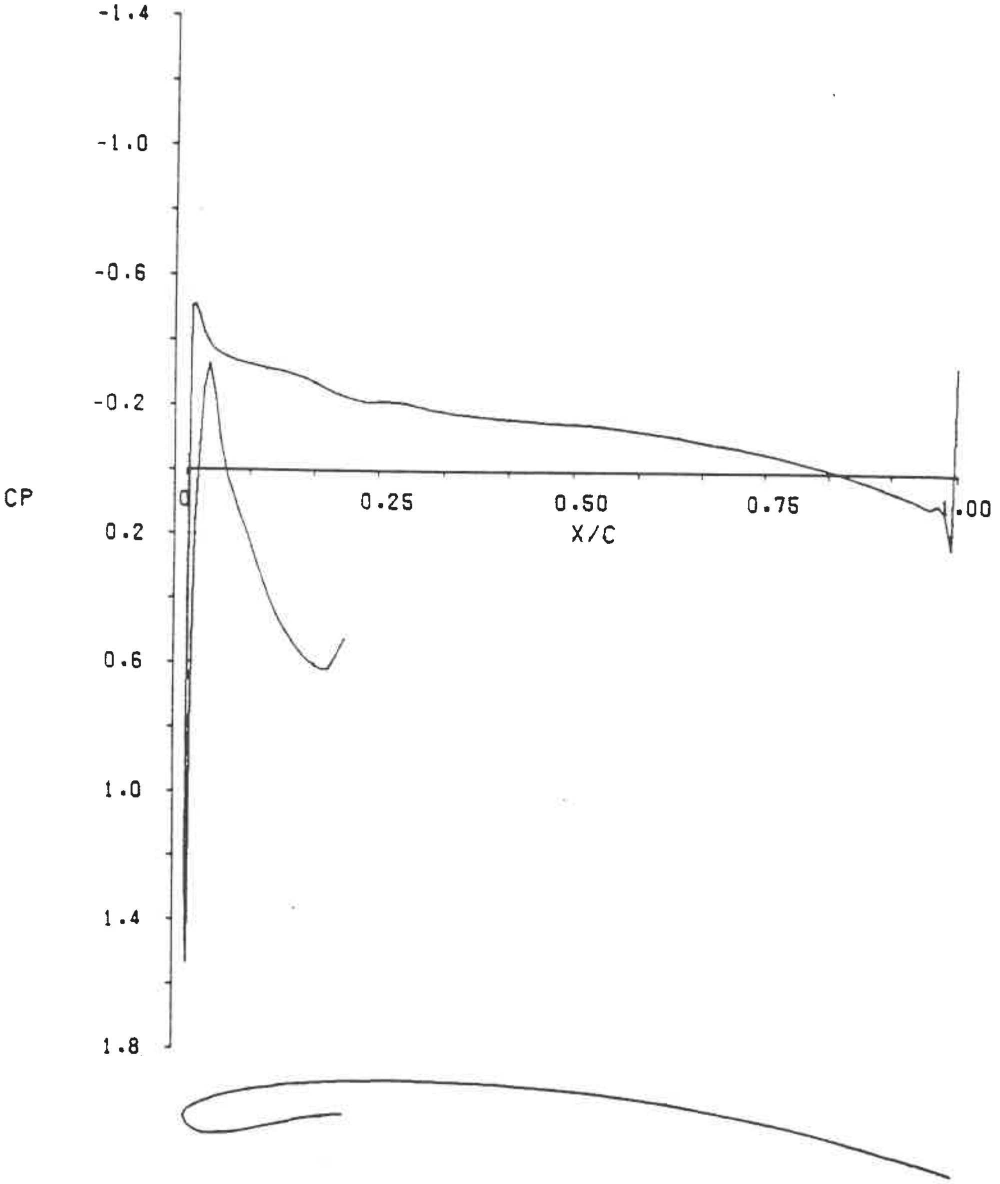


Figure 22 :

Pressure distribution calculated along the top line of the complete nacelle at M = 0.74, $\alpha = 0^\circ$ and MFR = 0.753 for a subsonic exhaust with $TPR = 1.05$ and $TTR = 1.18$.

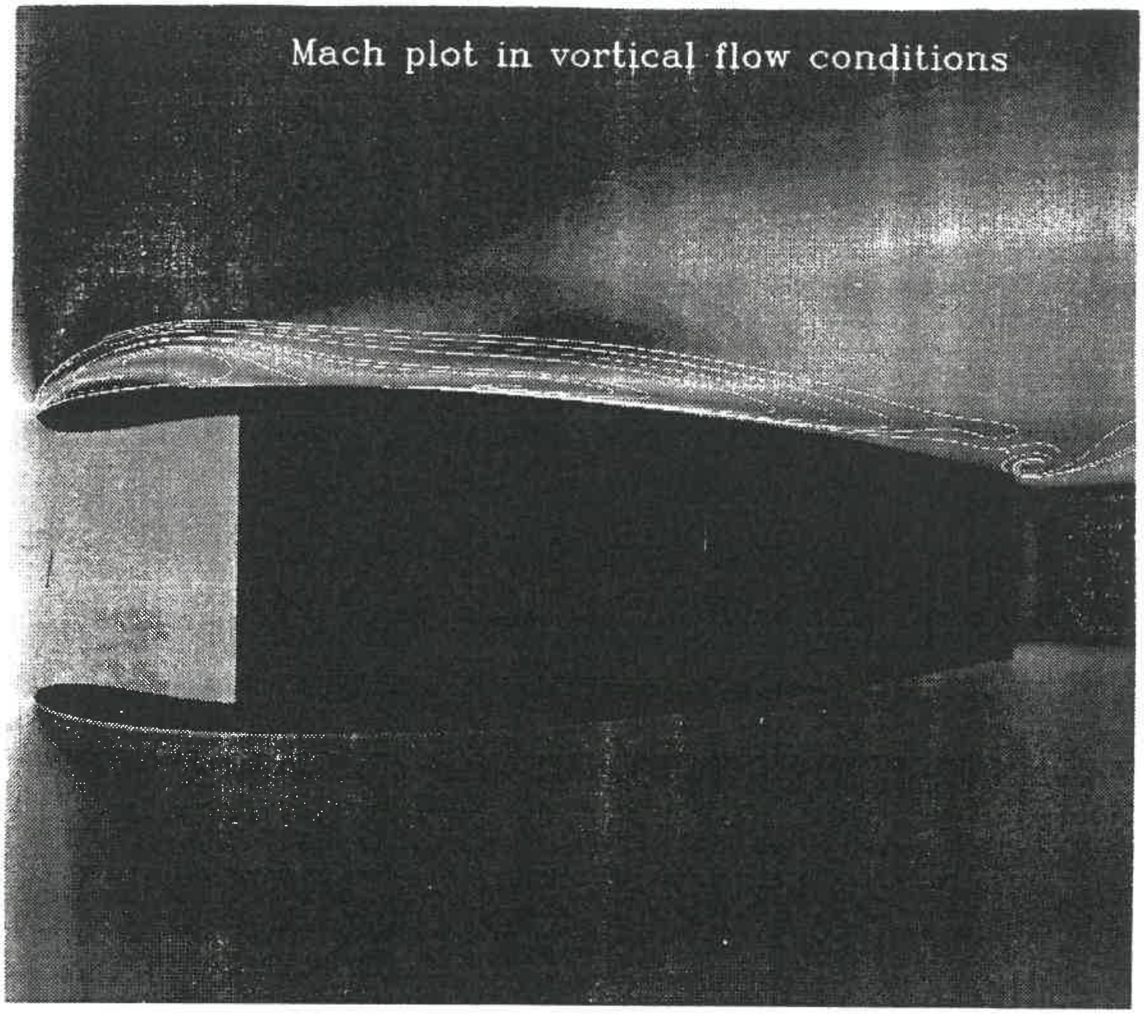


Figure 23 :

Mach plot showing vortical flow in the plane of symmetry of the complete nacelle at M = 0.75, $\alpha = 8^\circ$ and $MFR = 0.375$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.30$ and $TTR = 2.0$.

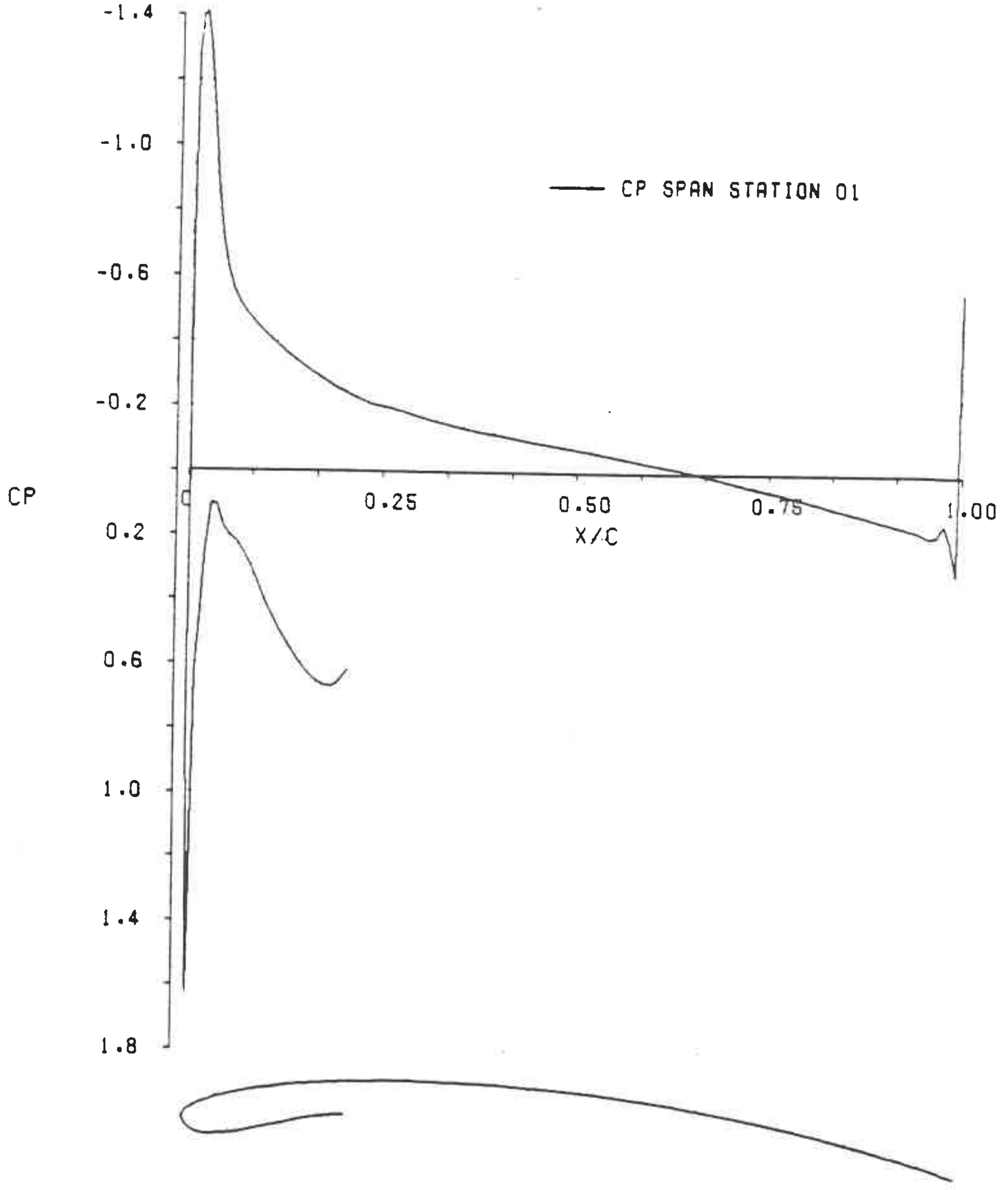


Figure 24 :

Pressure distribution calculated along the top line of the complete nacelle at M = 0.75, $\alpha = 8^\circ$ and $MFR = 0.375$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.30$ and $TTR = 2.0$.

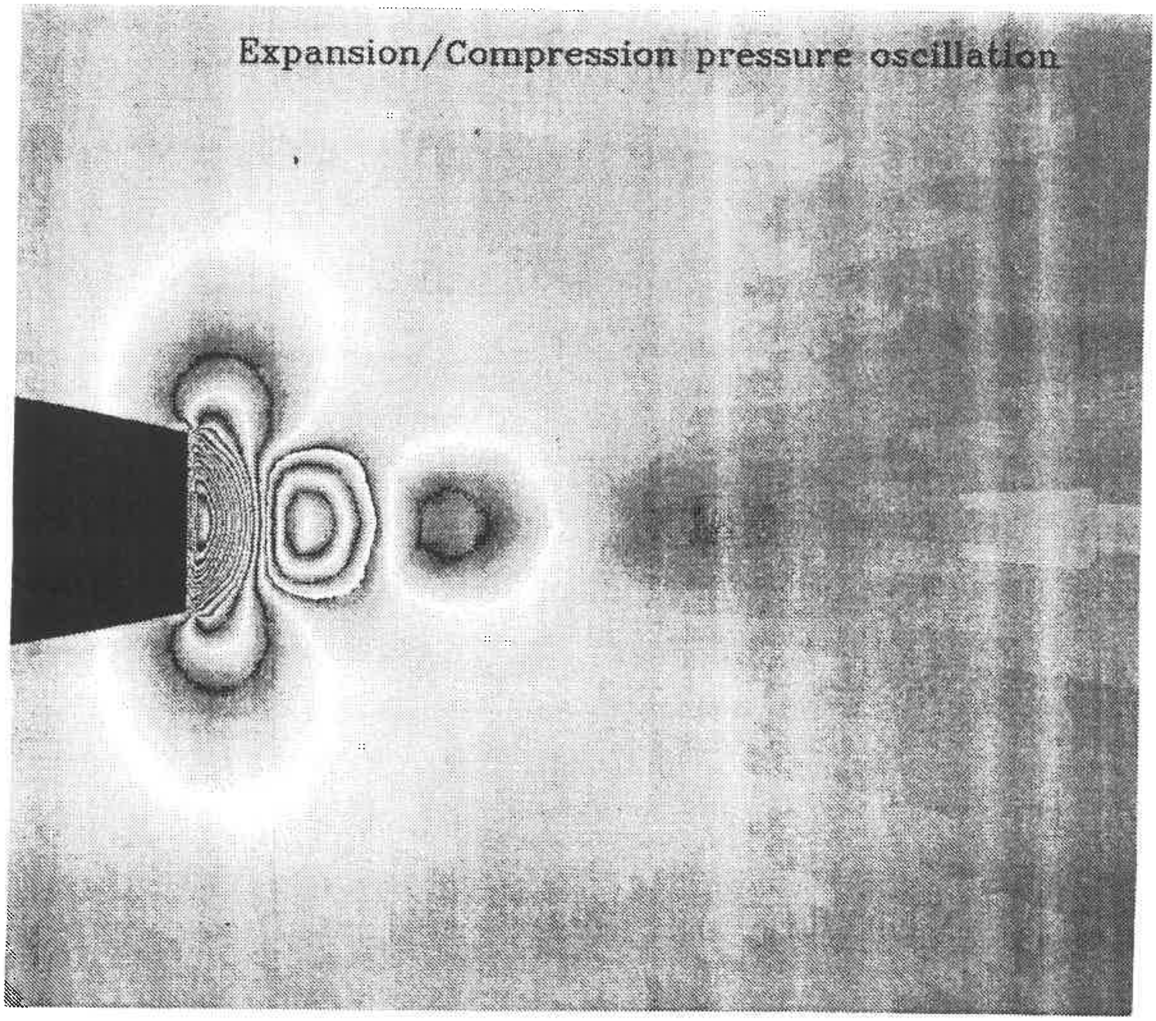


Figure 25 :

Pressure distribution in the plane of symmetry near the exhaust of the complete nacelle showing shock and expansion patterns. Solution calculated at M = 0.75, $\alpha = 8^\circ$ and $MFR = 0.375$ for a supersonic exhaust with $M_{jet} = 1.15$, $TPR = 1.30$ and $TTR = 2.0$.

5.5 INTERNAL BLOCK FACE BOUNDARY CONDITIONS

The boundary conditions at internal block faces are very important in a multi-block program since the equations are solved in each block independently. Adjacent blocks have the same number of grid points on the common face and the grid is continuous across the interface making it possible to impose a boundary condition which makes the interface virtually transparent to the flow field. This is done by carrying two layers of flow variables from grid planes of the neighbouring block known as "*halo grid planes*" (figure 26). One layer of flow variables on each side of the boundary is sufficient to calculate convective fluxes at the interface but two layers on each side are necessary to calculate dissipative fluxes. A block of grid points, augmented with the points in the "halo grid planes" is designated as an "*extended block*". In this approach, the variables at block interfaces are calculated virtually the same way as at interior grid points. Some authors use only one layer of variable from a neighbouring block, relying on a modified calculation of the dissipative fluxes at the interface [104]. This reduces the amount of additional information that needs to be carried when solving for a block. Others allow some discontinuity of the grid lines at the interface or admit neighbouring blocks with different numbers of grid points. An interpolation procedure is then required. This formulation was tested by A. Kassies and R. Tognaccini to domains with local grid refinement [105] but it requires special treatment at interfaces to limit the loss of accuracy. The formulation proposed in the present work gives a higher solution accuracy for H-H grids used on complex configurations, particularly when using a central difference method with artificial viscosity. The difficulty with this formulation is in the particular case of blocks adjacent to wings or solid surfaces, with truncated halo regions. Special treatments are then required to maintain the consistency and the stability of the discretization. More is said about the scheme in the following chapter. In subsequent testing, it was found that the multi-block algorithm implemented here converges very well even with shock waves traversing block boundaries. The code was written to handle interface orientations resulting from three different topologies, as shown in figure 27 :

- *Standard H-H topology* : in this case, the bottom face (face 3) of a block interfaces with the top face (face 4) of the block located underneath it. The orientations of the **i**, **j** and **k** indices are the same in the two blocks. The index numbers of the grid points in the halo of a block are the same as the index numbers of those grid points in the adjacent block.
- *C-H topology* : this topology is useful around wing-type components. In this case, in the wake behind the wing, the bottom face (face 3) of some blocks interfaces with the bottom face (face 3) of the blocks located underneath them. The orientations of the **i**, **j** and **k** indices are different in the interfacing blocks, with the **i** and **k** indices running opposite to each other. Both index numbers of the grid points in the halo of a block are running opposite to the index numbers of those grid points in the adjacent block. It was necessary to devise an algorithm ensuring that the halo grid points selected are at the proper physical location in space.
- *O-H topology* : this topology is useful for high alpha computations since it was demonstrated that this topology requires a fewer number of grid points to capture properly upper surface vortices (Hitzel, reference [106]). In this case also, the bottom face (face 3) of some blocks interfaces with the bottom face (face 3) of the blocks located underneath them. Again, the orientations of the **i**, **j** and **k** indices are different in the interfacing blocks, but only the **k** indices are running opposite to each other. This index numbers of the grid points in the halo of a block is running opposite to the index numbers of

those grid points in the adjacent block. The algorithm ensurcwesing that the halo grid points selected are at the proper physical location in space for this case also.

In order to reduce the number of points required to cover the space around the aircraft, an extension was written relaxing the constraint of identical numbers of grid points in adjacent blocks. A formulation was implemented allowing a block to have a number of grid points which is a multiple of the number of grid points in an adjacent face. It is thus possible to have twice or three times as many grid points in one block than in the next, as shown in figure 28. The difficulty here consists in the fact that the halo grid planes have a different number of grid points than the blocks in which they are located. It is therefore necessary to apply interpolation procedures. These interpolations must be such that they maintain conservation of the fluxes. This is better achieved when the number of cells on one face is a subset of the number of cells on the adjacent face of the opposite block. A mismatch in the number of grid points in neighbouring blocks has far reaching consequences in the building of the extended block.

5.6 SINGULAR LINES AND SINGULAR POINTS

The multi-block code accepts block faces which are collapsed to a single line or a single point, such as on an axis of symmetry, in front of an aircraft fuselage nose or behind the fuselage tail. The program prevents the addition of “halo grid planes” to collapsed block faces. The flow variables at nodes sharing a common grid point on a singular line or point are solved for independently and then replaced by their average during the calculations. Singular points may also be located on a solid surface, at the junction between several blocks. The program insures that identical boundary conditions are applied at all nodes sharing this location in all blocks.

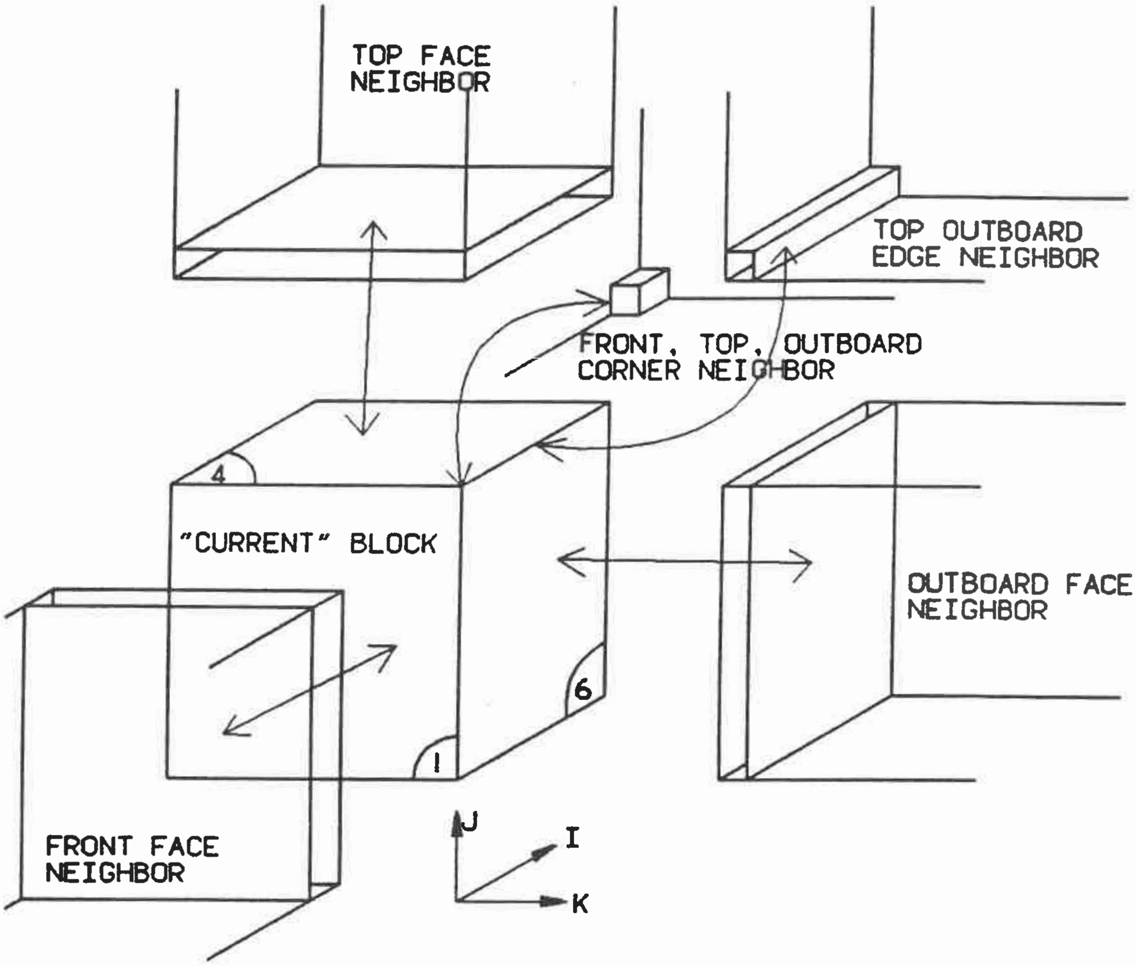


Figure 26 : Current block, extended block and halo grid planes.

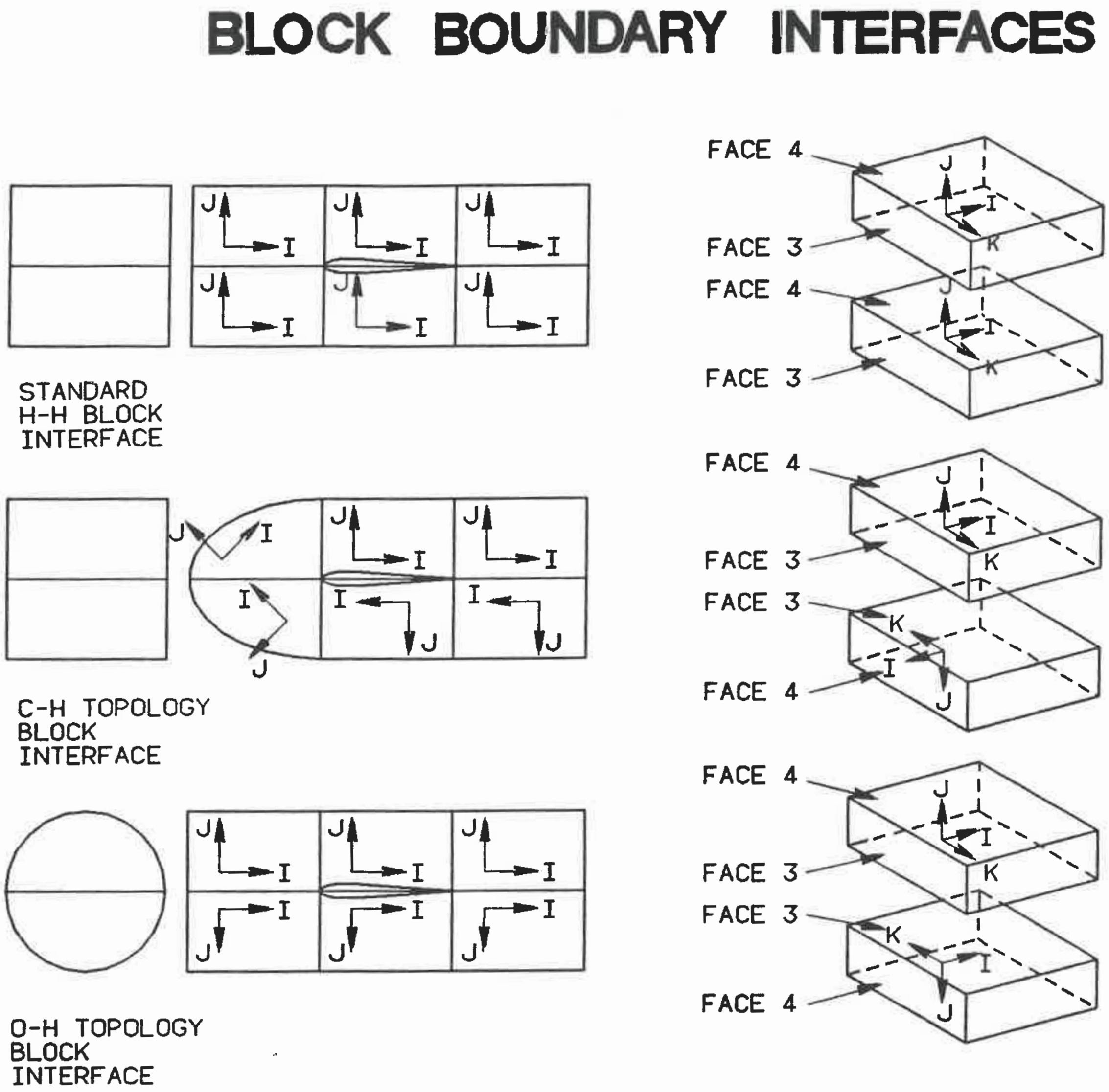


Figure 27 : Block boundary interfaces.

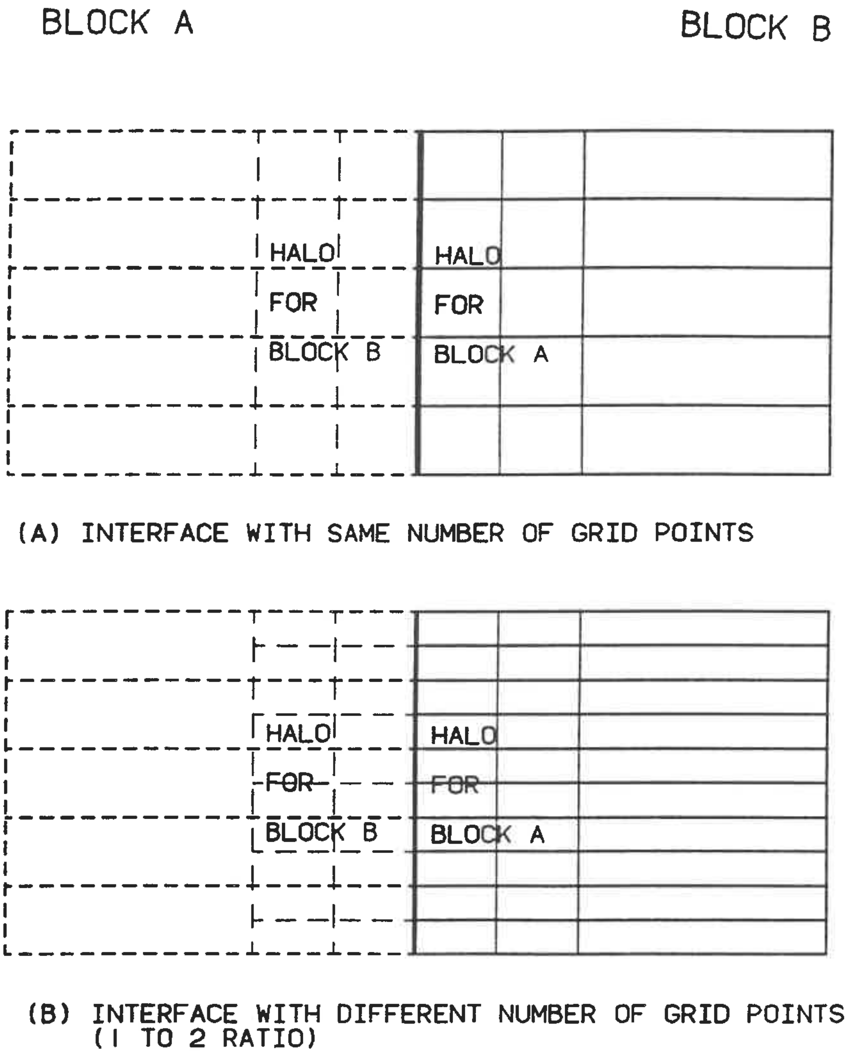


Figure 28 : Block interface boundary conditions.

CHAPTER 6 : Multi-Block Algorithm

6.1 TOPOLOGY DEFINITION

In the multi-block approach, the space around the aircraft configuration is divided into subregions or blocks. The flow equations are solved in each block in sequence as they would be in a single-block solver and appropriate transfer of information is arranged between the various blocks. Following the investigation of several possible options, we propose and have implemented in our code the following multi-block strategy which offers the balance of flexibility and accuracy required for production work in the aeronautical industry :

1. All blocks are hexahedrons with arbitrary shape in space but conceptually similar to a cube, with 6 faces. Some of the faces may collapse to a line or a point, giving physically singular blocks.
2. The grid generated inside each block has an H-H structure as shown in figure 29 . Each face of such a block corresponds to a constant value of one of the grid indices i, j and k. Each block edge corresponds to constant values of two of the grid indices.
3. Boundary conditions can only be applied at a block face. There is no cut or solid surface inside a block. At present, a face can only have one type of boundary condition associated with it and all the grid points on one block face must belong to the same type of boundary. The code is written to permit a proposed automatic grouping of blocks into *“block superelements”*. These superelements will effectively have different boundary conditions on different patches of a superelement face.
4. There is a one to one ratio of grid points on opposing faces of adjacent blocks and the grid

lines are continuous across the block interfaces. The solver requires only C0 continuity of the grid lines across the block faces but in order to improve the accuracy of the calculation, C2 continuity (continuous second derivative of the grid lines) was imposed across most interfaces of the grid generated for the Challenger configuration. The only exceptions occur at interfaces originating at solid boundaries where the orthogonality requirements allow only a C0 continuity in the H-H grid.

5. There are no interpolations at the block interfaces. The interfacing of adjacent blocks is through the use of a halo of two grid planes from the adjacent blocks. When solving for the equation at a cell vertex, the variables at the cell vertices located one mesh step around this vertex in all three directions are used to calculate the Euler fluxes. However, the calculation of the artificial viscosity coefficients at this vertex makes use of the flow variables at two mesh steps around the vertex in all three directions. Initial experience with the code indicated that in order to avoid solution discontinuities at block junctions, it is important that the solution at block interfaces be calculated as much as possible in the same manner as at block interior nodes. Since the artificial viscosity used in the code affects the accuracy and stability of the solution, we have maintained the baseline artificial viscosity discretization and elected to carry halos of grid points two grid planes deep inside adjacent blocks.
6. The solution is calculated once in each block at a given iteration level. The calculation progresses through all blocks in sequence, according to the block number. Once all the blocks have been relaxed, the cycle is repeated. The code uses local time-stepping. For steady flow calculations where intermediate solutions are not required, it is possible in the **MBTEC** code to run more iterations in the blocks where the solution converges more slowly, such as blocks with extensive supersonic flow.

7. The block decomposition, topology definition and the grid generation are done outside of the multi-block Euler code. A heavy emphasis was put on the ability to complete most of the grid definition work on the same CADAM or CATIA workstation where the geometry of the aircraft is defined. A translator program, MBLOCK, was written to extract from the CAD drawing all the information required for input to the grid generation codes. In addition to the aerodynamic and numerical input data, the **MBTEC** program reads two other input files : a **topology file** which defines the relationship of all blocks and block boundaries and a **grid coordinates input file** which gives the x, y and z coordinates of the grid points in all the blocks.
8. The program outputs several files which can be used to interpret the results : Convergence characteristics, aircraft surface data for analysis with AMI's OMNI3D software and field data for analysis with Sterling Software's SSV-4D software. Both graphics applications run on Silicon Graphics workstations.

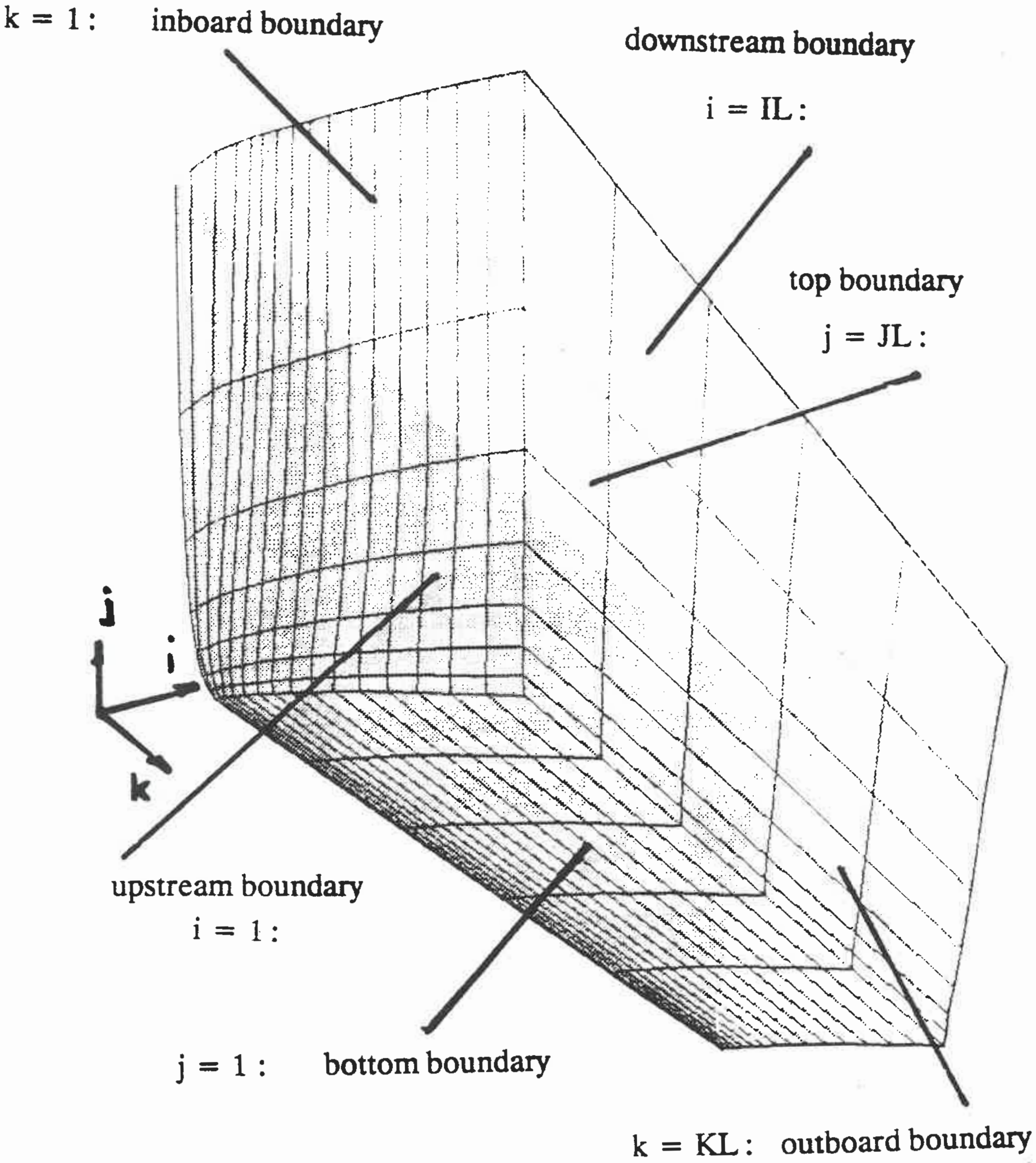


Figure 29 : H-H grid topology used in the multi-block code.

6.2 ALGORITHM STRUCTURE

The program is written in a modular form with seven important parts :

1. Multi-block input module
2. Multi-block data interpretation module
3. Geometric data computation module
4. Multi-block data management module
5. Flow solver module
6. Forces and moments calculation module
7. Output module

6.2.1 Multi-Block Input Module

This module is made of three subroutines : **INPUT**,**MESHIN** and **FLOWIN**. The code requires the input of three sets of data :

a. *Numerical and Aerodynamic Input Data*

- Title of the case.
- Numerical parameters controlling the calculations and the algorithm : number of iterations, time stepping parameters, artificial viscosity coefficients, residual averaging and enthalpy damping parameters.
- Aerodynamic flow conditions : Mach number, angle of incidence and yaw angle.
- Engine operating conditions. Inlet mass flow ratio; exhaust Mach number, total pressure and total temperature. In the multi-block code, the inlet and exhaust planes may be covered by several blocks. A short-cowled turbofan engine such as the CF-34 has separate fan and core exhaust with different flow conditions. The input data defines clearly the flow conditions pertaining to each block face located on the engine inlet or

exhaust planes.

- Geometrical data used to normalize the aerodynamic forces and moments : wing reference area, reference chord and moment reference center.

b. *Multi-Block Topology Input Data*

The information required to connect the various blocks into the correct physical domain must be supplied to the Euler code. This file, produced by the grid generator, is a direct interpretation of the information coded on the CADAM drawing by the user, during the domain decomposition phase. The multi-block Euler code requires information on the total number of blocks (**NBLK**) as well as the following topological information for each block :

- The block index number (**MB**).
- The maximum grid dimension in each coordinate direction (**IMAX**,**JMAX**,**KMAX**).
- The index of the *"group"* or **"block superelement"** to which the block belongs (**GROUP**). The topology requirements of the Euler code (one to one face matching, single boundary condition per face, etc.) result in large numbers of blocks for complicated aircraft configurations (600 blocks for the Challenger complete aircraft configuration. However, in order to simplify the grid generation process, the grid is generated in the smallest number of simply connected blocks that can be defined for the configuration (16 blocks for the Challenger). After a grid is defined in each one of these bigger blocks or *"block superelements"*, the grid generation program automatically subdivides them into the larger number of blocks required to satisfy the Euler code topology requirements. This procedure has two advantages : first, it ensure perfect grid continuity across the boundaries of block belonging to the same group; secondly, the group information supplied to the multi-block Euler codes allows the solver to amalgamate groups together if necessary to improve the efficiency of the solver.

- For each one of the six faces of the block : the face index number (l, 1 to 6) and a number indicating the boundary condition attached to the face (**IBC**) :

IBC = 0 : Side flow boundary
IBC = 1 : Inflow or outflow boundary
IBC = 2 : Inflow or outflow boundary
IBC = 3 : Plane of symmetry boundary
IBC = 4 : Solid surface boundary
IBC = 5 : Block interface (wing cut)
IBC = 6 : Block interface
IBC = 7 : Engine inlet boundary
IBC = 8 : Engine exhaust boundary
IBC = 9 : Interior face collapsed to a singular line

Initially, inflow and outflow boundaries were marked separately. However, the non-reflective boundary conditions used in the code permit the positioning of these boundaries relatively close to the aircraft surface. the direction of the flow on some of these boundaries is therefore not known a priori. If IBC = 1 or IBC = 2, the program tests every node on the boundary to establish whether it has inflow or outflow during the calculations. It then applies the appropriate boundary condition.

- The block index number of the block adjacent to the face (**IBLOK**, 0 if the face is on a solid surface of a far-field boundary).
- The number of the aircraft component to which the face belongs (**ICOMP**). This number is defined only for faces which are part of the aircraft surface. This information is required in order to identify upper and lower surfaces of a wing and to calculate the forces and moments acting on the aircraft surface.

- A face reversal parameter (**IREV**). If this parameter is not zero, it means that the adjacent block has a grid index running in the opposite direction; for example, in the wake of a wing modelled with blocks making an overall C-topology : the upper block has an i-index running downstream whereas the bottom block has an i-index running upstream.

c. *Mesh Input Data*

A file containing the x,y,z coordinates of the grid points for all the blocks specified in the topology file is read by subroutine **MESHIN**. This file is generated by the elliptic grid generation program. The file is usually very large. In order to reduce the time required to read the data, the file is unformatted and read in an asynchronous manner.

d. *Flow Computation Restart Data*

Euler calculation for complete aircraft configurations consume large amounts of CPU time on the computer. During the course of this development, it was found useful to implement a restart capability. At the end of a normal run, the program outputs a large file with the values of the conservative variables at all the grid node points. It is subsequently possible to read this file (subroutine **FLOWIN**) and continue the computations for an additional number of iterations or to use it as an initial flow field for computations at slightly different flow conditions. In order not to loose accuracy during the read and write operations, this file is written unformatted.

6.2.2 Multi-Block Data Interpretation Module

This module is made of the subroutines **CONECT**, **TAG** and **TAG2**. The program reads the coordinates of the grid in each block independently. An interpretation of the data structure of the multi-block grid is therefore necessary. The program uses the information in the topology file to interpret the grid characteristics, connect adjacent blocks, create halo cells, identify boundary conditions for each face and flag the faces belonging to the same aircraft component. The calculation of the Euler fluxes required to advance the solution at a node point (**i,j,k**) in the domain involves the variables at the vertices of the 8 cells surrounding the node point as shown in figure 30(a). The calculation of the artificial viscosity fluxes requires the knowledge of the values of the conservative variables at five node points along each axis, as shown in figure 30(a). There are two exceptions to this :

- At a solid boundary or on a plane or symmetry, where the solution is calculated using only variables lying on the inside of the domain (figure 30(b)).
- At an inflow, outflow or side boundary, where the last node point to be calculated is the one before the boundary. The Euler fluxes in this case are calculated in the same fashion as for an interior node point but the artificial viscosity fluxes are calculated using only four node points along the direction leading to the boundary since the fifth point is not available (figure 30(c)).

The multi-block code was formulated in such a way that the block boundaries are transparent to the solution. At a block boundary which is inside the flow field, the solution is calculated as much as possible in the same fashion as at any point inside the block. It was therefore necessary to provide at the edge of every block face inside the domain two layers of node points from the adjacent block. These extra layers of grid planes will be designated as the "**halo grid planes**". They provide two "**halo cells**" on the side of the block as shown in figure 26. The data from the input topology file is used by subroutine **CONECT** to identify the grid points from adjacent blocks making the "halo grid planes". The geometry of the halo grid points and halo cells (x,y,z coordinates of those grid points, cell volumes and cell face areas and normals) are identified once at the beginning of the execution and carried by the program along with the data of the block's own grid points. The flow variables at the halo grid points must be updated after each iteration to their new values obtained in the solution of the blocks to which they belong. Subroutine **CONECT** calculates for each block interface the index values of the associated "halo grid points" to make this update possible. By interpreting the data from the input topology file, the program determines and stores the following information for each block :

- The index number of the block adjacent to each one of the six *faces* of the "current" block MB : **IBLOK(l,MB)**. The index is zero if the face has no neighbour (face on domain boundary or solid surface).
- The index of the block adjacent to each one of the twelve *edges* of the "current" block MB : **IEDGE(m,MB)**. The index is zero if the edge has no neighbour (edge on domain boundary or solid surface).
- The index of the block adjacent to each one of the eight *corners* of the "current" block MB : **ICORN(n,MB)**. The index is zero if the corner has no neighbour (corner on domain boundary or solid surface).

This information is then used to construct an **"extended" block**, which is made of the grid planes of the current block plus the halo grid planes, the number of which may be zero or two for a given face, depending on the boundary condition on the face. The grid indices in all three spatial directions are modified to correspond to the extended block topology. Special cases occur at blocks adjacent to aircraft surfaces. In order to avoid discretizing the flow equations across solid boundaries, subroutine **CONECT** assigns parametric values to each block face indicating its location in relation to wing surfaces as shown on figure 31.

- Parameter **JPAR3(l,MB)** : this parameter indicates the location of face 3 of a block (see index convention on figure 4) with respect to the wing :

JPAR3(l,MB) = 0 : face 3 not on a cut of any wing surface.

JPAR3(l,MB) = 1 : face 3 located on a wing cut downstream of a wing surface.

JPAR3(l,MB) = 2 : face 3 located on a wing cut upstream of a wing surface.

JPAR3(l,MB) = 3 : face 3 located on wing cut upstream of a wing surface and downstream of another wing trailing edge.

Parameters **JPAR4**, **JPAR5** and **JPAR6** are similarly defined for faces 4, 5 and 6 of a block.

- Parameters **KPAR3, KPAR4,KPAR5** and **KPAR6** are defined to indicate whether face 3, 4, 5 or 6 of a block is located inboard (-1) or outboard (+1) of a wing tip.
- Parameters **JKPAR3, JKPAR4,JKPAR5** and **JKPAR6** are defined to indicate whether face 3, 4, 5 or 6 of a block is located diagonally wing cut.

0 : face not diagonally located with respect to a wing surface.

-1 : face located inboard and downstream of a wing surface.

1 : face located outboard and downstream of a wing surface.

-2 : face located inboard and upstream of a wing surface.

2 : face located outboard and upstream of a wing surface.

The topology input data serves also to locate special points in the grid (singular lines, singular points, blocks faces adjacent to singular lines or singular points) where the calculation of convective and dissipative fluxes must be done in a non-standard fashion. These points are flagged in subroutines **TAG** and **TAG2** for processing by the flow solver. The utmost care must be exercised in the treatment of these special points not to compromise the convergence of the calculations. This item necessitated lengthy experimentation of possible options.

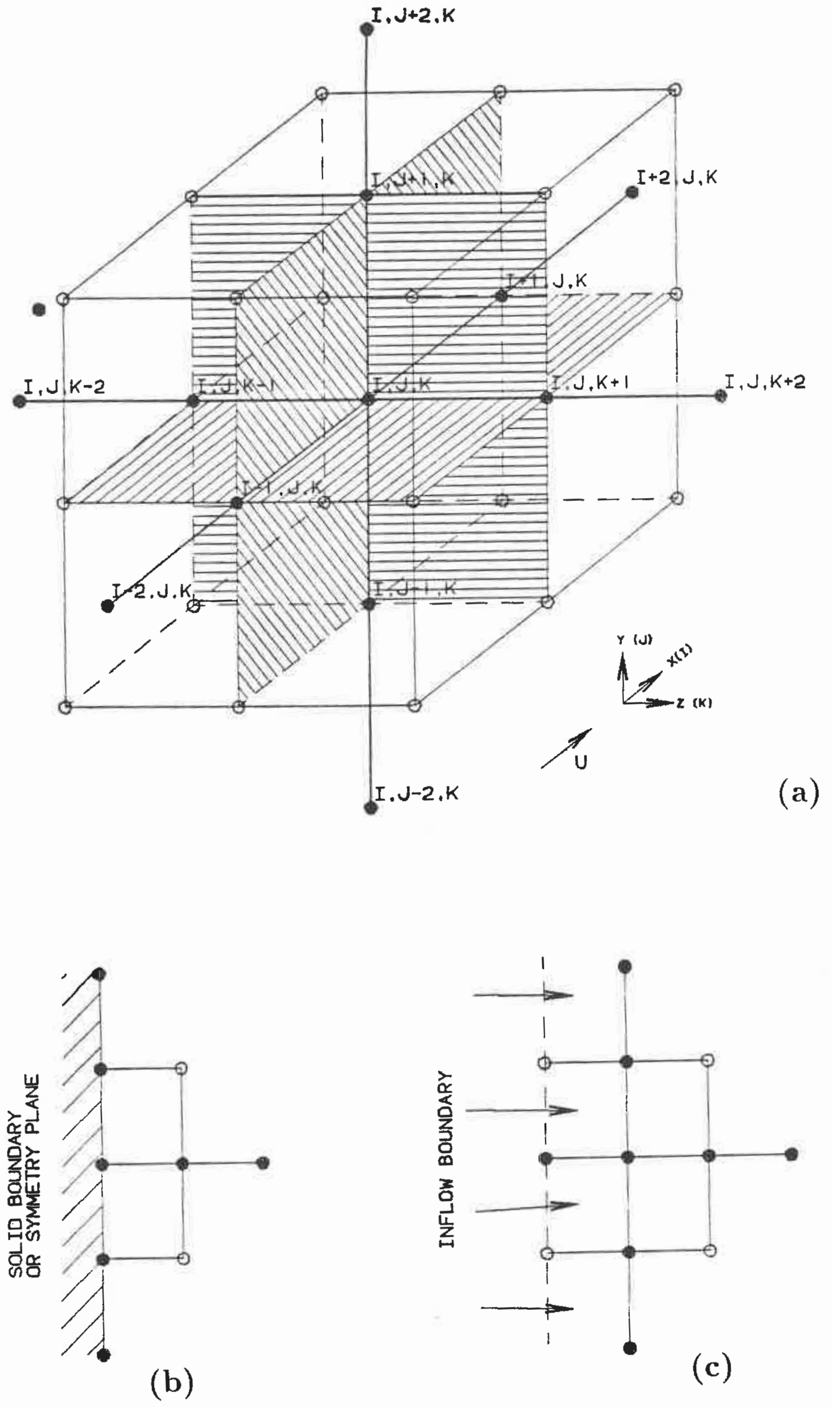


Figure 30 : Discretization cells used in the multi-block code. (a) Standard cell; (b) Solid boundary cell; (c) Inflow boundary cell.

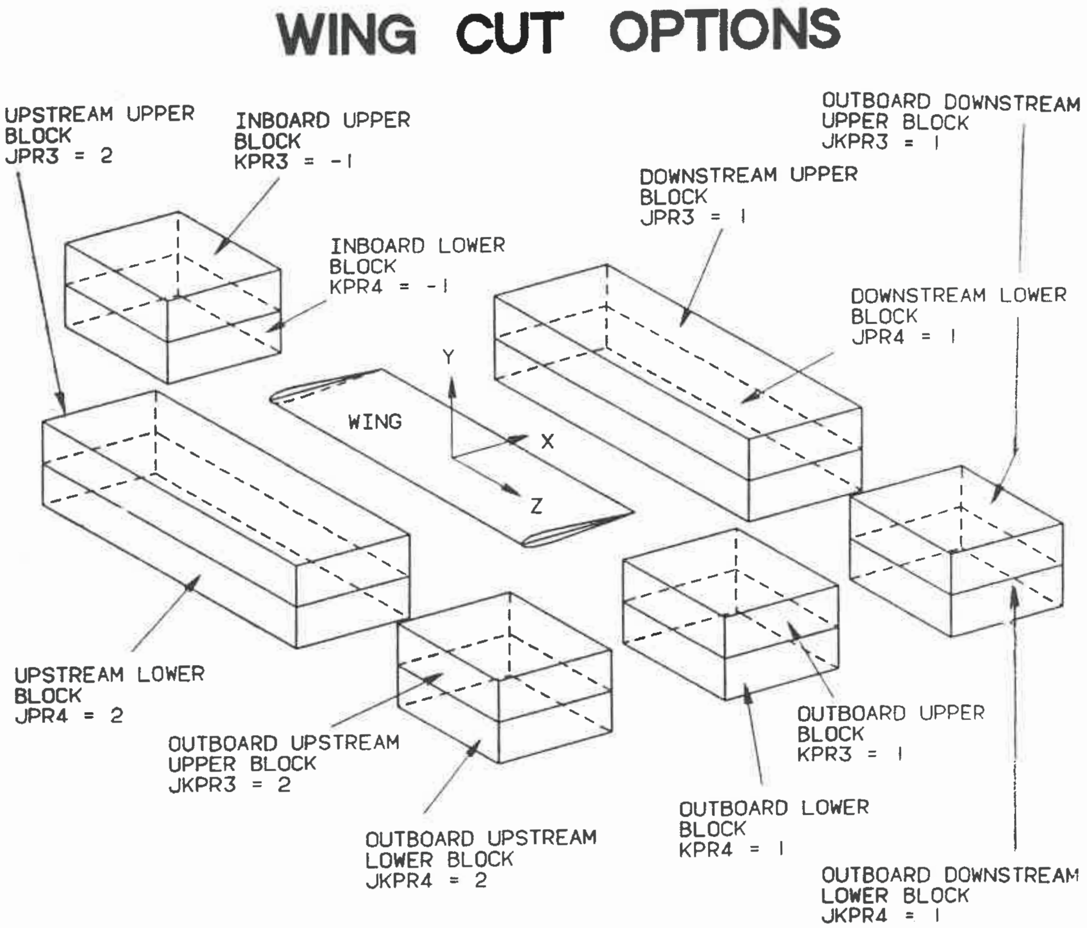


Figure 31 : Wing cut options at block boundary interfaces.

6.2.3 Geometric Data Computation Module

The calculation of the geometry characteristics of the grid in each block is organized by subroutine **SETUP**. The calculation is made through appropriate calls to subroutines **MBGEOM**, **METRIC** and **NORMAL** an the resulting information is stored in memory or on disk for each block. Subroutine **MBGEOM** builds the matrix of (x,y,z) coordinates for the "extended" blocks including the "halo" grid planes determined in subroutine **CONECT**. In the multi-block method proposed here, the Euler equations are discretized only at *"inner block"* grid points which are not on a far field boundary. The flow variables at grid points in the block interface "halo" grid planes are not updated. Subroutine **MBGEOM** uses the topology input data to determine two index limits for each block :

- The maximum dimensions **IL**, **JL** and **KL** of the "extended" blocks, including the "halo" grid planes.
- The lower and upper limit of the flow update in all three coordinate directions : **I1** to **I2**, **J1** to **J2**, **K1** to **K2**. The relation between these limits and the "extended" block maximum dimensions will be discussed in the section related to the multi-block flow solver.

The subroutine **METRIC** calculates the cell volumes, the projected areas of all cell faces and the total area of each block face. The subroutine **NORMAL** calculates the surface normals on block boundaries and the tangent vectors at all intersections between solid surfaces. The normals are used in the solid surface and far field boundary condition formulations. The tangents vectors are used to impose an additional boundary condition at the junction between solid surfaces and between solid surfaces and the plane of symmetry. The grid geometric characteristics are calculated once for each block at the beginning of the execution of the program and saved by subroutine **SETUP** in a large array from which the data is recovered each time a block is solved for. The multi-block code developed here uses an H-H grid topology. This topology and the constraint of one type of boundary condition per block face results in block junctions intersecting the wing surface as shown in figure 32. This figure shows the junction of four blocks at a wing leading edge. Since the equations are solved and the boundary conditions applied in each block independently, it is important to make sure that the normals to the surface used at the junction point in all four blocks are the same. The program assigns to every node point a normal obtained as the average of the four cell faces joining at that node. At the leading edge node, the normals calculated in the normal fashion would be different in the upper and lower blocks. Subroutine **SETUP** uses the topology data to identify nodes with this problem and places a unique value of the surface normal in all four blocks.

6.2.4 Multi-Block Data Management Module

The flow solver works on one block at a time, performing normally one complete time integration cycle. The block being solved for is designated in the rest of the text as the *"current block"*. The grid information and variables at the grid points of the blocks not being solved for must be preserved. The data is stored either in memory if the space available on the computer is large enough or written to disk. At the end of a cycle in a block, the current block updated variables are stored and the information pertaining to the next block to be calculated is retrieved. An outer loop introduced in the program cycles the flow solution through all blocks at each iteration. The information restored before the calculation is performed includes the grid points x,y,z coordinates, the cell volumes, the cell face areas and cell face directed normals and the conservative variables at the grid points. The solution is then advanced through one time step and the updated flow variables stored again in core memory for use at the following iteration for that block. This information exchange is organized by subroutines **RBLOK** and **WBLOK**. **RBLOK** reads from storage the geometric data and the flow variables of to the next block to be solved for and transfers the information into arrays used by the flow solver. This subroutine retrieves also the same information for all the grid points in the "halo" of the block. At the end of the iteration for the block, subroutine **WBLOK** writes the updated flow variables of the "inner block" for use at the next iteration. There are three types of data which must be manipulated by the multi-block flow solver :

1. The geometric characteristics of the grid cells. This large amount of data is calculated and stored once at the beginning of the execution for all the blocks in the flow domain. The data for each block is then read by **RBLOK** before the solution is advanced in that block.
2. The block topological data, the geometric data related to the block faces and the block convergence characteristics. This smaller amount of information is calculated at the beginning of the execution and stored in multi-dimensional matrices in which the block number is one of the indices.
3. The flow variables. These are stored in core by **WBLOK** at the end of every iteration in a block and read again by **RBLOK** at the beginning of the next iteration for the same block.

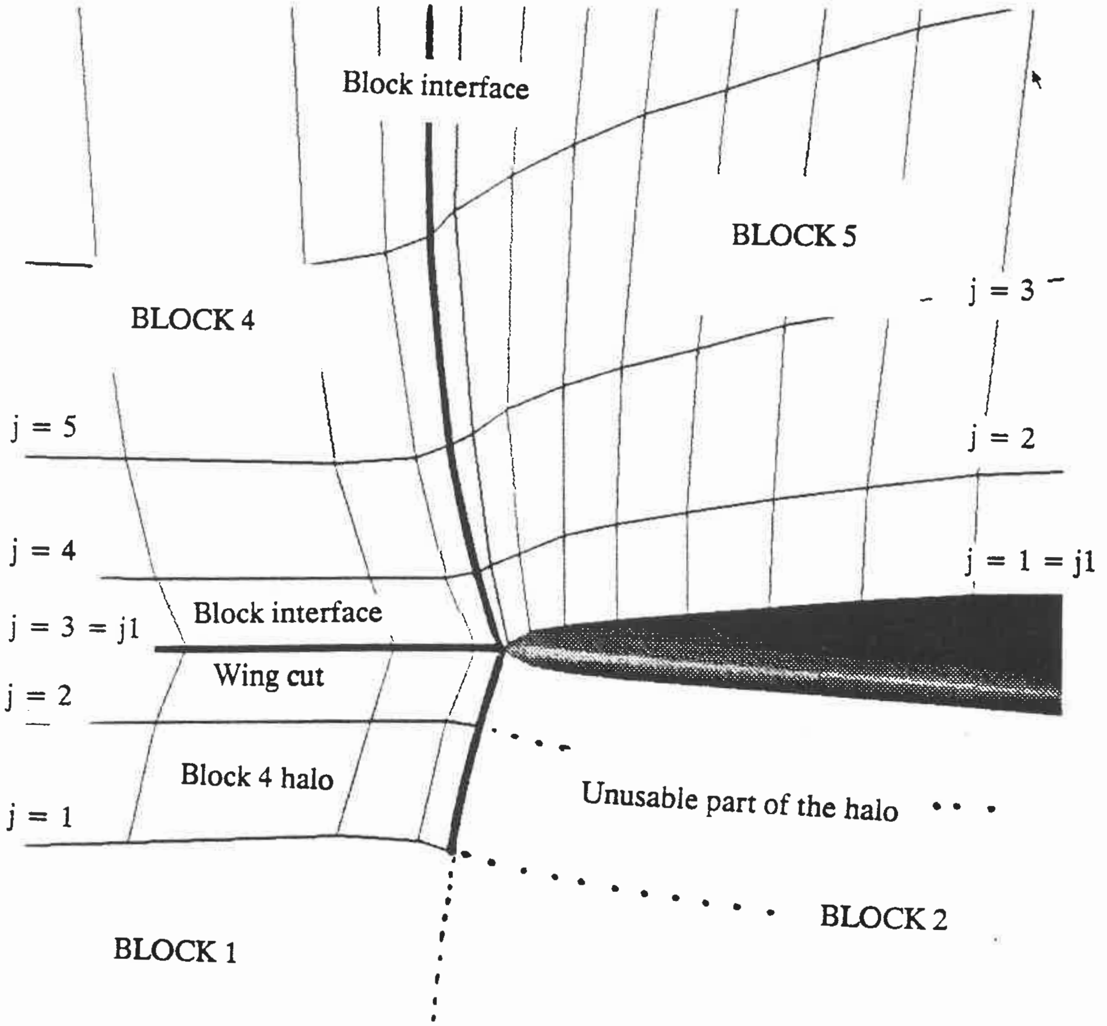


Figure 32 : Special problems of block interfacing near the wing leading edge.

6.2.5 Multi-Block Flow Solver Module

The multi-block formulation has an impact on the calculation of the Euler fluxes and the artificial viscosity fluxes presented earlier in the text. The solver is also affected by the fact that any boundary condition can be specified on any one of the faces of a block. Finally, the flow solver was written to take into account the cases of the special points introduced near aircraft surfaces by the H-H grid topology. The flow solution module is written for a generic block with six faces and an H-H grid. This block is the "extended" block, which is to the "inner" block modified by the addition of "halo" grid planes from neighbouring blocks. Figure 4 indicates the orientation of a typical block with an H-H grid, as def ined earlier. The faces are numbered in the following sequence :

Face 1 : *"Upstream face"* normal to the **i** direction (longitudinal direction)

Face 2 : *"Downstream face"* normal to the **i** direction

Face 3 : *"Bottom face"* normal to the **j** direction (vertical direction)

Face 4 : *"Top face"* normal to the **j** direction

Face 5 : *"Inboard face"* normal to the **k** directioni (lateral direction)

Face 6 : *"Outboard face"* normal to the **k** direction

The maximum limits of variation of the indices in the "extended" block are :

- **i** direction : from **1** to **IL**
- **j** direction : from **1** to **JL**
- **k** direction : from **1** to **KL**

Some of the grid planes are far field boundary planes or are halo grid planes belonging to a neighbouring block. The variables at points on far field boundaries are updated when the boundary condition is applied. The variables at points on the "halo" grid planes are used for the block interface boundary condition and are not updated when solving for the "current" block. The actual range of grid points at which the solution is updated when solving for the block depends on the boundary conditions attached to the six faces. The range is defined as follows :

i = i1 : first grid plane to be updated in the **i** direction
i = i2 : last grid plane to be updated in the **i** direction
j = j1 : first grid plane to be updated in the **j** direction
j = j2 : last grid plane to be updated in the **j** direction
k = k1 : first grid plane to be updated in the **k** direction
k = k2 : last grid plane to be updated in the **k** direction

These limits are calculated once for each block at the beginning of the execution. A summary for the cases most often encountered is shown in table 2. For example, at the front face of a block located at the inflow boundary, the solution is updated in subroutine **EULER** only from **i = i1 = 2**. The variables at points on the grid plane at **i = 1** are obtained as a result of the application of the inflow boundary condition. If the downstream face of the block is an interface with another block, the program adds a "halo" of two grid planes from the adjacent block to be used for discretizing the equation at the interface. The index limit **IL** of the "extended" block including the "halo" grid planes is therefore higher than the original block index limit by two. The solver calculates the solution from **i1 =2** to **i2 = IL-2**.

Table 2 : Definition of block index limits.

Block Face		**Boundary Condition**		**Index limit**
1	Upstream	1	Inflow	i1 = 2
		6	Interface	i1 = 3
		8	Exhaust	i1 = 2
2	Downstream	2	Outflow	i2 = IL-1
		6	Interface	i2 = IL-2
		7	Inlet	i2 = IL-1
3	Bottom	0	Side flow	j1 = 2
		1	Inflow/Outflow	j1 = 2
		3	Symmetry plane	j1 = 1
		4	Solid surface	j1 = 1
		5	Interface/cut	j1 = 3
		6	Interface	j1 = 3
4	Top	0	Side flow	j2 = JL-1
		1	Inflow/Outflow	j2 = JL-1
		3	Symmetry plane	j2 = JL
		4	Solid surface	j2 = JL
		5	Interface/cut	j2 = JL-2
		6	Interface	j2 = JL-2
5	Inboard	0	Side flow	k1 = 2
		1	Inflow/Outflow	k1 = 2
		3	Symmetry plane	k1 = 1
		4	Solid surface	k1 = 1
		5	Interface/cut	k1 = 3
		6	Interface	k1 = 3
6	Outboard	0	Side flow	k2 = KL-1
		1	Inflow/Outflow	k2 = KL-1
		3	Symmetry plane	k2 = KL
		4	Solid surface	k2 = KL
		5	Interface/cut	k2 = KL-2
		6	Interface	k2 = KL-2

6.2.5.1 Calculation of the Permissible Time Step

The **MBTEC** program uses a fully explicit time stepping scheme. This scheme has a maximum permissible time step limited by the Courant-Friedrichs-Lewy (CFL) condition. This condition states that for the computations to remain stable, the domain of dependence of the numerical scheme must contain that of the partial differential equations. The information conveyed by a partial differential equation travels at speeds corresponding to the characteristic wave speeds. The characteristic speeds for the Euler equations can be found by diagonalizing the equations and solving for the eigenvalues. Let s be the highest characteristic wave speed. In a given time Δt, the information propagates therefore as far as over a distance given by $s\Delta t$. Consider a simple one-stage numerical scheme. At every iteration, the solution is advanced by a time step Δt while numerical information is transferred over a distance Δx corresponding to the support of the numerical scheme. The CFL condition limits the maximum time step to a value such that :

$$\Delta x \geq s\Delta t \tag{6.2.1}$$

The maximum permissible time step for this single-stage scheme is therefore :

$$\Delta t = \frac{\Delta x}{s} \tag{6.2.2}$$

The expression CFL $= s\frac{\Delta t}{\Delta x}$ is known as the *"Courant number"*. For the multi-stage scheme used in **MBTEC**, it is found that the maximum Courant number that can be used is $2\sqrt{2}$. The code uses local adaptive time stepping as a means of accelerating the convergence to steady state. This means that at each grid point, the solution is advanced by the highest Δt compatible with local stability of the scheme. Subroutine **STEP** calculates first the average spectral radius of the Jacobian matrices in each one of the three directions in space.

For the flux across the cell face in the **i** direction, it can be estimated as :

$$\lambda_i = |\vec{u}\vec{S}_i| + c\sqrt{S_{ix}^2 + S_{iy}^2 + S_{iz}^2} \tag{6.2.3}$$

where c is the speed of sound, $\vec{u}$ the local fluid velocity and $\vec{S}_i$ the cell face directed area in the **i** direction. The code calculates similarly the spectral radii λ_j and λk in the **j** and **k** directions. A conservative estimate of the local time step for a nominal Courant number of 1 is given by :

$$\Delta t = \frac{\Omega_{ijk}}{\lambda_i + \lambda_j + \lambda_k} \tag{6.2.4}$$

where Ω_{ijk} is the control volume. The spectral radii calculated in **STEP** are also used to scale the dissipative coefficients of the artificial viscosity. The constraints of the multi-block code cause the presence of high aspect ratio mesh cells in the grid. To avoid large unbalances of the coefficients in the **i**, **j** and **k** directions, the artificial dissipation is redistributed between the three coordinate directions. This is done by scaling the spectral radii in the following manner :

$$R_i = \lambda_i(1 + \frac{1}{r_{ij}} + r_{ki}) \tag{6.2.5}$$

$$R_j = \lambda_j(1 + \frac{1}{r_{jk}} + r_{ij}) \tag{6.2.6}$$

$$R_k = \lambda_k(1 + \frac{1}{r_{ki}} + r_{jk}) \tag{6.2.7}$$

where :

$$r_{ij} = (\frac{\lambda_i}{\lambda_j})^\alpha \tag{6.2.8}$$

$$r_{jk} = (\frac{\lambda_j}{\lambda_k})^\alpha \tag{6.2.9}$$

$$r_{ki} = (\frac{\lambda_k}{\lambda_i})^\alpha \tag{6.2.10}$$

The exponent α can be used to construct different dissipation schemes. The best results in the present work were obtained with an exponent $\alpha = 1$.

The H-H topology used in the multi-block code generated special cases of blocks having a face adjacent to a wing solid surface. When calculating the spectral time step close to such a boundary, care must be taken not to discretize across a solid surface in the halo of the block. The **MBTEC** code uses a cell-vertex formulation in which the control volume for a node point is made of the eight cells surrounding the node. The H-H block formulation generates special cases where the control volume can be made of two, three, four or six cells, as shown in figure 33. The evaluation of the spectral radii and of the time step was formulated to cater for all these special cases. The dissipative coefficients, calculated in this subroutine, are set to zero on all faces with solid surface or plane of symmetry boundary conditions. The code also uses adaptive time stepping. In this scheme, the local time step is modified by an adaptive coefficient built as a sensor of the pressure gradients in the three coordinate directions :

$$\Delta t_{adaptive} = \beta \Delta t \tag{6.2.11}$$

where

$$\beta = \frac{1}{1 + min[min((CFL-1),0), 2(\nu_i + \nu_j + \nu_k)]} \tag{6.2.12}$$

CFL is the Courant number. The adaptive sensor ν_i in the **i** direction is defined as :

$$\nu_i = |\frac{p_{i+1,j,k} - 2p_{i,j,k} + p_{i-1,j,k}}{p_{i+1,j,k} + 2p_{i,j,k} + p_{i-1,j,k}}| \tag{6.2.13}$$

The sensors ν_j and nu_k in the **j** and **k** directions are defined similarly. Near leading edge where the H-H grid topology gives **i+1** locations inside the airfoil, a mirroring procedure is used which gives $p_{i+1,j,k} \;=\; pi-1,j,k$. A similar procedure was used in the **j** and **k** directions near solid surfaces.

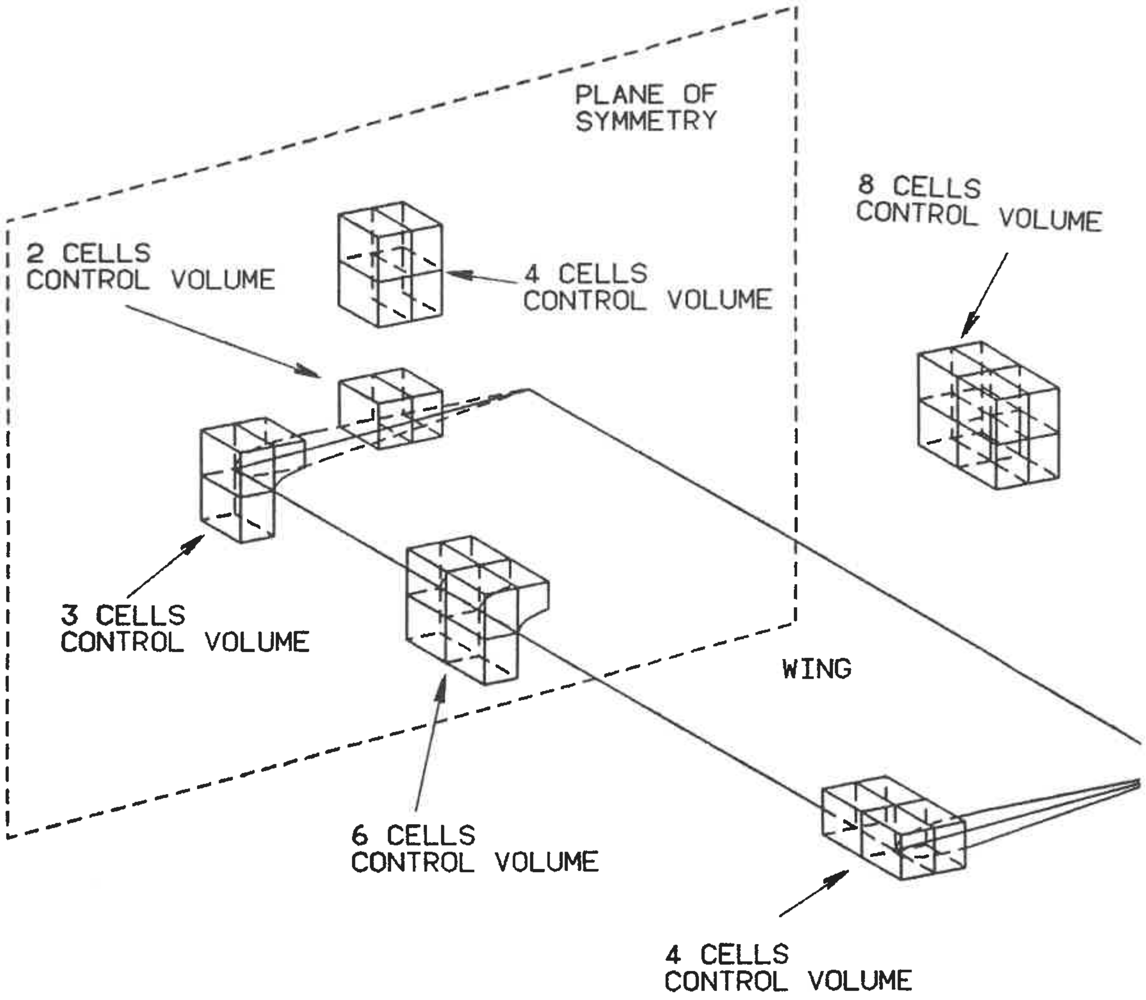


Figure 33 : Control volumes used in the multi-block code.

6.2.5.2 Calculation of the Convective Fluxes

The algorithm used in the code requires the calculation of the convective fluxes $Q(W)_{i,j,k}$ through the boundaries of the control volume defined around every node point in the grid. A normal control volume is made of the eight hexahedral cells joining at the node. The mass, momentum and energy fluxes are calculated through the faces of each one of the eight cells making the control volume. These fluxes are then accumulated. The fluxes through cell boundaries internal to the control volume cancel each other and the remaining values represent fluxes through the outer boundary of the volume, as described in chapter 4 :

$$Q(W)_{i,j,k} = \sum_{\ell=1}^{6} F_\ell . S_\ell \tag{6.2.14}$$

On a solid surface or on the plane of symmetry, the mass and energy fluxes are zero and the three components of momentum fluxes are reduced to the contribution of the pressure acting on the cell faces. The multi-block code is formulated in a general manner which allows this to occur at any one of the six faces of a mesh cell, depending on the boundary conditions. The calculation of the cell fluxes takes into account that in the multi-block code, the control volume at special points can be made of two, three, four or six mesh cells, instead of eight. Another special case occurs at points near block interfaces adjacent to solid surfaces. Tests were added not to accumulate fluxes through solid boundaries that may exist in the halo of the block. This situation arises from the fact that the flow update boundaries in two adjacent blocks are not always the same. Figure 32 shows an example with an isolated wing case. Block 4, for instance, has an interface with block 1 at its bottom face and hence **J1 = 3** for this block. However, the bottom face of block 5 is the wing upper surface and therefore **J1 = 1** in this block. The solution in block 4 uses the variables at the first two grid planes in block 5 (**i = IL-1** and **i = IL**). The variables at **J = 1** and

J = 2 in these "halo" grid planes are therefore located underneath the wing and should not be used without care. To signal that such a situation may arise, a separate boundary condition code (**IBC = 5**) was introduced, for block interfaces that terminate on a solid surface (this boundary condition is referred to as *"interface/cut"* in table 2).

Both the convergence of the code and the accuracy of the results are affected by the quality of detail with which the fluxes are accumulated near such interfaces. An important part of the formulation of the multi-block code was therefore devoted to clearly identifying cells lying on solid surfaces adjacent to leading edges or wakes and cells lying on interfaces adjacent to solid surfaces. A challenge was to organize in subroutine **EFLUX** the calculation of these special cases without jeopardizing the overall loop structure needed for efficient vector computations.

6.2.5.3 Calculation of the Artificial Viscosity Fluxes

The discretization scheme in the **MBTEC** code uses central spatial differencing and requires therefore the addition of artificial viscosity to suppress oscillations. The addition of the artificial viscosity to the Euler equation, the discretization of these equations and the calculation of the dissipation fluxes $D(W)_{i,j,k}$ were discussed in chapter 4. **The actual implementation of the artificial viscosity scheme in the code proved during the present development to be the most critical part of the multi-block algorithm.** Special attention was paid to the formulation of the dissipative terms $D(W)_{i,j,k}$ near boundaries since *improper treatment was found to lead to distortions of the flux balances and to result in either loss of convergence or in the calculation of inaccurate results.* A great deal of experimentation was necessary in particular to arrive at a satisfactory calculation of these fluxes at the wing cut interfaces (**IBC = 5**). We were helped in this exploratory work by three excellent papers by Turkel [70], Caughey and Turkel [67] and Turkel and Vatsa [71] discussing the effects of artificial viscosity on finite volume algorithms.

The successful discretization of the artificial viscosity terms in subroutine **DFLUX** required attention to the following four items :

- The formulation of the dissipative fluxes second order and forth order differences near solid surfaces, especially near wing leading edges. In particular, it was found that artificial viscosity fluxes $D(W)_{i,j,k}$ calculated in different blocks joining at a wing leading edge must be calculated in the same manner in all these blocks.
- The scaling of the artificial dissipative fluxes required to obtain the right balance of dissipation in the three coordinate directions. This is necessary because of the high aspect ratio cells typically present in multi-block grids generated to conform the topology of this program. The calculation of the scaling coefficients $R_{i,j,k}$ was described in the section on the evaluation of the time step.
- The treatment of the artificial viscosity across the wakes, where they must be calculated. For the calculation of dissipative fluxes, it is better not to treat the wakes as solid surfaces.
- The treatment of the artificial viscosity at block interfaces, particularly at the block junctions near solid surfaces. It is desirable to have a formulation of the dissipative terms which is conservative, that is, which does not add mass, momentum or energy in the field. To achieve this requires that all dissipative terms cancel when summed over the entire flow field. We seeked therefore to ensure that, for each dissipative contribution to a particular point, there was an equal and opposite contribution to a neighbouring

point. This was most difficult to achieve across block boundaries.

6.2.5.4 Time Integration

An algorithm was written to integrate the Euler equation in an arbitrary block with H-H topology. The ordinary differential equation solved at each node point in the flow field is :

$$\frac{\partial}{\partial t}(V_{i,j,k}W_{i,j,k}) + Q(W)_{i,j,k} - D(W)_{i,j,k} = 0 \tag{6.2.15}$$

This integration is performed for each block in subroutine **EULER**, using convective fluxes $Q(W)_{i,j,k}$ calculated in subroutine **EFLUX** and dissipation fluxes $D(W)_{i,j,k}$ calculated in subroutine **DFLUX**. The solution is advanced in time using the five-stage Runge-Kutta integration schemei presented in chapter 4 and the local time steps evaluated in subroutine **STEP**. The maximum permissible time step is calculated at every iteration, the convective fluxes at every Runge-Kutta stage and the dissipative fluxes at stages 1, 3 and 5; the dissipation fluxes are frozen at their previous stage values at the other stages of the scheme. **Euler** also calls subroutine **BCIMP** to apply the appropriate boundary conditions at block faces. It applies the enthalpy damping and calls subroutine **PSMOO** for the residual averaging designed to improve the convergence characteristics. The program uses flags calculated in the topology interpretation subroutine **TAG** to identify blocks located in the vicinity of jet engine exhausts. Enthalpy damping is switched off in these blocks.

6.2.5.5 Application of the Boundary Conditions

The boundary conditions in the multi-block codes are applied in four separate subroutines. subroutine **BCFAR** is used to apply far field boundary conditions on any one of the six faces of a block. These conditions can be inflow, outflow or side flow conditions and relate to subsonic or supersonic boundaries. The formulation of the boundary conditions was presented in the previous chapter. The solid surface and plane of symmetry boundary conditions are enforced in subroutine **BCIMP**. Here too, any one of the six faces of a block can be labelled as a solid surface. The program also detects intersections between solid faces and enforces the tangency of the flow there. Modifications were made to invoke also **BCIMP** for block faces which are not solid surfaces, but which have one edge located on such a surface. A boundary condition similar to the one applied on that edge in the adjacent block sitting on the solid surface is implemented. A similar procedure is used for blocks having only a corner point located on a solid surface. An index logic was devised in order to garantee that all cases are being properly treated without undue checking at every iteration. Subroutine **BCIMP** also equalizes the corrections to the residuals at all nodes collapsed on a single point. This ensures the uniqueness of the solution there. The engine inlet boundary conditions are applied in subroutine **BCINLET** in the manner described in the previous chapter. Engine exhaust boundary conditions are applied in subroutine **BCNOZLE**.

6.2.5.6 Residual Averaging

The residuals at the node points of every block are replaced by a weighted average of residuals at neighbouring points. By smoothing the residuals and increasing the support of the scheme, this procedure relaxes the restriction on the time step imposed by the CFL condition. For maximum convergence improvement, theses averages are calculated implicitly. In a one dimensional case, the residual $R(W)_i$ is replaced by $\overline{R}(W)_i$ using the following expression :

$$-\varepsilon\overline{R}(W)_{i-1} + (1+2\varepsilon)\overline{R}(W)_i + \varepsilon\overline{R}(W)_{i+1} = R(W)_i \qquad (6.2.16)$$

For an infinite interval, this equation has the explicit solution :

$$\overline{R}(W)_i = \frac{1-r}{1+r} \cdot \sum_{q=-\infty}^{\infty} r^q R(W)_{i+q} \tag{6.2.17}$$

where :

$$\varepsilon = \frac{r}{(1-r)^2} \quad , \quad r < 1 \tag{6.2.18}$$

In the case of a finite interval with non periodic boundary condition, one has to choose boundary conditions such that $R(W)_i$ given by the expression above is a solution of the implicit equation, with $R(W)_{i+q} = 0$ if $i+q$ lies outside the interval. This solution can be achieved by setting :

$$\tilde{R}(W)_{I1} = R(W)_{I1} \tag{6.2.19}$$

$$\tilde{R}(W)_i = R(W)_i - r(R(W)_i - \tilde{R}(W)_{i-1}) \tag{6.2.20}$$

$$for \quad I1 + 1 \leq i \leq I2$$

$$\overline{R}(W)_{I2} = \frac{1}{1+r}\tilde{R}(W)_{I2} \tag{6.2.21}$$

$$\overline{R}(W)_i = \tilde{R}(W)_i - r(\tilde{R}(W)_i - \overline{R}(W)_{i+1}) \tag{6.2.22}$$

$$for \quad I1 \leq i \leq I2 - 1$$

In the three-dimensional case of the present work, the smoothing is applied in product form :

$$(1 - \varepsilon_x \delta_x^2)(1 - \varepsilon_y \delta_y^2)(1 - \varepsilon_z \delta_z^2)\overline{R} = R \tag{6.2.23}$$

where **R** is the unsmoothed residual, δ_x^2, δ_y^2 and δ_z^2 the second difference operators and ε_x, ε_y and ε_z the smoothing coefficients given in input to the program. Subroutine **PSMOO** smooths the residuals in the **i**, **j** and **k** directions successively.

6.2.5.7 Convergence Monitoring

The program monitors the convergence of the numerical computations by calculating and storing at every iteration the values of the maximum and average residuals of the discretized equation in the field, the values of the maximum and average enthalpy and the location in the field of the maximum residual and maximum enthalpy. The program also tracks at every iteration the number of supersonic points in the field and the values of the lift coefficient C_L and the drag coefficient C_D. The average contraction ratio of the residuals over one iteration is also calculated at the end of the computations. In the multi-block code, it is necessary to calculate and store these parameters for each block at every iteration. In order to calculate the contraction ratio of the solution in each block, it is necessary to keep constantly in memory the residual values of all the blocks. The program calculates the forces acting on each face of each solid surface boundary at every iteration. The evolution of the total aircraft forces and moments during the calculation is also monitored. The multi-block code is formulated such that the variables at block interfaces are solved for twice, once in each block. The variables in the "halo" grid planes are not updated during a block iteration and remain frozen at their starting values during the Runge-Kutta time-stepping. At an interface, these "halo" grid planes are on the left in one block and on the right in the other block. This, together with some directional aspects of the flow calculation, introduce very small differences in the values calculated at the interface inside the two blocks. These differences are erased by replacing in both blocks the variables at the interface by the average of the values calculated in the two blocks. At node points where more than two blocks interface, the variables are replaced by the average of the values in all joining blocks. This procedure, implemented in subroutine **AVE**, was found to lead to satisfactory convergence characteristics in most of the flow field when applied carefully. A special arrangement was necessary at wing cut interfaces. As shown before, the control volumes used by the blocks located above the wing cut are different from the control volumes used by the blocks located underneath the wing cut, at the nodes in contact with the wing surface. The averaging procedure described above is applied once per iteration, after the solution has been advanced by one time step inside all the blocks. This by itself did not appear to be sufficient to ensure proper convergence of the calculation at wing cut planes near the wing surface. The reason is the fact that the convective fluxes across the mesh cells are evaluated five times during a flow iteration, once at each Runge-Kutta time stage. The dissipative fluxes are evaluated three times during each iteration. It was discovered that the small differences due to the different control volumes on upper and lower blocks were sufficient to cause the convective and dissipative fluxes to be more and more different from one Runge-Kutta evaluation to the next and result in different variables calculated at the end of the cycle. A new subroutine **AVELE** was written to average the values of the conservative variables at wing leading edge, trailing edge and tip points after each Runge-Kutta stage. This procedure improved considerably the overall convergence characteristics of the computations.

6.2.6 Forces and Moments Module

In the multi-block code, any one of the six faces of a block can be a solid surface with any orientation in space and may belong to the wing, the body, or any other component of the aircraft. To obtain the forces and moments on an aircraft component requires the accumulation of the forces and moments calculated, usually in different blocks, on the various block faces making up that component. The following scheme was therefore implemented in order to calculate properly the forces and moments acting on the aircraft :

- In the topology input data, the block faces which are solid surfaces are identified with the parameter **IBC** = **4**).
- In the topology input data, the component to which a solid surface face belongs is identified by the parameter **ICOMP**. The values **ICOMP** = 1 to 98 designate wing-type components and the values **ICOMP** = 99 and above designate body-type components.

  For example, the 600 blocks complete Challenger aircraft case has the following component numbers :

Wing + winglet	:	component # 1
Horizontal tail	:	component # 2
Engine Pylon	:	component # 3
Vertical fin	:	component # 4
Engine nacelle	:	component # 99
Fuselage	:	component # 100

- Subroutine **SECPRP** calculates the pressure coefficients at the nodes located on all block faces identified as solid surfaces.
- Subroutine **PATCHF** integrate these pressure coefficients to obtain the three force components **CX**, **CY**, **CZ** and three moment components **CMX**, **CMY**, **CMZ** on each block face identified as a solid surface. These forces and moments are calculated in the global coordinate system used to model the geometry of the aircraft. This is such that they can be added later without coordinate transformation. The values of these forces and moments for each face are stored in memory for all blocks.
- Subroutine **COMPF** adds the forces and moments of all the block faces belonging to the same component. **COMPF** also calculates the forces and moments in the wind axis

system and converts all the aerodynamic coefficients to the reference areas and lengths given in the input for the complete configuration. The force and moment coefficients of all the various components are then added to obtain the global aircraft characteristics.

In many cases, the upper and lower surface of an aircraft wing is made up of several blocks in the chordwise and spanwise directions. The program uses the data in input to reconstitute the complete wing surface pressure distribution. This is in view of the link to an interactive boundary layer calculation. This data can then be interpolated to yield the chordwise pressure distribution calculated on any wing section.

6.2.7 Output Module

The multi-block Euler code is used to analyze complex aircraft configurations. It was therefore necessary to introduce new tools for sorting out the large amount of data that is calculated. The multi-block program produces after each run several output files :

- **Printer output file**

  The program outputs a line printer file **OUTPUT.OUT** including :

  - An echo of the numerical and aerodynamic input data.
  - A list of the topology relationships of all the blocks, as calculated by **CONECT** including : the index numbers of the faces, edges and corner neighbours for all blocks, the list of wing cut options (**JPR3**, etc.) calculated for all blocks, the minimum and maximum dimensions of the *"inner"* and *"extended"* blocks.
  - The areas of all the blocks faces, as calculated by **METRIC**.
  - The list of all the flags determined for special points by subroutines **TAG** and **TAG2**.
  - The number of grid points inside each *"inner"* and *"extended"* block as well as the

total number of grid points in the field.

- The convergence parameters (maximum and average residuals : value and location) for all blocks, after a number of iterations specified by the user in input.
- A summary of the convergence characteristics (starting and end residual values, average contraction ratios) for all blocks, at the end of the calculations.
- A list of forces and moments for each aircraft component and for the complete configuration.

- **Convergence history output file**

The program outputs a file **OUTPUT.HIST** containing the history of the convergence of the residuals in each block. This file can be used to produce graphic representations of the convergence characteristics.

- **Restart file**

The program outputs an unformatted file **OUTPUT.REST** containing the values of the conservative variables at all grid points in all blocks obtained at the end of the calculations. This file can be used to restart the computations for additional iterations or as an initial flow field for calculations at slightly different flight conditions.

- **Surface data graphics file**

The program outputs a file **OUTPUT.OMNI** with the geometry, velocity and pressure distribution on the surface of the aircraft. This file is used as input to the A.M.I. OMNI3D software to visualize the surface geometry, the grid and the pressure distribution on a Silicon Graphics workstation. It is also used as input to the Canadair program STREAMLINE which calculates the streamlines on the surface of the aircraft, performs single-pass boundary layer calculations along these streamlines and determines possible locations of flow separation. This file is also used in conjunction with the block topology information to obtain interpolated chordwise pressure distributions at any spanwise station along the wing. These pressure distributions can then be compared with experimental data on line plots.

- **Field data graphics file**

The program outputs a file **OUTPUT.SSV** with the (x,y,z) coordinates of the grid points and the values of the conservative variables at every node of every block. This file is used as an input to the Sterling Software SSV-4D program to visualize the geometry, grid and field aerodynamic characteristics on a Silicon Graphics workstation. It includes also topological information to allow the visualization software to predict flow field streamlines across block boundaries. It is also used as input to the Canadair program OFFLINE which calculates off-body streamlines around the aircraft.

CHAPTER 7 : Applications

7.1 VALIDATION OF THE MULTI-BLOCK PROCEDURES

To minimize the CPU and memory requirements of the three-dimensional Euler code during its development, a generic test case was designed, with topology characteristics similar to those of a typical wing configuration but sufficiently simple to guide the development of the code. The testing of the program was carried out in steps designed to address concerns specific to various stages of the code development. A simple grid was generated around an isolated wing with the geometry used for a wing-body configuration tested by NACA [107]. The wing has a symmetrical NACA 65A006 airfoil section, 45° sweepback an aspect ratio 4 and a taper ratio 0.6. The domain around the wing was divided in 12 blocks as shown in figure 34. The first six blocks are within the wing span and the remaining six blocks outboard of the wing. Blocks 1 to 3 are located below the wing mean plane and blocks 4 to 6 above the wing mean plane. The wing upper surface corresponds to the entire face 3 of block 5 and the wing lower surface to the entire face 4 of block 2. The testing was carried out in three steps :

- Step 1 : 3-blocks test case

Objectives:

- Validate the H-H grid flow solver and the boundary conditions.
- Verify the interpretation of the topology data.
- Check the multi-block data storage in memory.
- Verify the interface boundary conditions in the **i** direction.

A test case was designed using the inboard top three blocks of the 12 blocks grid. The three blocks were numbered 1 to 3 in the downstream direction (figure 35). Relatively coarse H-H grids were generated inside each block : block 1 had ($9\times12\times6$) grid points, block 2 ($17\times12\times6$) grid points and block 3 ($9\times12\times6$) grid points. Inflow boundary conditions were applied on face 1 of block 1 and outflow boundaries on face 2 of block 3. Solid surface boundary conditions were applied on the bottom faces of blocks 1, 2 and 3 and symmetry plane boundary conditions on the inboard faces of the three blocks. The top and outboard faces of the three blocks were constrained to have a side flow boundary condition. The case was run at Mach 0.8 and $\alpha = 0^{\circ}$ and was therefore that of a flow in a channel with a bump on the bottom floor bump in block 2. Figure 36 shows the results obtained after 200 iterations. The residuals converged by five orders of magnitude in all three blocks. In this solution, there is no averaging of the variables calculated at the interface of two adjacent blocks. However, the converged solutions obtained in the two blocks at the interface are for all practical purposes identical. There is an option in the program to force the values of the variables at the block interfaces to be equal to their average. This option guaranties a single value of the flow variables at the interface throughout the computations and was found to give the same solution in the 3-block test case.

Step 2 : 6-blocks test case

Objectives :

- Verify the interpretation of the topology data.
- Verify the symmetry of the solution (wing upper and lower surfaces).
- Verify the interface boundary conditions in the j direction.
- Verify the wing cut boundary conditions.

This test case used the six inboard blocks of the 12 blocks grid, as shown in figure 37. The three upper blocks were the same as in the 3 blocks test case. The three lower blocks had the same number of points as the upper blocks. There was C1 continuity of the grid lines across the block boundaries in both the i and j directions. Inflow boundary conditions were applied to the upstream faces of blocks 1 and 4 and outflow boundary conditions to the downstream faces of blocks 3 and 6. The plane of symmetry condition was applied to the inboard faces of all the blocks and a side boundary condition to the outboard faces of all the blocks as well as on the the bottom faces of blocks 1, 2 and 3 and the top faces of blocks 4,5 and 6. A solid surface boundary condition was applied to the top surface of block 2 (wing lower surface) and to the bottom surface of block 5 (wing upper surface). The interfaces between blocks 1 and 4 in front of the wing and 3 and 6 behind the wing were given the special boundary condition **IBC = 5** (interface/wing cut) because they originate or terminate on a solid surface (wing trailing edge and leading edge). The interface procedure is the same as on all other block interfaces except on points located at the edge of the wing where a special discretization was required. The flow in this case was therefore a flow over a wing constrained by an infinitely large end plate at the tip. A calculation was made at M = 0.84 and $\alpha = 0.0^\circ$ to test the symmetry of the solid surface boundary conditions. The top surface of the wing is a **j =1** boundary plane in block 5 whereas the bottom surface of the wing is a **j = JL** boundary in block 2. The boundary conditions are therefore applied in different sections of the code and it was necessary to verify that the solution was symmetrical. A run was made for 200 iterations, and convergence to five orders of magnitude was obtained. Figure 38 shows the pressure distribution on a cut located around the midspan and figure 39 shows a general view of the pressure distribution obtained. This figure shows that a good level of symmetry is obtained, given the relatively coarse mesh size in the field. The lift coefficient calculated on the wing was $C_L = 0.0013$.

A practical problem arose with this configuration at the leading edge. To impose the boundary condition on the wing surface, the program calculates at every node point on the solid surface the vectors normal and tangent to the surface. The calculation at a given node i is performed numerically using the coordinates of the surrounding node points, notably at **i-1** and **i+1**. In a C-grid wrapping around the airfoil, these **i-1** and **i+1** node points are on the airfoil surface, ensuring continuity of the normal and tangent vectors calculated at points near the leading edge. In a multi-block grid with H topology, the situation is different. When solving for block 5 on top of the wing, the first node point on the airfoil surface is the point at the leading edge. The information on the shape of the airfoil lower surface is not available. As a result, the normal and tangent vectors calculated at the leading edge point in the top and bottom blocks are not the same. The mesh spacing of this test case is very coarse and the curvature of the thin leading edge of this wing is poorly approximated by the straight lines through the first grid points in the top and bottom blocks. To improve the scheme, normal and tangent vectors common to the two blocks were calculated in **SETUP**, using averages of the values calculated in each block. This improved the results but the situation is responsible for the slight discontinuity of the solution at the leading and trailing edge visible in the results presented in this relatively coarse mesh case. Results presented further in the text with refined meshes will confirm that complete continuity of the leading edge pressures was achieved.

- Step 3 : 12-blocks test case.

Objectives :

- Verify the interpretation of the topology data.
- Verify the interface boundary condition in the **k** direction.
- Verify the wing tip cut boundary conditions.

This was a test case with interfaces in the **i**, **j** and **k** directions. The grid lines and slopes were continuous across all block boundaries except for a discontinuity in slope in the **j**-plane boundary between the blocks above and below the wing plane caused by the orthogonality required near the wing surface. The blocks were numbered as shown in figure 34, with inflow conditions applied to the upstream boundaries of blocks 1, 4, 7 and 10, and to the bottom boundaries of blocks 1, 2, 3, 7, 8 and 9 . Outflow conditions were imposed on the downstream faces of blocks 3, 6, 9 and 12 and to the top faces of blocks 4, 5, 6, 10, 11 and 12 and side flow conditions to the outboard faces of blocks 7 to 12. The special boundary condition **IBC = 5** (interface/wing cut) was applied all the interfaces in the **j** plane surrounding the wing mean plane : top faces of blocks 1, 3, 7, 8 and 9 and bottom faces of blocks 4, 6, 10, 11 and 12. Calculations were made at Mach 0.94 and $\alpha = 3^\circ$. Figure 40 shows the top surface pressure distributions. The flow is supersonic on the upper surface. However, with only 17 points along the chord, the mesh is too coarse to capture any existing shock wave. The coarseness of the mesh is also responsible for opened pressure distributions at the leading and trailing edges as discussed earlier. Several modifications were made to the code during this testing phase. The final results of the test indicated that the formulation of the multi-block code was working properly for a wing configuration. The next step was to test the code using more refined grids and produce results with engineering accuracy. This was done using the ONERA M6 wing geometry.

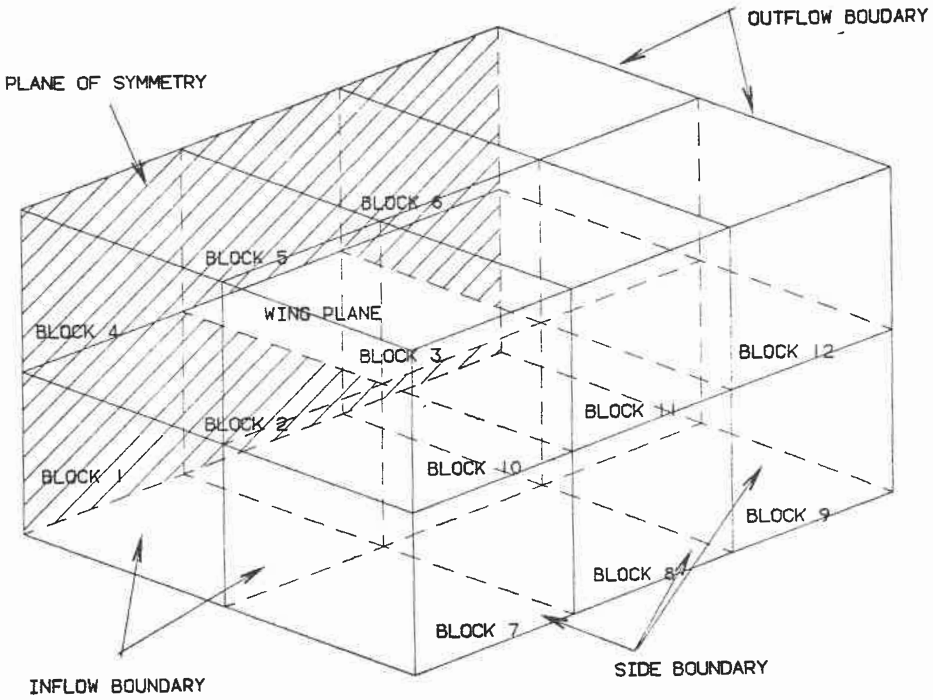


Figure 34 : Topological block decomposition of the space around an isolated wing used in the 12-block test case.

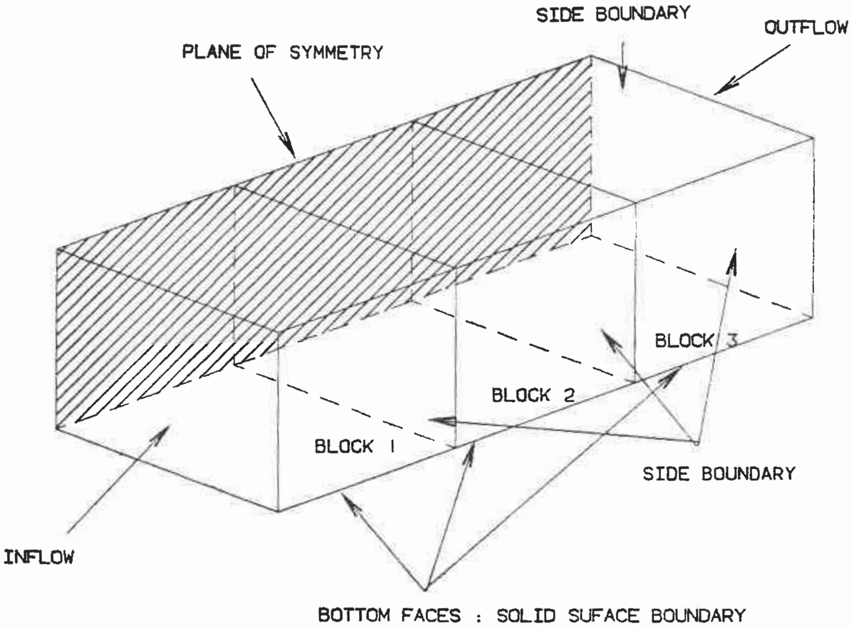


Figure 35 : Topology definition of the 3-block test case.

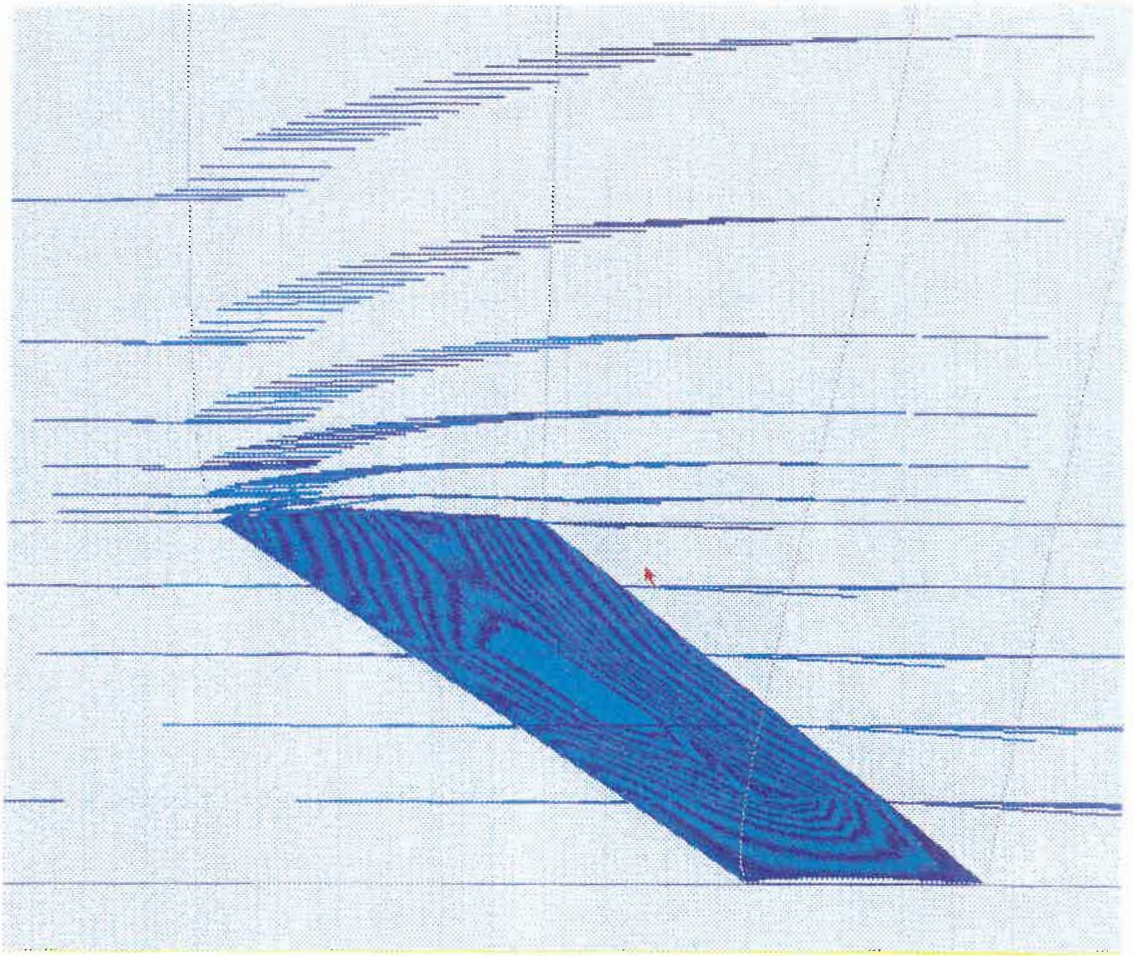


Figure 36 :

Isobars on the floor bump (j=1) and velocity vectors calculated in. the plane of symmetry (k=1) and upstream and downstream of the floor bump. 3-block test case, M = 0.80 and $\alpha = 0^\circ$.

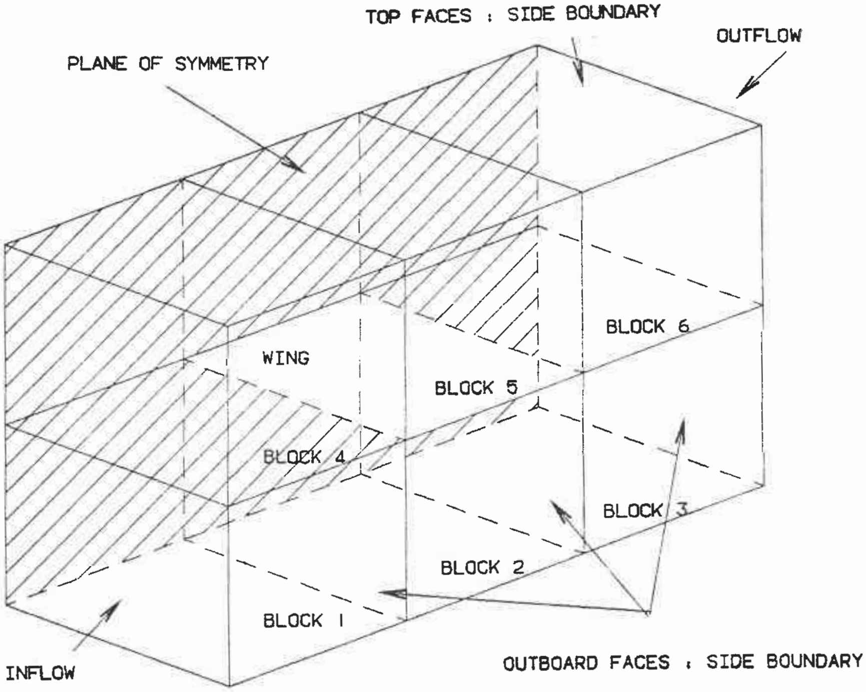


Figure 37 : Topology definition of the 6-block test case.

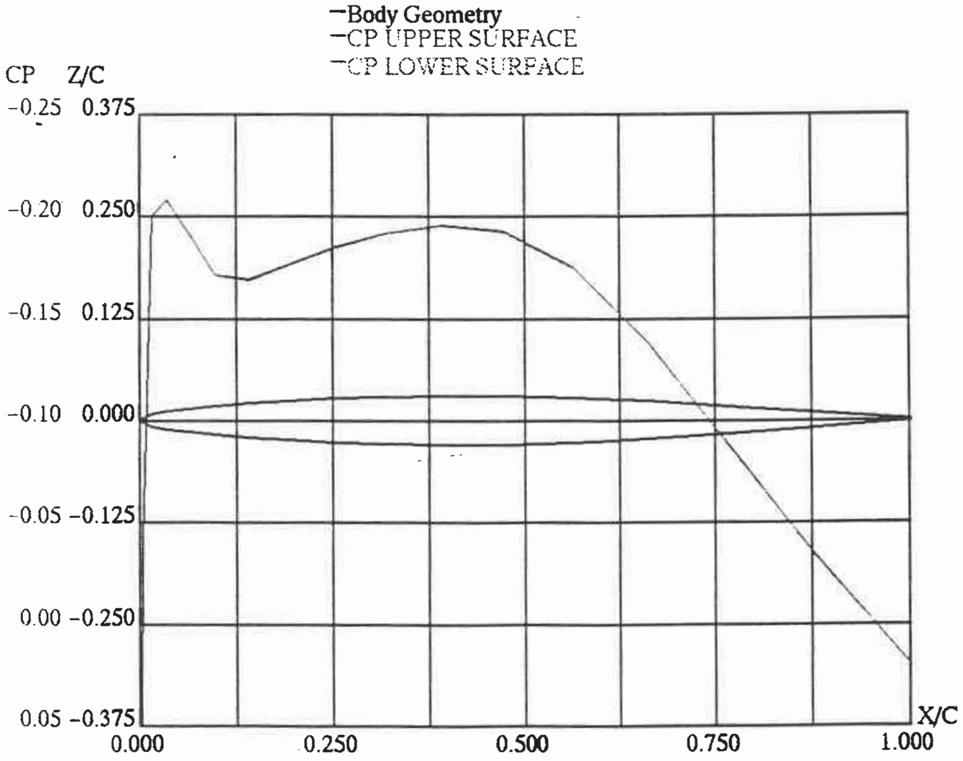


Figure 38: Sectional pressure distribution calculated at midspan on the NACA L51F07 wing at $M = 0.84$ and $\alpha = 0^{\circ}$ using 6 blocks.

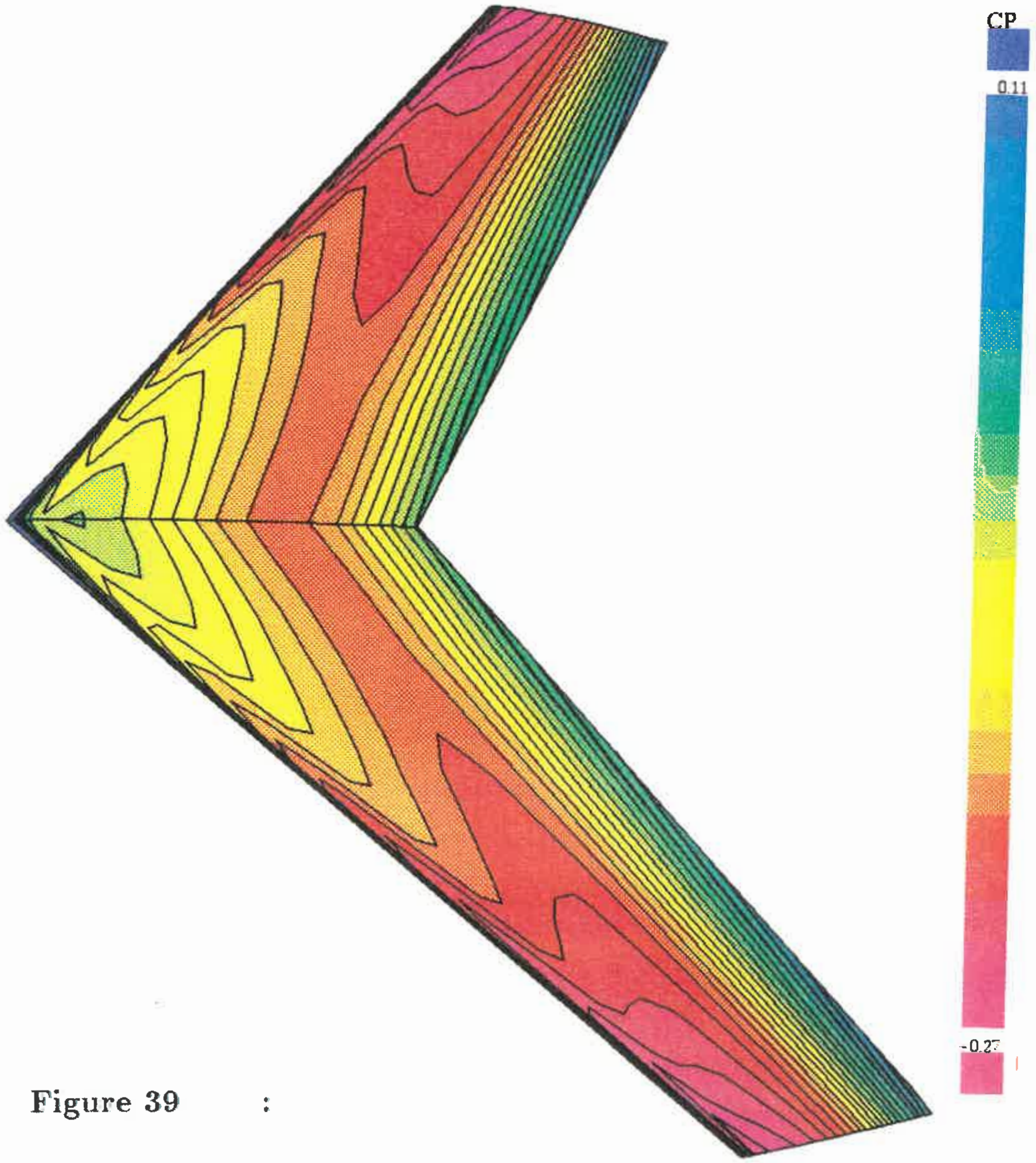


Figure 39 :

Upper surface pressure distribution calculated on the NACA L51F07 wing at M = 0.84 and $\alpha = 0^{\circ}$ using 6 blocks.

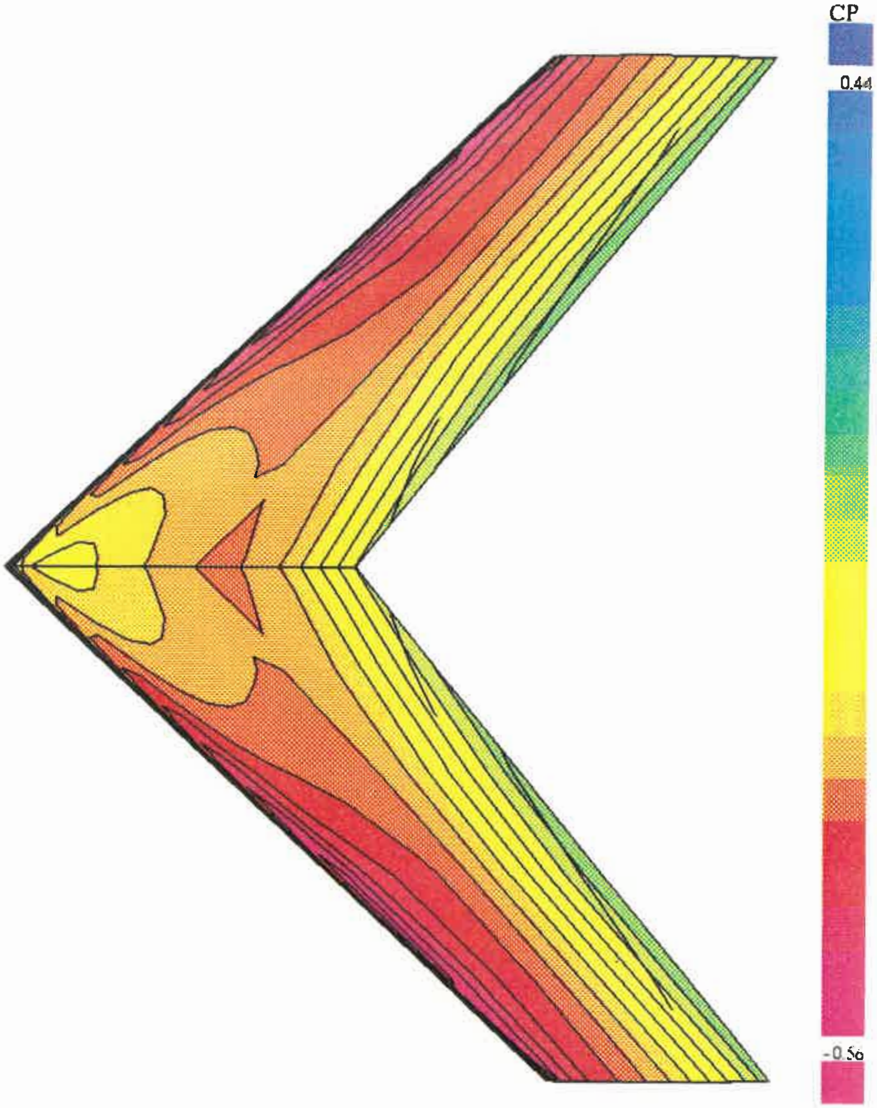


Figure 40 :

Upper surface pressure distribution calculated on the NACA L51F07 wing at M = 0.94 and $\alpha = 3^\circ$ using 12 blocks.

7.2 ONERA M6 WING TEST CASE

The multi-block Euler code was also tested using the ONERA M6 wing because of the availability of reliable surface pressure data [108]. The wing has an aspect ratio 3.8, a taper ratio 0.562, 30single symmetrical section nearly 10% thick. A grid with an H-H topology made of 12 blocks was generated for this wing. The 6 inboard blocks cover the span between the plane of symmetry and the wing tip. The other 6 blocks are located beyond the wing tip. The blocks were numbered as shown in figure 41. Initial tests were carried out using only the three bottom blocks, using solid surface boundary conditions on the top faces and side boundary conditions on the side and bottom faces. These tests were made to check the multi-block H-H solver, the correct selection of "face neighbours" and validate the block internal face boundary conditions in the i-direction. In a second step, the top three blocks 4, 5 and 6 were added to verify the selection of "edge neighbours" and the application of the interface boundary conditions in the i and j directions simultaneously. This case was also used to verify the symmetry of boundary conditions applied on the top of block 2 and the bottom of block 5. The evolution of the residuals in each of the six blocks during the computations and the evolution of the number of supersonic points in block 5 are shown in figure 42 for calculations performed at Mach 0.9 and = 0 on a grid with 181,566 node points. After 500 iterations, the residuals have been reduced by five to six orders of magnitude. The number of supersonic points in block 5 does not change after 350 iterations. Finally, calculations were made using the 12 blocks to verify the selection of "corner neighbours" and the application of interface boundary conditions in the three spatial directions i, j and k.

A calculation was made for the ONERA M6, wing using the twelve previously defined blocks, at Mach 0.84 and = 3.06 with a 12-block grid containing a total of 293,964 node points. The planform pressure distribution obtained after 500 iterations is shown in figure 43. It has the λ shock structure, with a weak highly swept forward shock followed by a stronger, nearly straight aft shock. The two shocks merge into a single one near the wing tip. A comparison of the pressure distributions with experimental data measured at Re = 11.7×10^6 [108] is shown on figure 44.

Calculations were also made on the ONERA M6 wing using Jameson's single-block FLO67 program with a similar number of chordwise and spanwise stations. A comparison of these results with **MBTEC** multi-block results, shown in in figure 45, indicates similar accuracy. The differences seen can be attributed to the codes different grid topologies : C-H grid topology for FLO-67, H-H grid topology for the multi-block code. At the wing tip ($\eta = 0.96$), the **MBTEC** results are actually closer to the experimental data. This is thought to be due to a better grid configuration off the wing tip in the multi-block model.

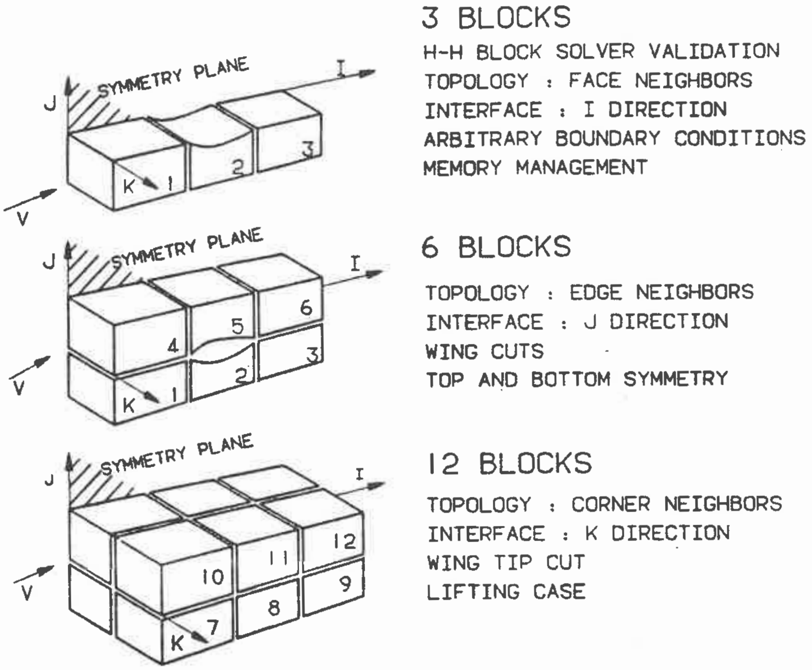


Figure 41 : Test cases used to validated various parts of the multi-block Euler code.

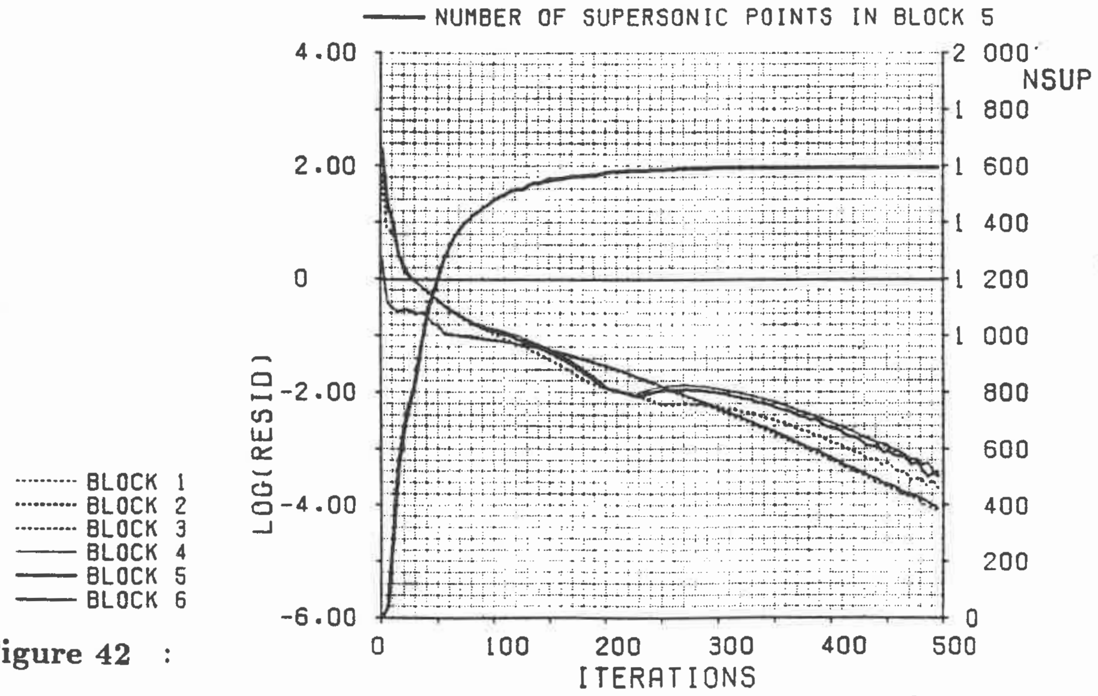


Figure 42 :

Convergence characteristics of the multi-block Euler code. Evolution of the maximum residual and the number of supersonic points. MBTEC calculations on ONERA M6 wing M = 0.90 and $\alpha = 0^\circ$, (6 blocks).

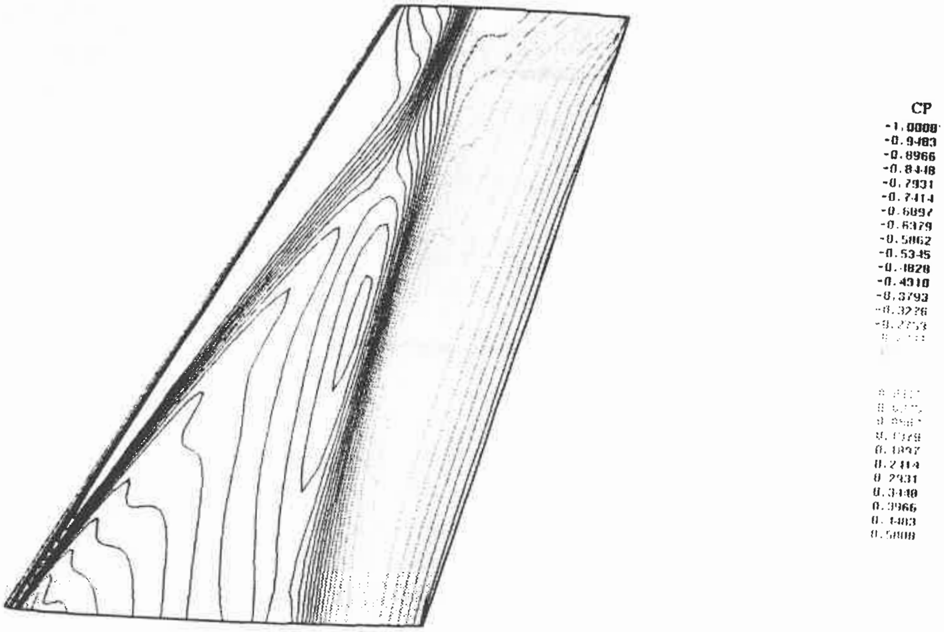


Figure 43 : Planform isobars for the ONERA M6 wing. M = 0.84 $\alpha = 3.06\ ^\circ$.

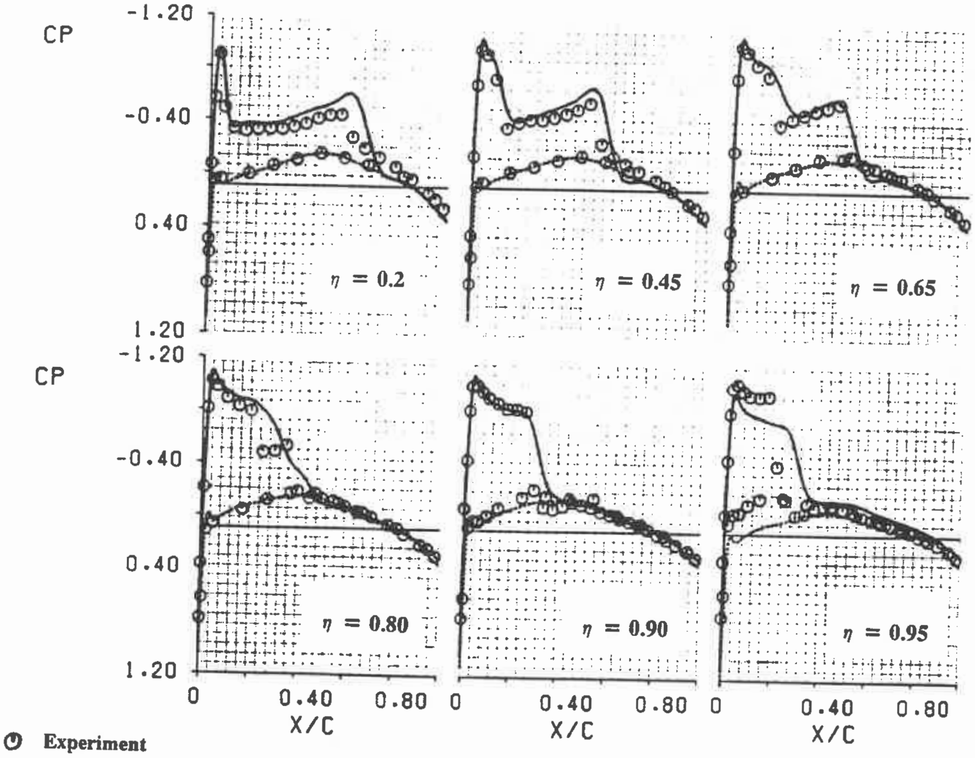


Figure 44 : Pressure distributions for the ONERA M6 wing. M = 0.84 $\alpha = 3.06\ ^\circ$.

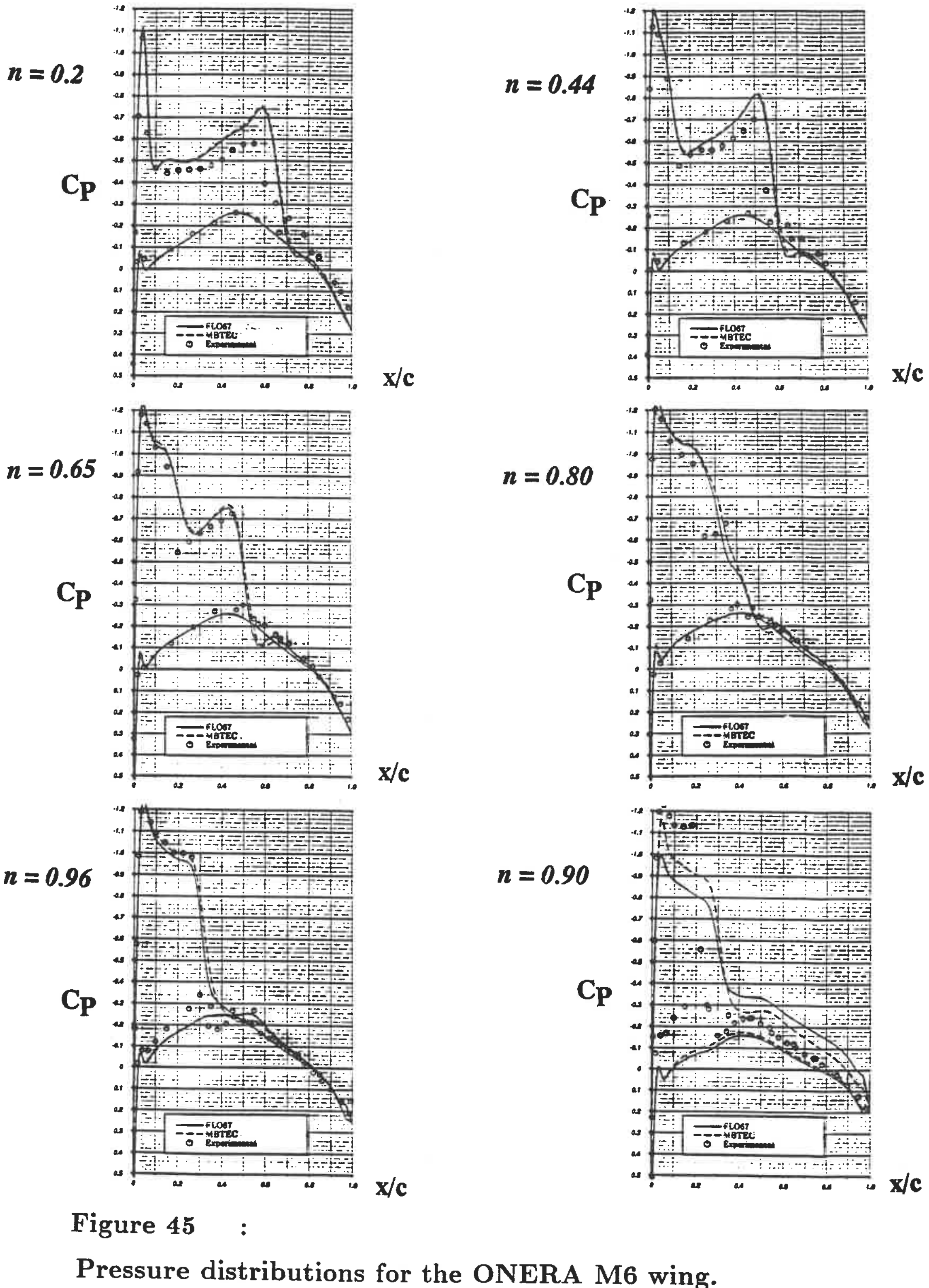


Figure 45 :

Pressure distributions for the ONERA M6 wing. M = 0.84 $\alpha = 3.06\,^{\circ}$. Comparison of single-block FLO67 results with MBTEC 12-block results.

7.3 CHALLENGER FORWARD FUSELAGE TEST CASE

The second calculation to be carried out with more refined grids was performed on the Challenger forward fuselage, shown in figure 46. The grid was made of 8 blocks. Four blocks were located upstream of the fuselage. The fuselage solid surface is represented by the inboard faces of blocks 6 and 7. This case presented two special problems :

- First, the inboard faces of blocks 2 and 3 are degenerated as shown on figure 47. All the grid points on these faces are collapsed into a single line. The points on the upstream grid lines of the inboard faces of blocks 6 and 7 are also collapsed into the point at the fuselage nose. The variables at the grid points on these collapsed boundaries are calculated for each grid point independently, but in order to guarantee that they converge towards the same values related to their unique physical location, an averaging of the residuals calculated at these grid points was done. A special procedure for the calculation of the normals to these collapsed boundaries was implemented. Several tests were also included in the code to prevent operation overflows due to the zero areas of the cell faces on these boundaries.
- Secondly, the intersection between the fuselage and the plane of symmetry in this grid configuration is between the inboard faces of blocks 7 and 8 above the fuselage and between the inboard faces of blocks 5 and 6 on the lower surface. The code was written to apply special boundary conditions at the intersection between two solid surfaces or between a solid surface and a plane of symmetry when both these intersecting surfaces belong to the same block. To accommodate this particular grid topology, the multi-block code had to be reprogrammed to allow also for the specification of exact boundary conditions at intersections between surfaces modelled in different blocks.

The number of grid points in the **i**, **j** and **k** directions in this test case is as follows :

Block 1 : 17×17×21

Block 2 : 17×15×21

Block 3 : 17×25×21

Block 4 : 17×17×21

Block 5 : 29×17×21

Block 6 : 29×15×21

Block 7 : 29×25×21

Block 8 : 29×17×21

The domain was thus covered with 71,484 grid points. Inflow boundary conditions were specified at all the upstream faces of blocks 1 to 4, at the top faces of blocks 4 and 8 and at the bottom faces of blocks 1 and 5. The upper and lower far field boundaries were set at such an angle that for most angles of attack of interest, they have inflow boundary conditions. Outflow boundary conditions were applied at the downstream planes of faces 5 to 8. Side flow boundary conditions were applied to the outboard faces of all eight blocks, solid surface boundary conditions on the inboard faces of blocks 6 and 7 and symmetry boundary conditions at the inboard faces of all the other blocks. Figure 48 shows the surface pressure distribution obtained on the fuselage surface at Mach 0.82 and alpha=1.5 degrees after 100 iterations. The velocity vectors calculated in the plane of symmetry around the fuselage are shown in figure 49. The pressure distribution on the top and bottom fuselage lines are shown in figure 50. This test indicated that the multi-block code was behaving properly for isolated body configurations. The next step was the calculation for more realistic aircraft configurations.

7.4 CHALLENGER WING/BODY/WINGLET TEST CASE

The next results are for a Challenger wing/body/winglet symmetrical configuration at Mach 0.82 and = 1.5. The grid has 40 blocks and 550,000 node points. The block above the wing and winglet has 65 chordwise grid points and 33 spanwise grid points, 7 of them over the winglet span. The calculations were made using 100, 100 and 500 iterations in the coarse grid, intermediate grid and fine grid respectively. The pressure distribution on the aircraft surface is shown in figure 51 in the form of isobars displayed by the OMNI3D graphics software on a workstation screen. These results in inviscid flow indicate the presence of a strong shock on the wing upper surface, continuing on the winglet. The block edges on the aircraft surface are superimposed showing that a continuous pressure distribution is obtained across the block boundaries. The longitudinal C_P distribution on the top and bottom centerlines of the fuselage are given in figure 52(a). The pressure distribution at the wing break station is shown in figure 52(b). The pressure distribution at a winglet mid-span section is shown in figure 52(c). The calculated wing/winglet lift coefficient is $C_L = 0.71$.

7.5 CF-34 TRANSONIC NACELLE TEST CASE

The next calculations were performed on the Challenger CF-34 engine nacelle using a grid with 11 blocks. A grid was generated for the nacelle, positioned in the same location as it would be on the complete CL-601 aircraft. The engine model has the 2 degrees toe-out angle and 1.5 degrees incidence it has on the aircraft. The complete space around the nacelle, including the inlet and exhaust ducts, was covered with a grid centered on the engine longitudinal axis (figure 53). Because of the canted attitude of the nacelle, the calculation for the CF-34 nacelle at $M = 0.74$ and $\alpha = 0$ (measured with respect to the aircraft centerline) is the first Euler solution at Canadair for a non-symmetrical configuration in yawed flow. Calculations were made for an inlet mass flow ratio $MFR = 0.6838$ and for two different exhaust flow conditions :

		Case (1)	Case (2)
Fan Total Temperature Ratio	:	1.25	1.13
Fan Total Pressure Ratio	:	2.10	1.46
Fan Exhaust Mach Number	:	0.49	0.49
Core Total Temperature Ratio	:	3.20	2.88
Core Total Pressure Ratio	:	1.60	1.11
Core Exhaust Mach Number	:	0.35	0.35

The pressure distributions obtained for case (1) are shown on figure 54. The flow in the inlet duct is smoothly decelerated. For the inlet mass flow ratio specified, the peak velocity on the outside of the nacelle is low and the pressure is fairly constant on a large part of the fan cowl. For this high fan exhaust pressure ratio, the static pressure at the fan nozzle exit plane, where the flow is locally sonic, is much higher than the freestream static pressure. Since the static pressure near the fan cowl trailing edge is only slightly higher than freestream static pressure, the fan exhaust flow is underexpanded at the fan nozzle exit plane. The flow then expands rapidly. As it moves downstream, the curvature effect of the core cowl allows further expansion. The flow goes through several expansion-compression cycles of reducing amplitude. The presence of shock diamonds in the solution is further evidenced in the OMNI3D surface pressure distribution in figure 55. This figure shows also that the pressure contour are continuous across the shear layers surrounding both exhaust jets. The plot in figure 56 gives the corresponding pressure distribution for case (2) with lower total pressures and temperatures at the fan and core exhaust planes. The flow is still sonic at the fan duct exit plane but the amplitudes of the expansion-compression cycles are smaller. Figure 57 gives the Mach number distribution in a planar cut across the engine as visualized on the workstation using the Sterling Software SSV-4D program. The first Euler code results on the turbofan engine prompted many questions which will be addressed as comparisons with experimental data become available. As shown in reference [79], the inlet duct, fan and core cowl pressure distributions predicted by the Euler code should compare favorably with test data provided a sufficient number of grid points is used. However, the implications of the exhaust plume modelling and the influence of grid fineness and the artificial viscosity used in the code on the shear boundary between regions with different total pressures and total enthalpy are yet to be fully determined.

7.6 CHALLENGER WING-BODY CONFIGURATION

Additional runs were made on the Challenger wing-body configuration, in order to compare the calculated pressure distributions with experimental results obtained during high speed wind tunnel tests. The computational model was constructed with 40 blocks and a fine grid generated with 405,688 nodes. A view of the grid at the surface of the configuration was shown is figures 58 (a) and (b). The **MBTEC** model has a square wing tip instead of the round shape found on the aircraft. Results are shown here for calculations made at Mach 0.819 and for an angle of incidence $\alpha = 1.494^{\circ}$. The computations were made on a sequence of three grids : 100 coarse grid iterations , 100 intermediate grid iterations and 500 fine grid iterations . The total CPU time on the CONVEX C220 computer was 11 hours, 52 minutes and 36 seconds. The program required 54.5 Megabytes of memory to run this case. The surface pressure distribution obtained on the surface of the configuration is shown in figure 59, indicating a well developed shock wave far aft on the wing chord. The inviscid lift coefficient calculated is $C_L = 0.7487$. A comparison of these pressure distributions with experimental data obtained in the tunnel at a Reynolds number Re = 3.8 million is shown on figures 60 (a) to (f) for six stations on the wing span. Theoretical results obtained with the KTRAN Transonic Small Disturbance code previously developed at Canadair [11] are also shown on these figures. The Challenger has aft loaded supercritical wing sections which are prone to substantial viscous effects. As expected, the inviscid theoretical results display stronger shock waves located further aft on the chord. The pressure distributions in the cove region, on the lower surface, also show a larger recompression than experimental results. The coupling to a boundary layer is expected to bring these results closer to the experimental data. Calculations by Mokhtarian [109] with a single-block Euler/Boundary layer code at the same flow conditions indicated that, at tunnel Reynolds number, the shock is 10% chord further forward with viscous corrections applied (figure 61). This correlates very well with the differences seen in figure 60 between inviscid theory and experimental data. The results in figure 60 also indicate that the Euler results capture more accurately the upper surface pressure plateau than the KTRAN results. The lower surface pressure distributions are also closer to the experimental data. The **MBTEC** results are especially better near the wing tip ($\eta = 0.85$), where the inviscid shock wave remains aft of the experimental location.

7.7 CHALLENGER COMPLETE CONFIGURATION

The last application for the Challenger was the modelling of the flow around the complete aircraft configuration, including a wing-body-winglet, as studied previously, fuselage mounted CF-34 short cowl turbofan nacelles and nacelle pylons and a complete empennage with a horizontal tail mounted in a T-tail fashion at the top of the fin. The particular topology of the aircraft required a large number of blocks. The grid was constructed with 600 blocks organized into 16 groups. The outline of the block edges on the surface of the aircraft are shown in figures 62(a) and 62(b). The fine grid generated for this configuration contained 1,571,580 nodes. A view of this grid is shown in figure 63. Because of limitations of memory and computing time on the CONVEX C-220, the solutions presented here were calculated using an intermediate level grid, which contained 242,994 nodes. A view of this grid on the aircraft surface was shown on figures 64(a) and 64(b). Calculations were made at Mach 0.799 and $\alpha = 0.93\ ^\circ$, for comparison with available flight test data. The engine inlet and exhaust conditions are identical to those of case (2) of the isolated nacelle test case. The surface isobars obtained after 500 iterations on the intermediate grid are shown in figures 65(a) and 65(b). Of interest is the detailed flow field captured by the program in the nacelle and pylon areas. An additional top view of these results is given in figure 66 to indicate that the location of the block edges is transparent to the solution. A comparison of these pressure distributions with flight test data is shown for six wing stations on figures 67 (a) to (f). The theoretical inviscid results are seen to compare fairly well with the flight test results, obtained at a mean chord Reynolds number of about 12 million. The program calculates a complete aircraft $C_L = 0.4368$. The pressure distribution calculated on the fuselage top and bottom centerlines is shown in figure 68. The C_p distribution on the horizontal tailplane at 45% semi-span is shown on figure 69. The tailplane is seen to be downlifting. Figure 70 shows a cut of the nacelle pylon at half-span between the fuselage side and the fan cowl. In figure 71, pressure distributions of the nacelle top and bottom centerline are shown along with the wing pressure distribution at the same spanwise station (y = 88.0). A closer view of the nacelle pressures is shown on figure 72. These results were used to predict streamline patterns on the aircraft surface, as shown on figure 73. The ability to predict such streamlines and to carry out streamline boundary layer calculations is critical in the design of wing/body fairings, wing/winglet junctions and aft fuselage shapes of airplanes. Additional calculations were made for comparison with wind tunnel test data. The runs were made at Mach 0.76, which is a typical long range cruise Mach number on the aircraft, and at three different angles of incidence. The lift coefficients obtained for the complete configuration after 400 iterations are shown in table 3 in comparison with wind the experimental values.

Table 3 : CL-601 complete configuration results at Mach 0.76

MACH	**Alpha**	**C_L (MBTEC)**	**C_L TWT-433**
0.7607	2.103°	0.4986	0.408
0.7602	2.775°	0.5933	0.516
0.7611	3.366°	0.6719	0.616

A comparison of the pressure distributions obtained at $\alpha = 2.10^{\circ}$ with wind tunnel measurements is shown in figure 74, for six spanwise stations. A finer grid is probably required to capture the shock wave as sharply as on the wing-body fine grid case in figure 60. However, the agreement is slightly better than the the comparison with flight test data shown in figure 67 for a higher cruise Mach number. The pressure distributions calculated at two winglet stations are shown in figure 75 in comparison with experimental data. The resulting (C_L, α) curve is shown in figure 76, indicating that the theoretical results overestimate the lift, as expected from an inviscid solution. The lift curve slope is slightly underestimated. The wind tunnel test included measurements of pressure distributions at longitudinal stations on the rear fuselage located above and below the nacelle pylon. A comparison of these pressures with calculated results is given at two radial locations on the fuselage in figure 77. The program predicts fairly well the general level of the pressures, despite the existence of a fairly thick boundary layer this far on the fuselage. There were no nacelle pylon pressure measurements available for this particular test condition. Figure 78 gives the predicted pressure distribution near the pylon centerline. This figure shows that the pylon has positive lift near the leading edge and negative lift on the rear, resulting in a near zero lift for the complete pylon, a desired design characteristic. During execution, the program monitors the evolution of the maximum and average residuals inside each one of the 600 blocks. The convergence is the slowest in the blocks above the wing, where the flow is supersonic. Figure 79 (a) shows the convergence characteristics of the average residual in five blocks located above the wing, over 400 iterations, for calculations at Mach 0.76 and $\alpha = 2.10^\circ$. The calculations were made for 200 iterations first, then restarted for 200 more iterations. This is responsible for the peak at 200 iterations. However, the curves return quickly to the overall trend, indicating between four and five orders of magnitude reduction in the residual. Figure 79(b) gives the evolution of the number of supersonic points in the same five blocks. There is no discontinuity at 200 iterations due to the restart process and the curves are converged in most of the blocks after 400 iterations. The convergence characteristics of the **MBTEC** program for a 600-block full aircraft case, is similar to that of a 12-block isolated wing shown earlier. A multi-grid procedure, to be implemented in the near future, is expected to accelerate the convergence substantially. Since such a procedure is best suited to solutions with few large grids, a grouping of small blocks will probably be necessary when making calculations on multiple grid levels. After completing the evaluation of the code for the full CL-601 configuration, additional calculations were made for the wing-body geometry, over a range of Mach numbers in order to investigate the ability of the code to predict the evolution of compressibility drag. The drag rise curve obtained for C_L =0.6 is shown in figure 80. There are no scales shown to protect the confidentiality of the data. Although the program overpredicts the lift for a given angle of incidence, the lift-drag relationships are much closer to experimental values. This was confirmed by more recent theoretical and experimental results obtained on Canadair's new Global Express configuration. These are the results of the very first theoretical calculations made at Canadair with a transonic flow code on the *complete* Challenger configuration. These results demonstrate that the **MBTEC** program can produce transonic flow information on the aircraft surface pressure distribution with a level of detail similar or superior to the one obtained with sophisticated 3D panel methods for subsonic flow. In addition, the **MBTEC** field code produces automatically information of the complete flow field surrounding the aircraft and not just on the aircraft surface. Understanding the flow structure in the space around the aircraft is a necessary step in the design of new configurations.

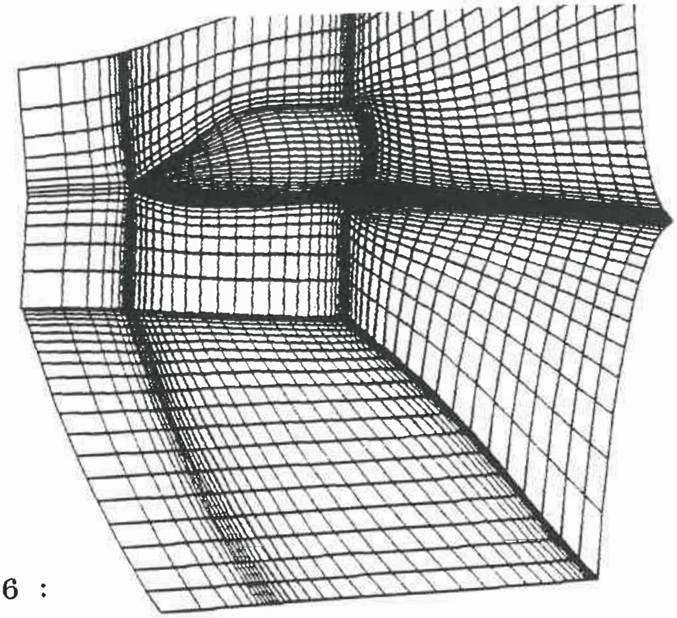


Figure 46 :

View of the grid used to compute solutions on the Challenger forward fuselage.

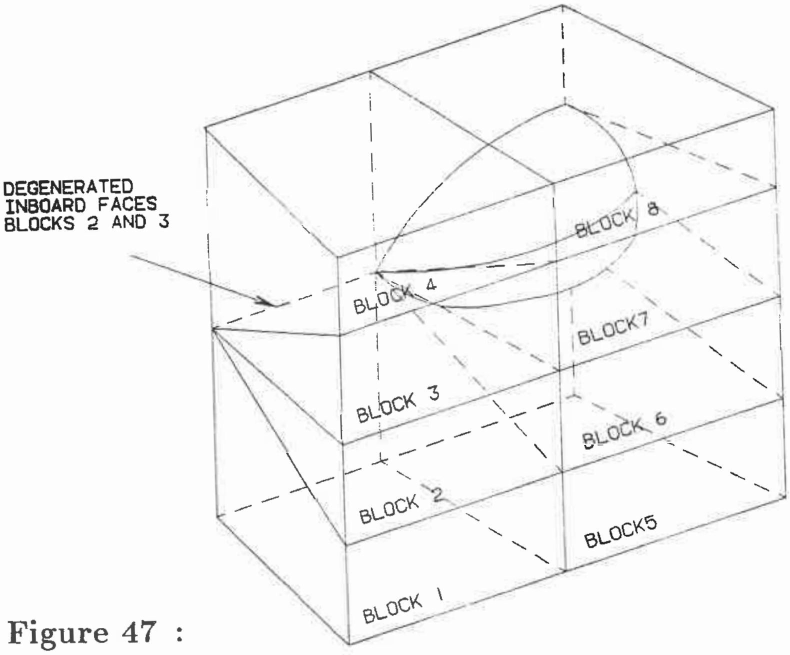


Figure 47 :

Topology definition of the Challenger forward fuselage grid using 8 blocks. The inboard faces of blocks 2 and 3 have degenerated to single lines in the plane of symmetry.

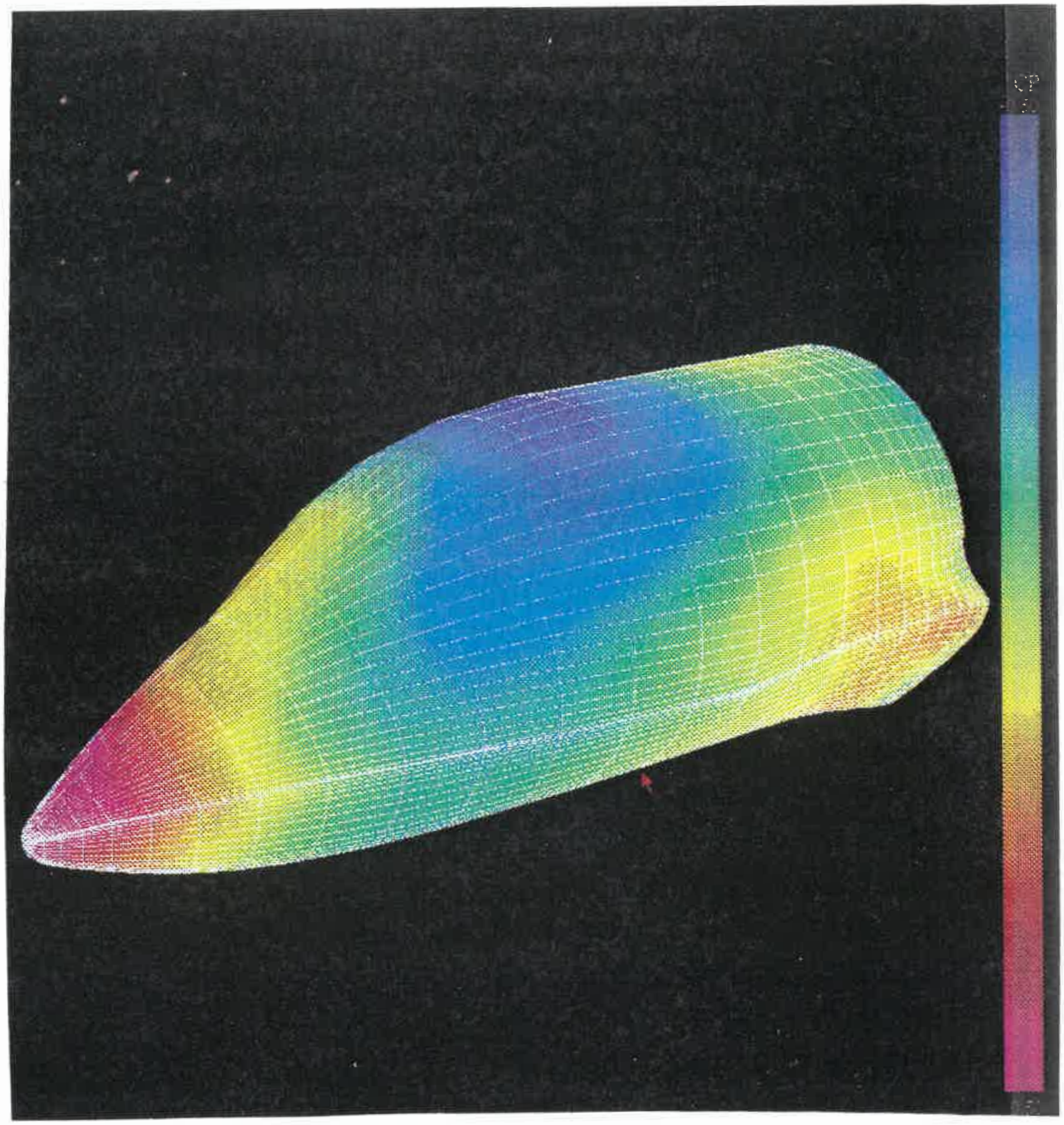


Figure 48 :

Surface pressure distribution calculated on the Challenger forward fuselage at $M = 0.82$ $\alpha = 1.5^\circ$ using 8 blocks.

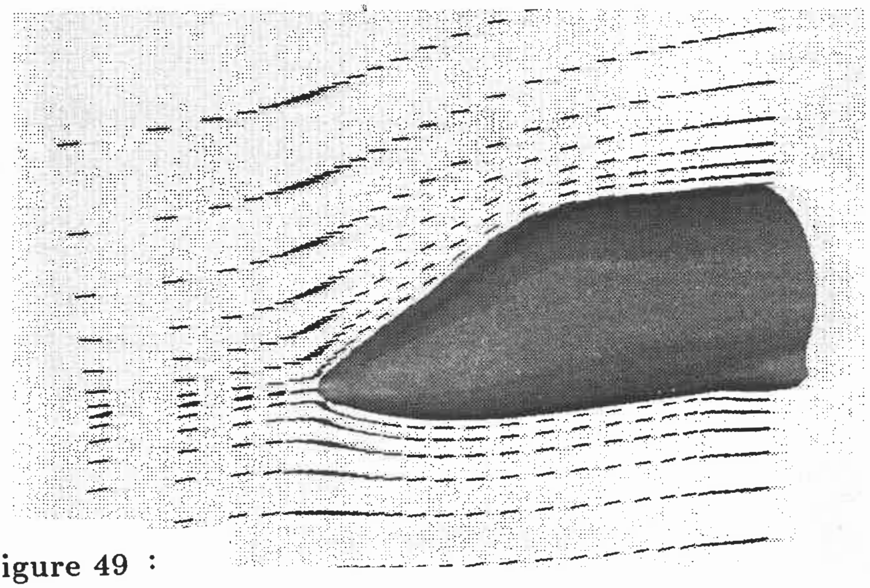


Figure 49 :

Velocity distribution calculated in the plane of symmetry for the Challenger forward fuselage at M = 0.82 $\alpha = 1.5°$ using 8 blocks.

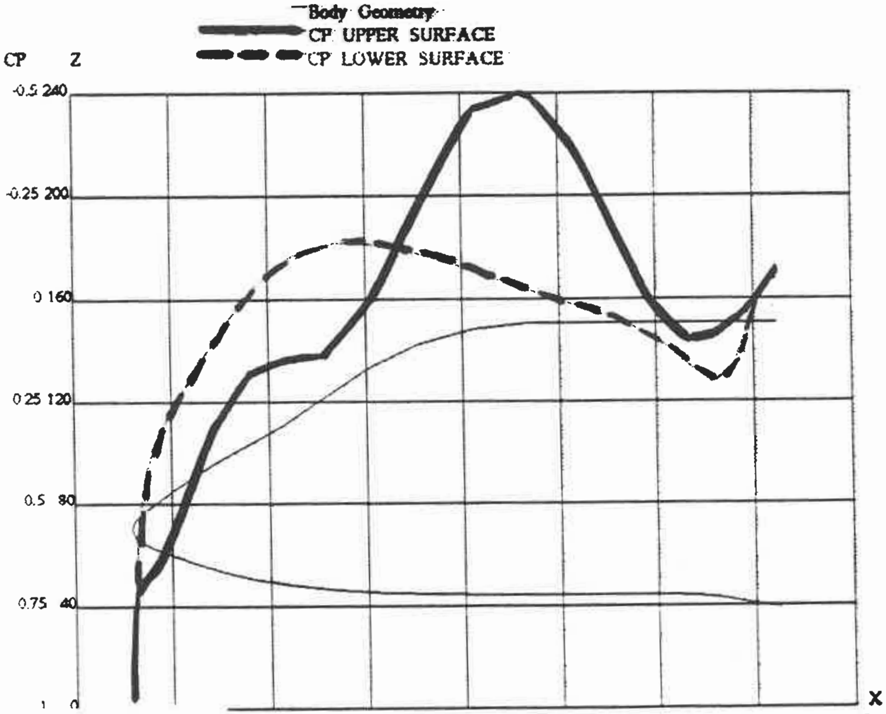


Figure 50 :

Pressure distribution calculated in the plane of symmetry for the Challenger forward fuselage at M = 0.82 $\alpha = 1.5°$ using 8 blocks.

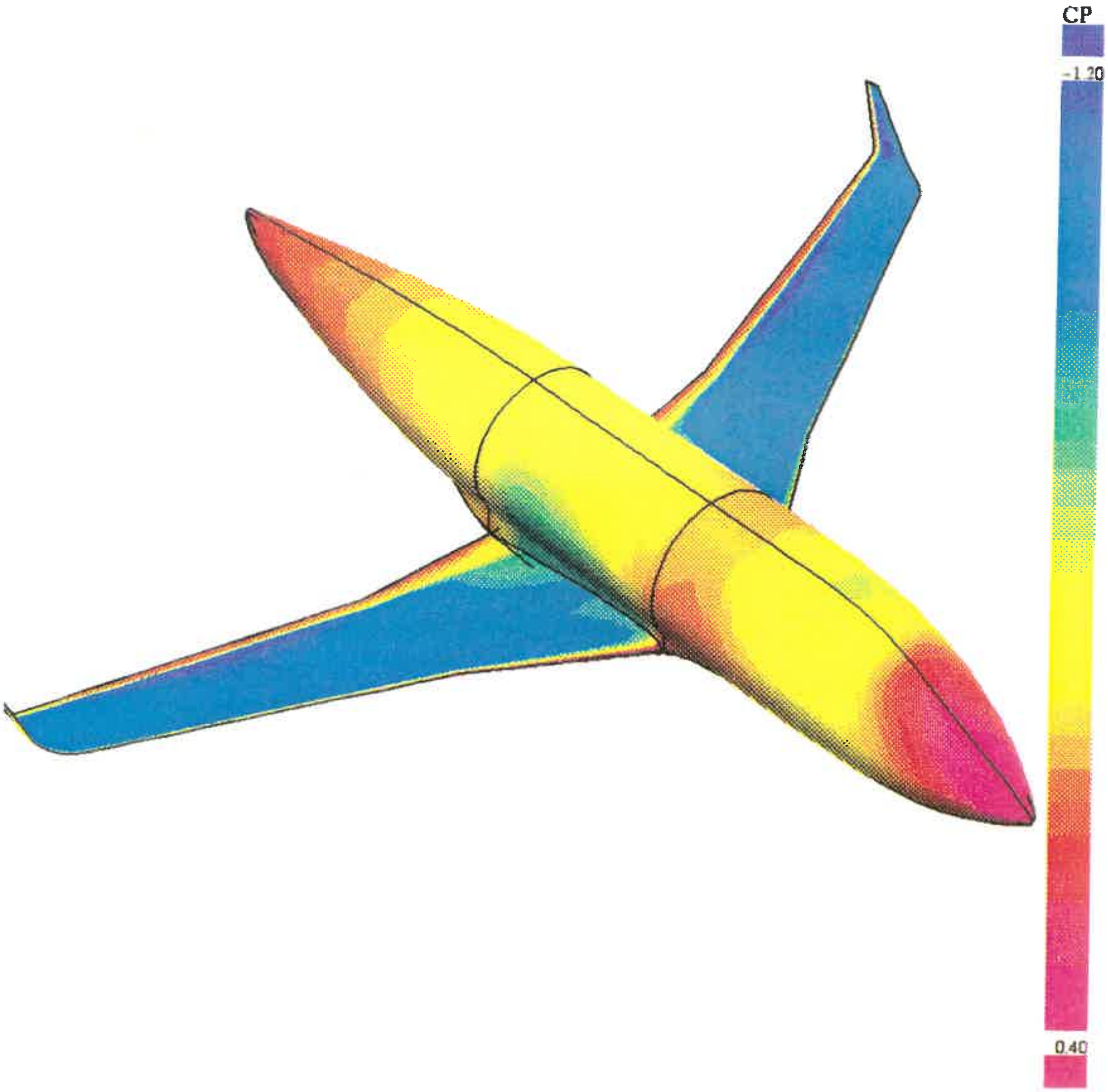
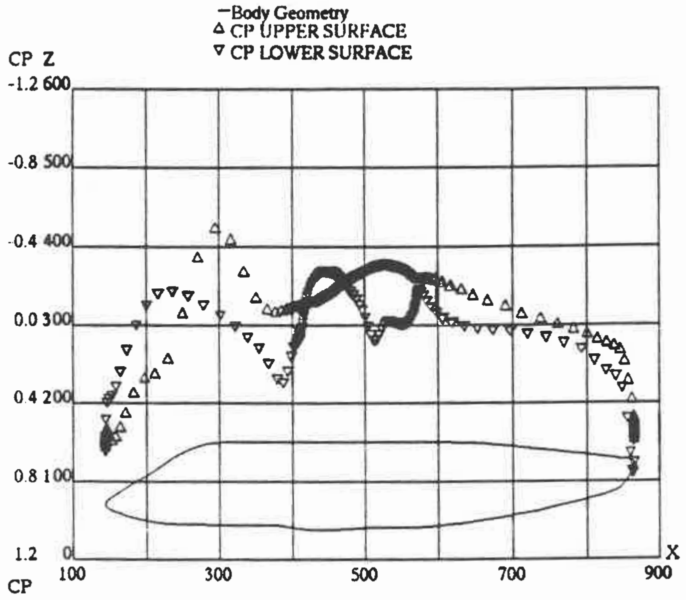
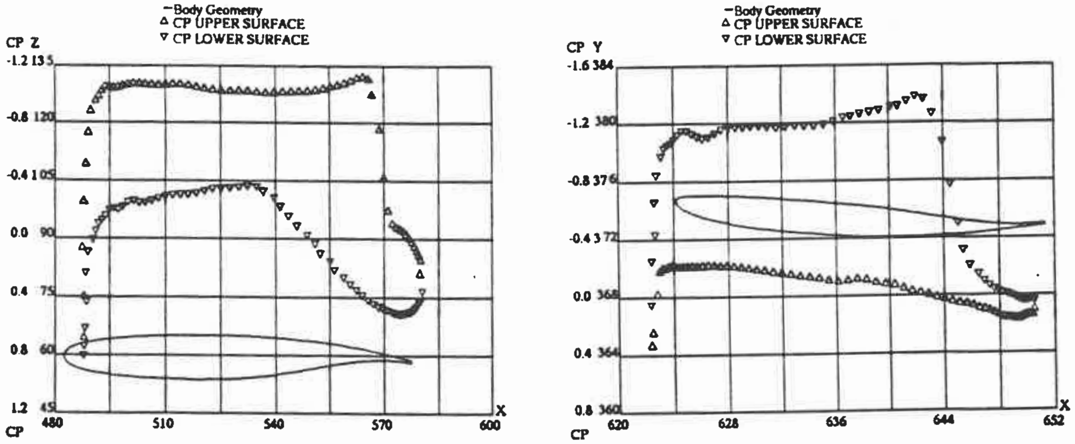


Figure 51 :

Surface isobars for the Challenger wing/body/winglet at $M = 0.82$ $\alpha = 1.5\ ^{\circ}$.
40-block MBTEC solution (100 + 100 + 500 iterations).



(a) BUTTLINE CUT Y- 0.00



(b) BUTTLINE CUT Y- 148.40

(c) WATERLINE CUT Z- 80.00

Figure 52 :

Pressure distributions calculated on the Challenger wing/body/winglet configuration at M = 0.82 $\alpha = 1.5$ °. 40-block MBTEC solution (100 + 100 + 500 iterations).
(a) Fuselage top and bottom centerlines;
(b) wing break section, station 148;
(c) Winglet mid-span station, waterline 80.

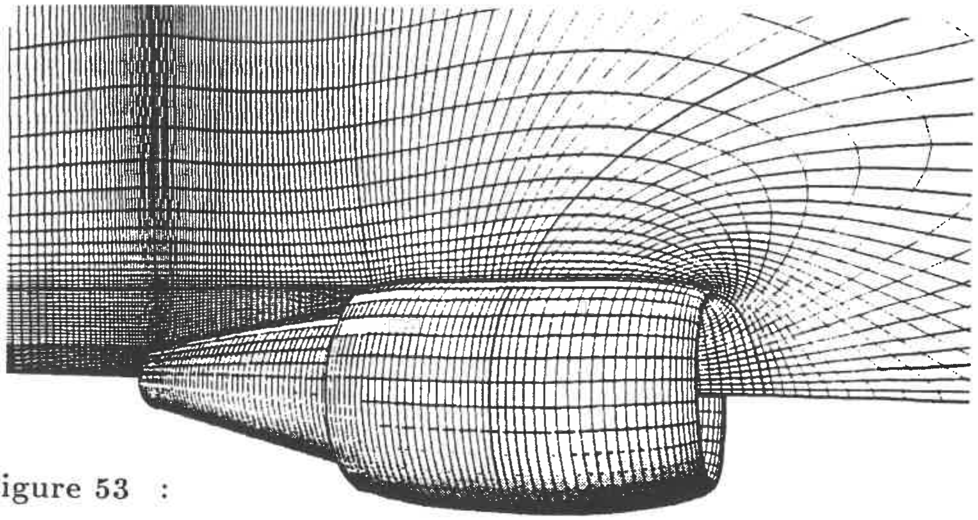


Figure 53 :

View of the 11-block body-fitted grid of the isolated CF-34 nacelle. The block boundaries are outlined on the drawing.

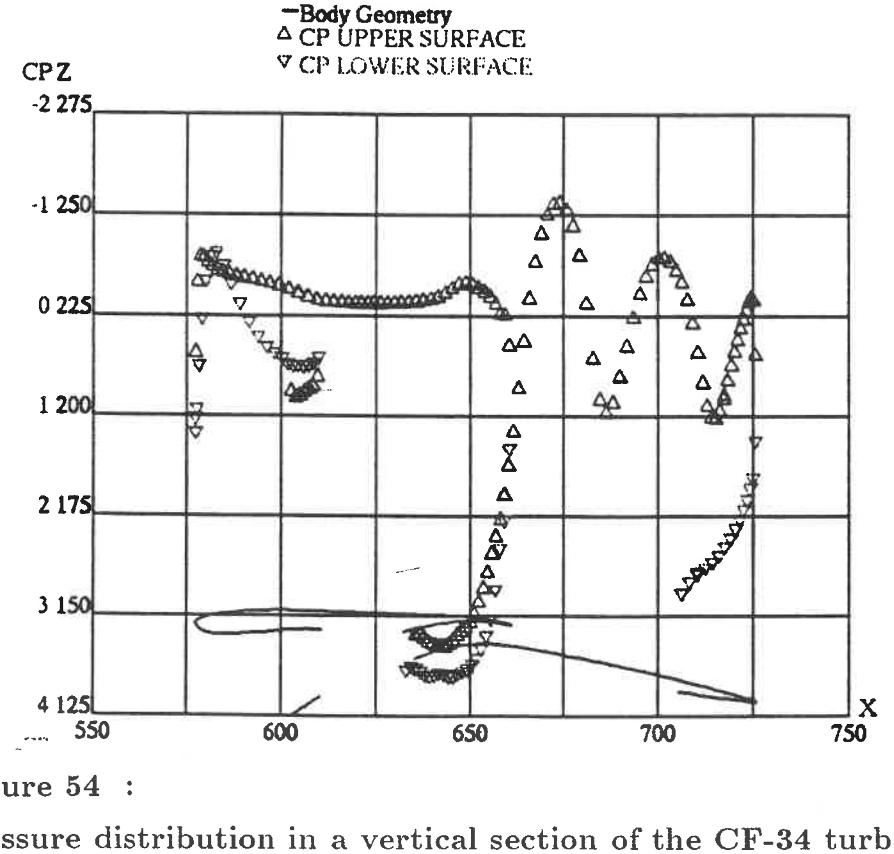


Figure 54 :

Pressure distribution in a vertical section of the CF-34 turbofan nacelle. 11-block solution at $M = 0.74$ and $\alpha = 0°$.
Case (1) engine operating conditions.

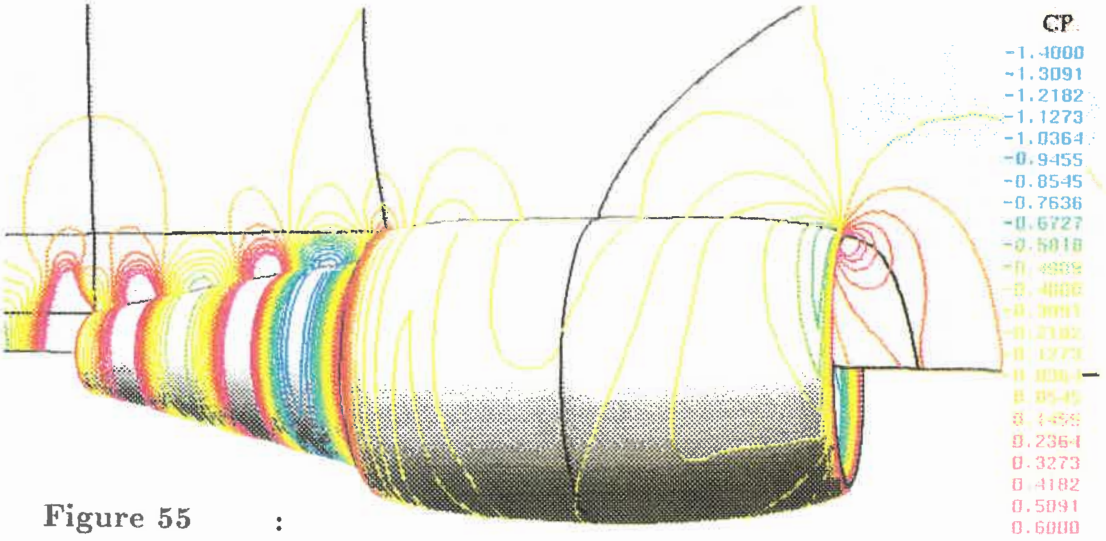


Figure 55 :

OMNI3D display of isobars on the CF-34 turbofan nacelle showing the expansion and compression cells. 11-block MBTEC solution. M = 0.74 and α = 0°. Case (1) engine operating conditions.

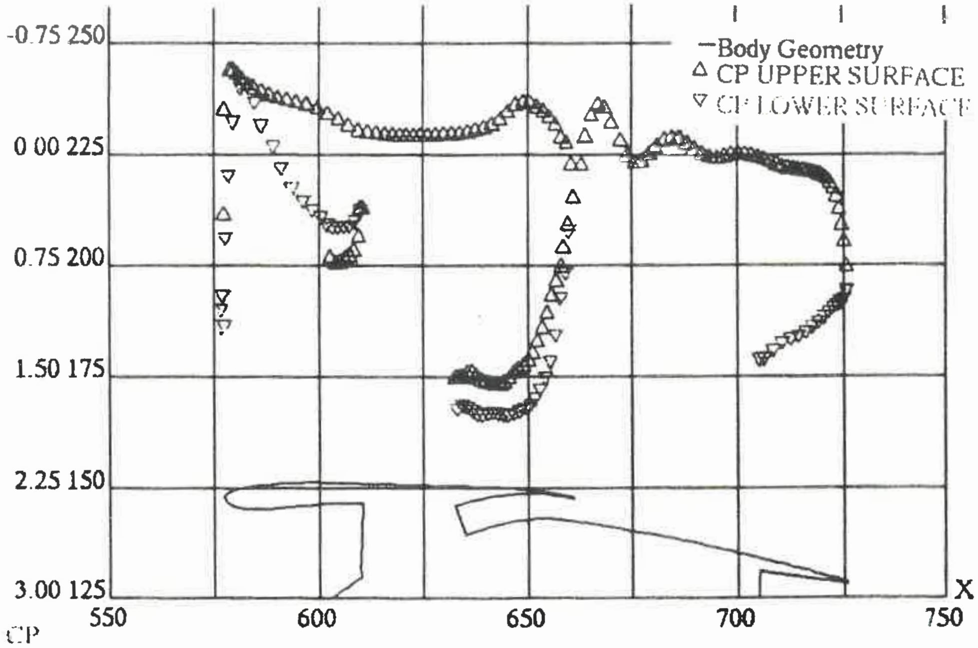


Figure 56 :

Pressure distribution in a vertical section of the CF-34 turbofan nacelle. 11-block MBTEC solution at M = 0.74 and α = 0°. Case (2) engine operating conditions.

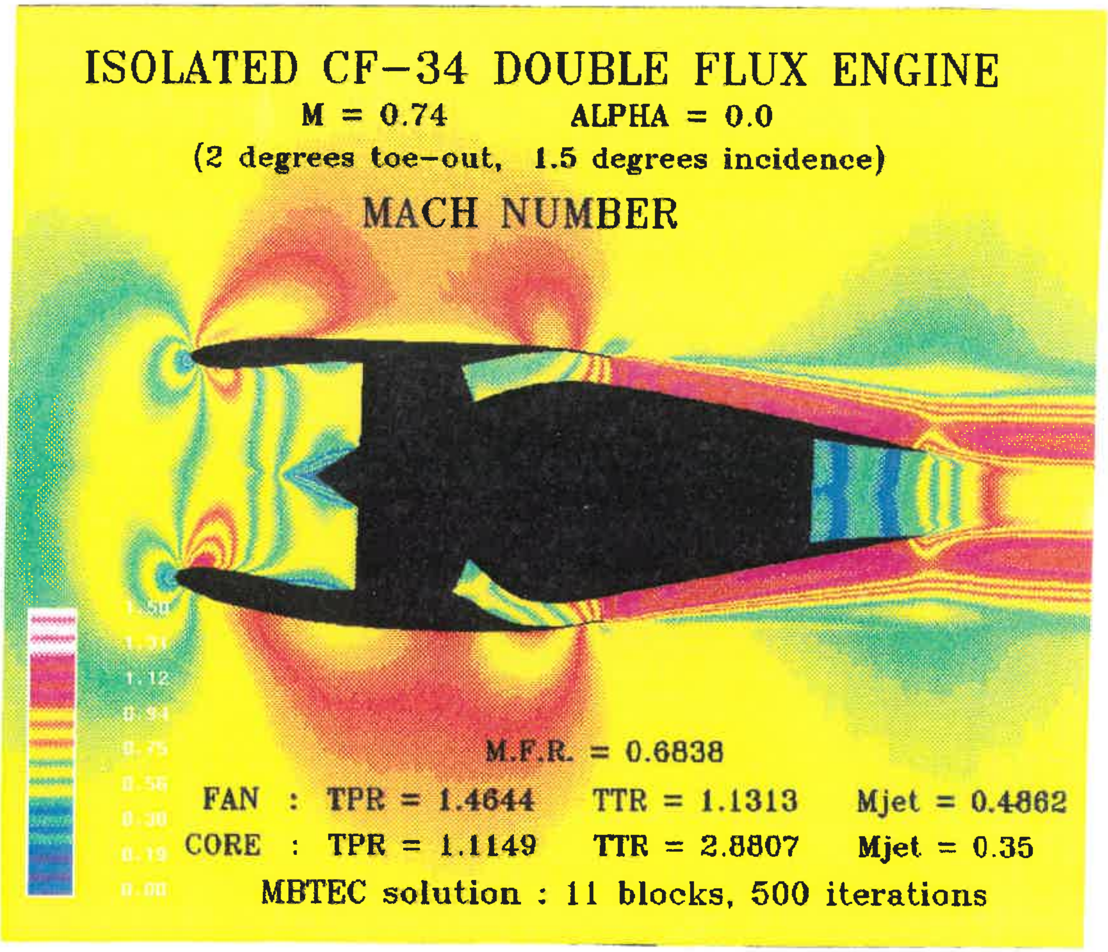


Figure 57 :

SSV-4D display of Mach numbers in the field around the CF-34 turbofan nacelle. 11-block MBTEC solution at $M = 0.74$ and $\alpha = 0^\circ$. Case (2) engine operating conditions.

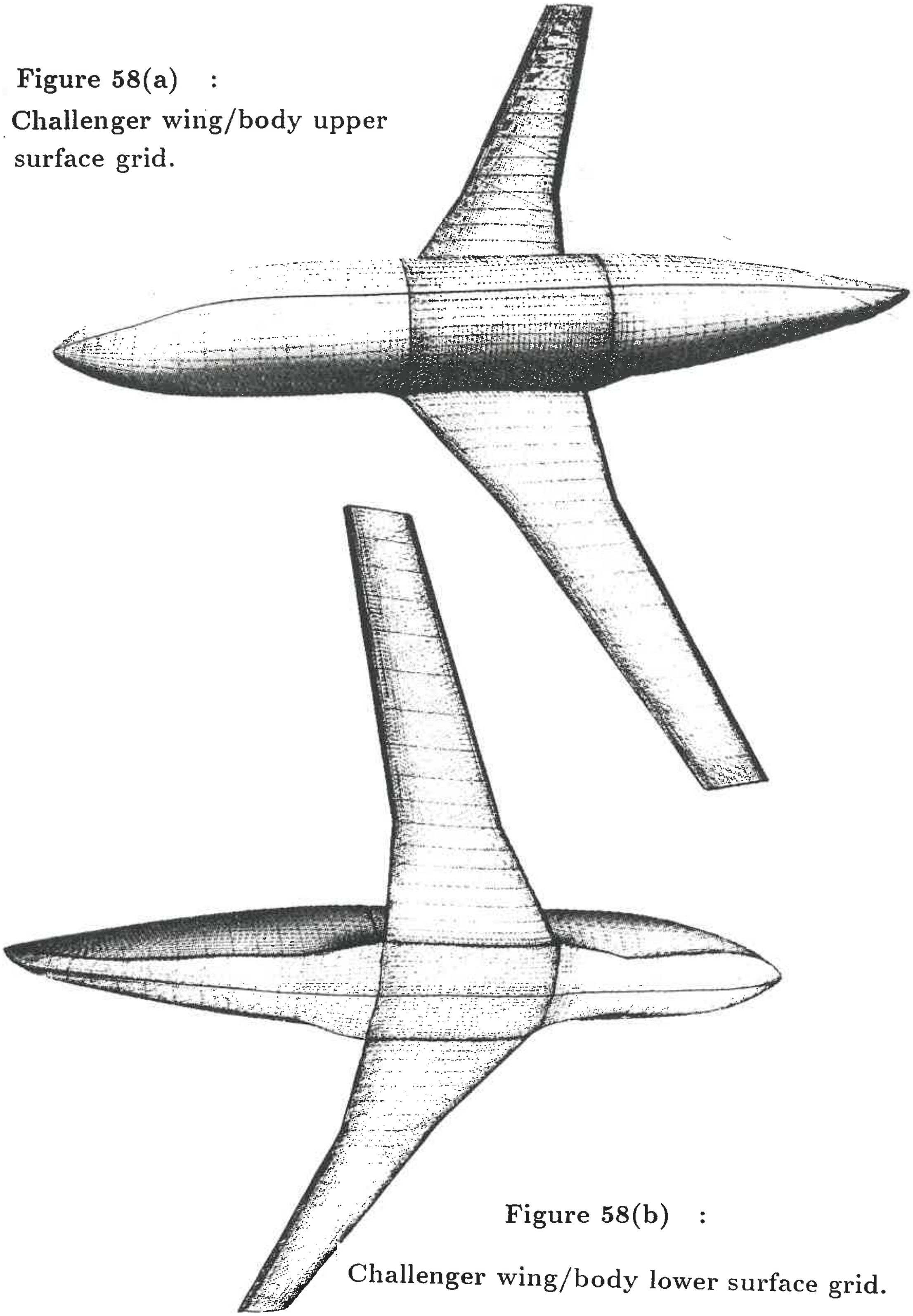


Figure 58(a) : Challenger wing/body upper surface grid.

Figure 58(b) : Challenger wing/body lower surface grid.

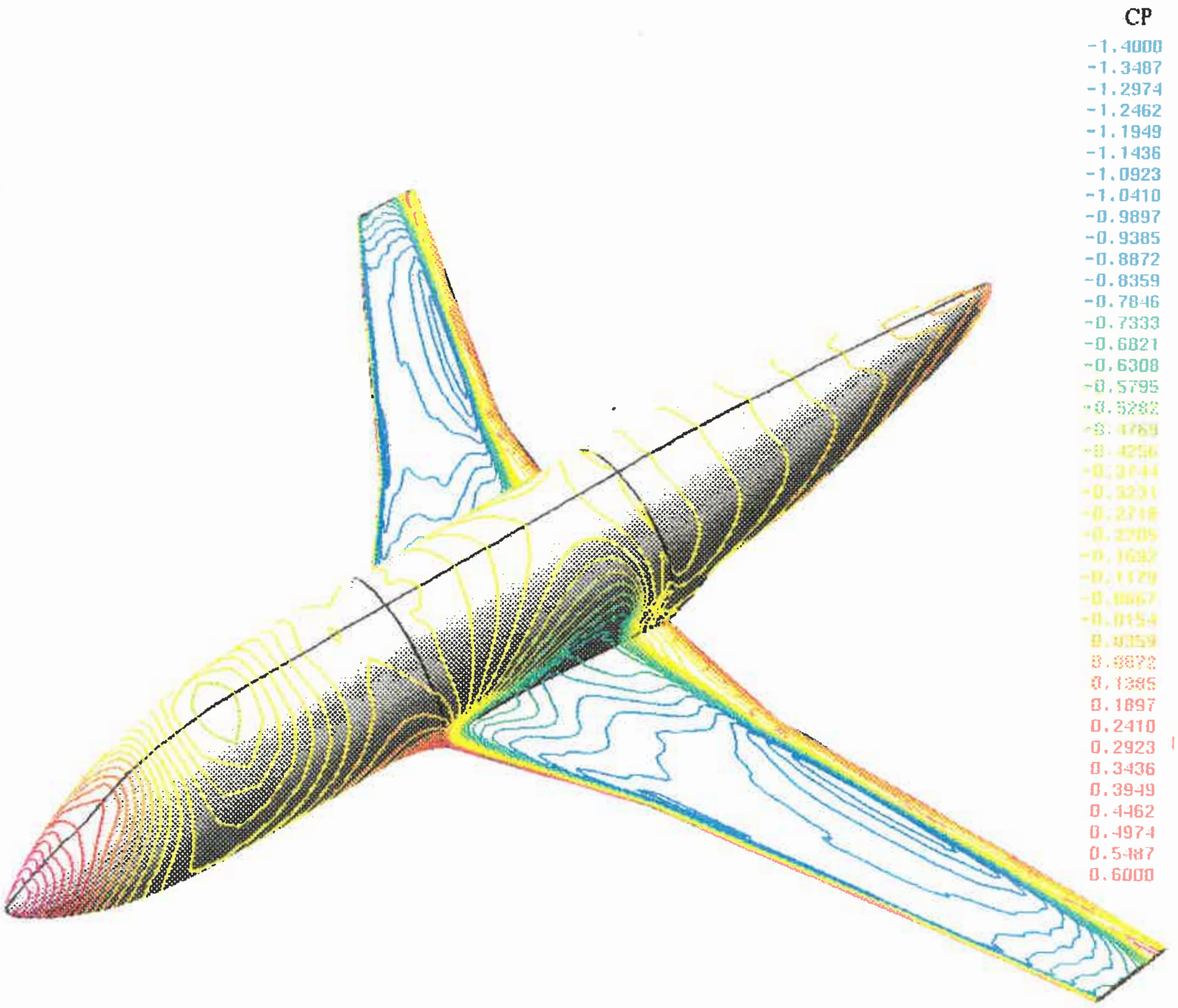


Figure 59 : Challenger wing/body surface isobars
40-block solution at M = 0.819 and $\alpha = 1.494^\circ$.

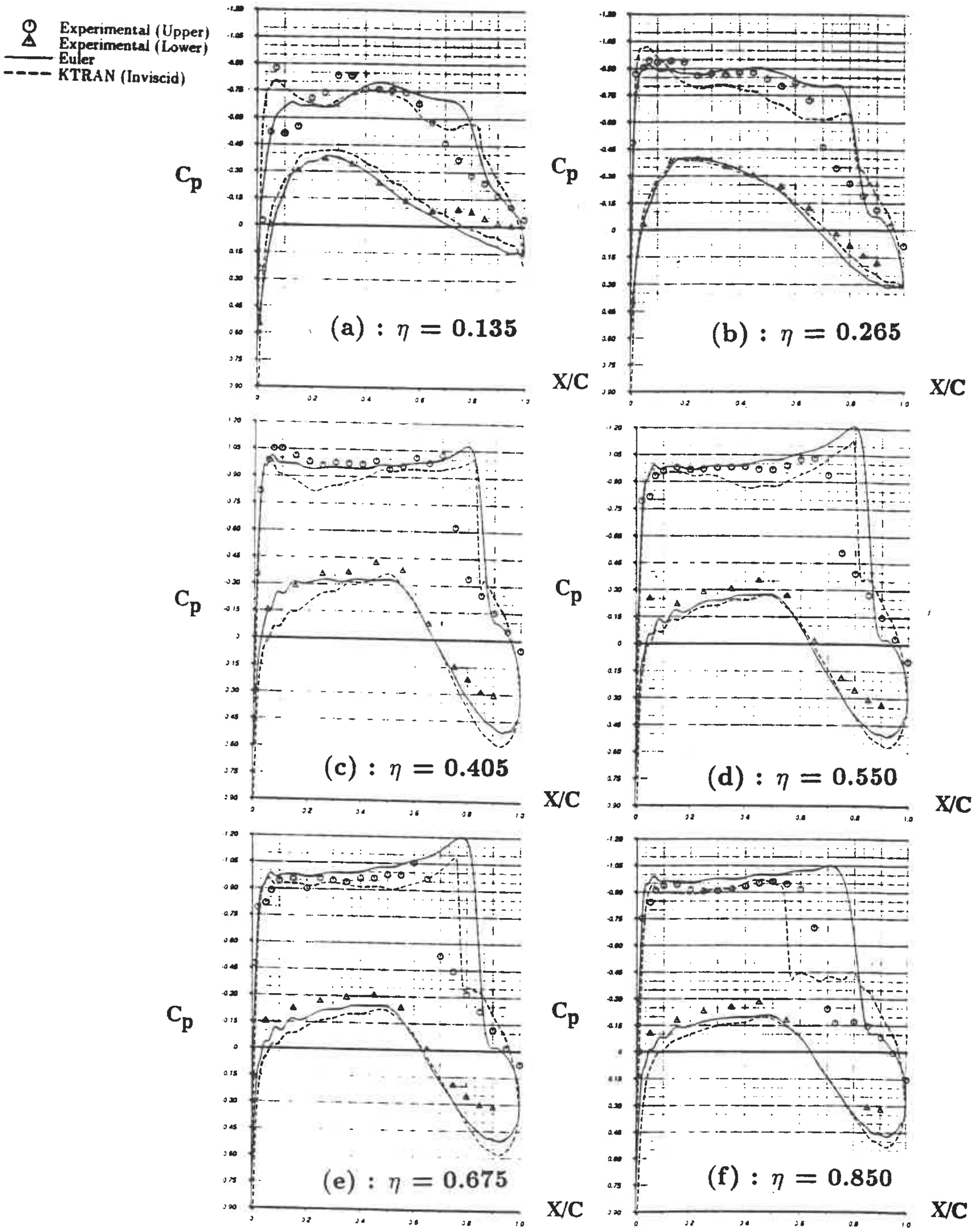


Figure 60 :

Challenger wing/body chordwise pressure distributions at six spanwise stations. M = 0.819 and $\alpha = 1.494°$.

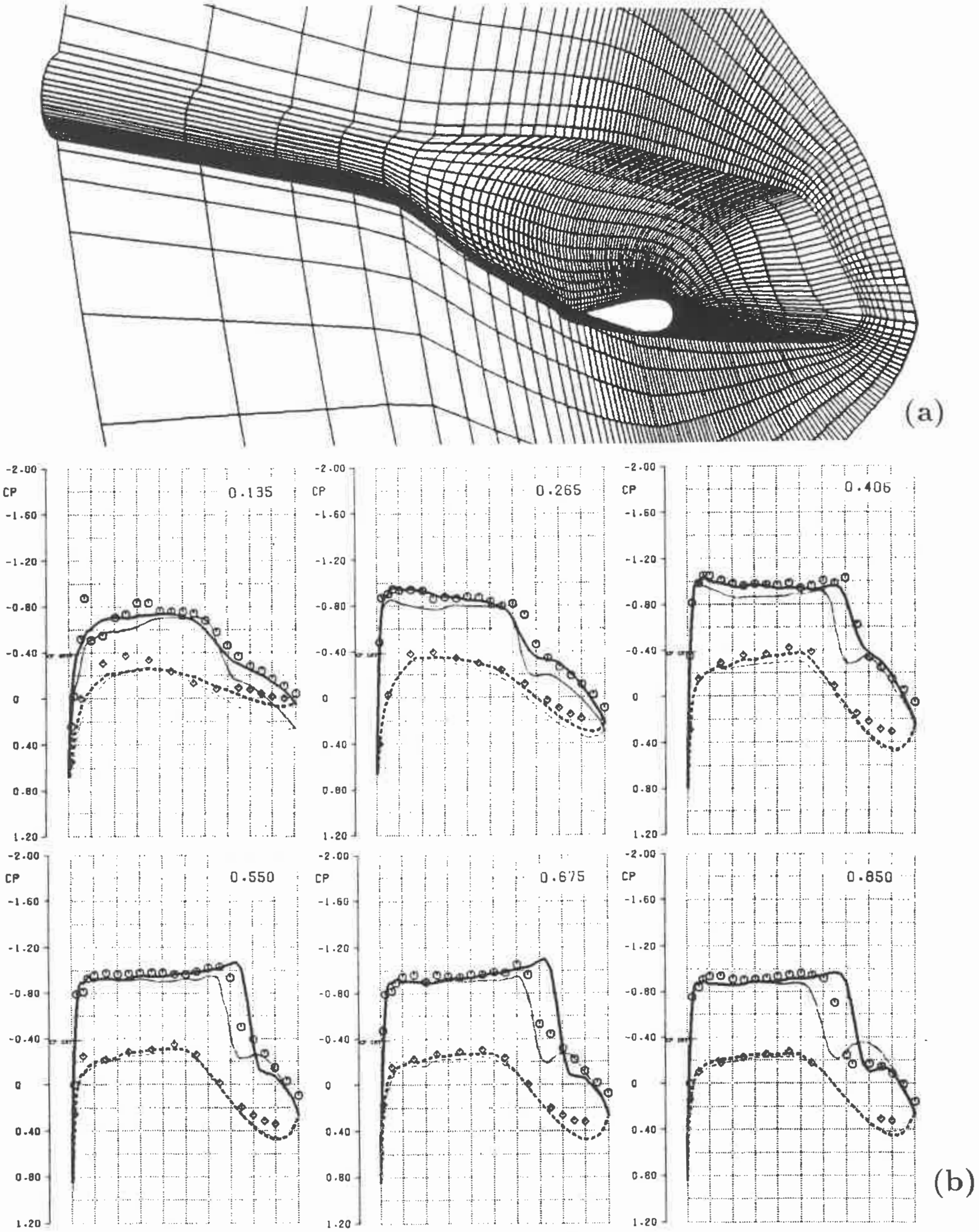


Figure 61 :

Challenger wing/body single-block Euler/boundary layer solution. M = 0.819 and $\alpha = 1.494^\circ$.
(a) C-H grid plane K=1 covering body and plane of symmetry
(b) Chordwise pressure distributions at six spanwise stations (Mokhtarian [108])

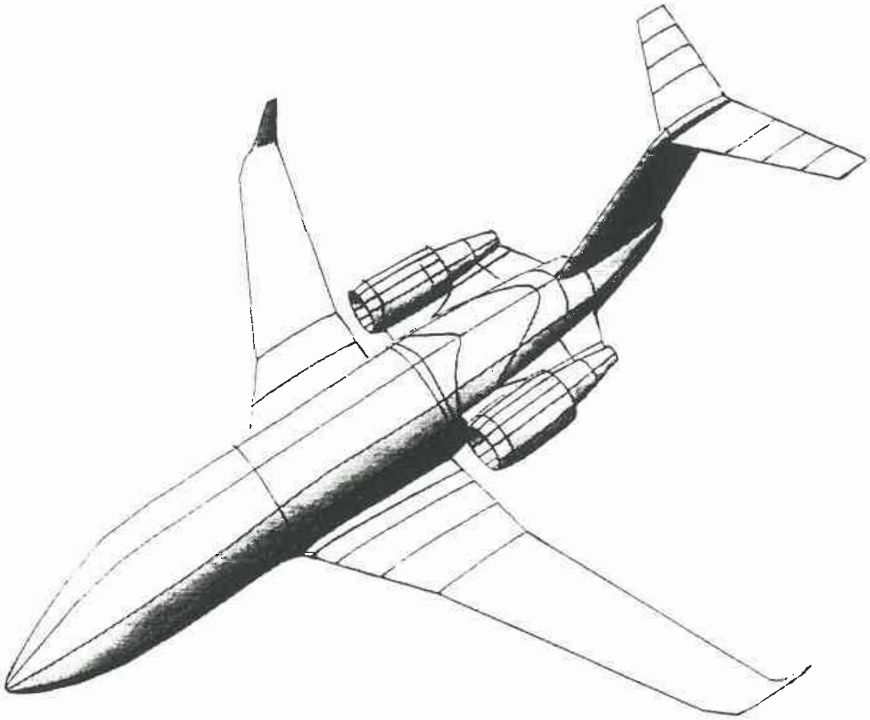


Figure 62(a) : Block boundaries outlined on the upper surface of the Challenger CL-601 aircraft.

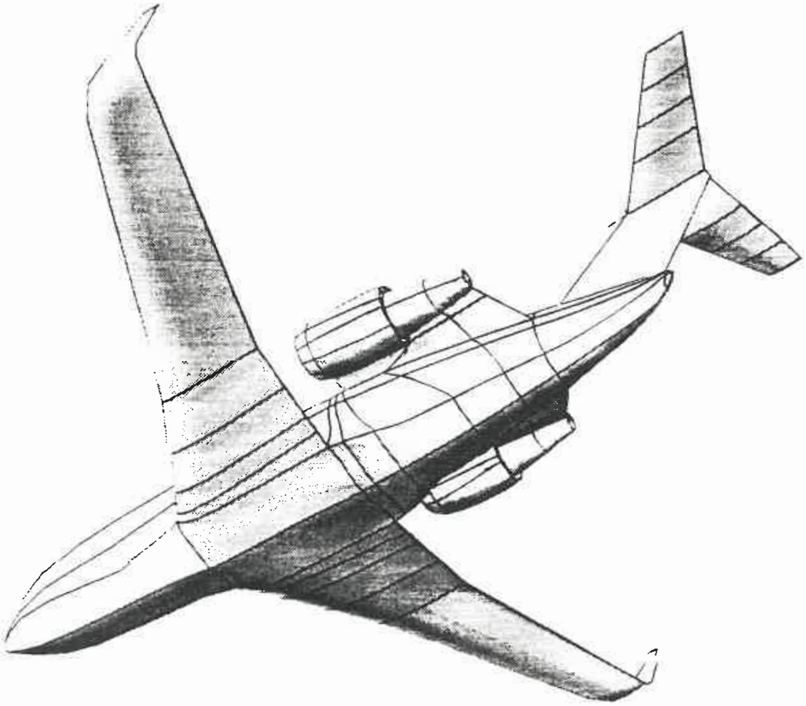


Figure 62(b) : Block boundaries outlined on the lower surface of the Challenger CL-601 aircraft.

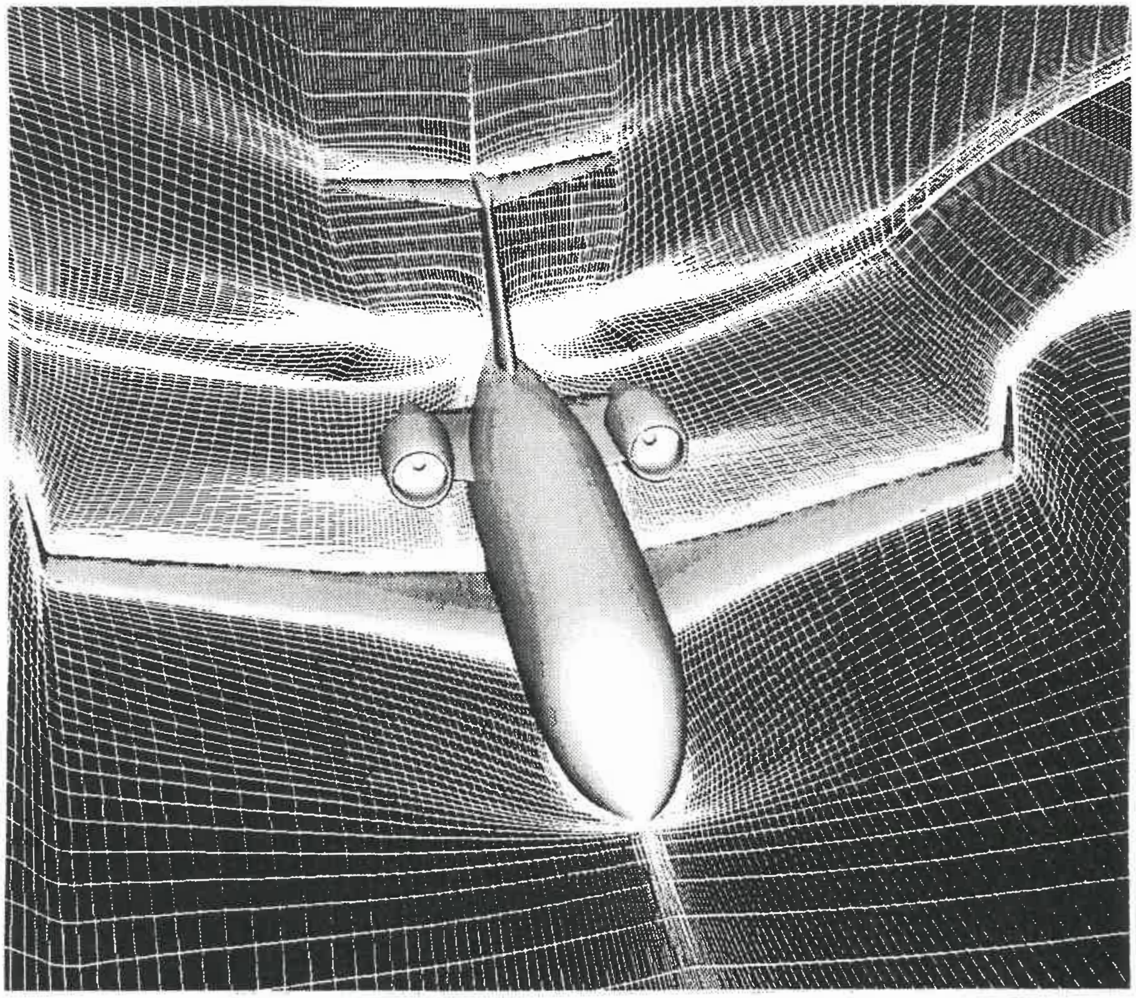


Figure 63 : Multi-block grid for the Challenger CL-601 full aircraft

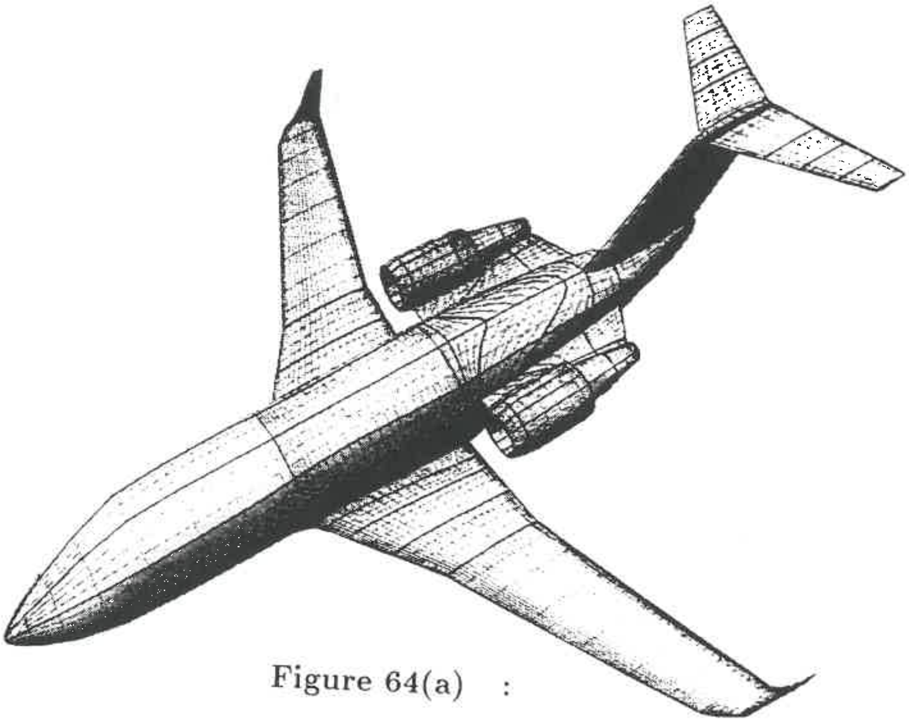


Figure 64(a) :

Challenger CL-601 aircraft upper surface grid (intermediate)

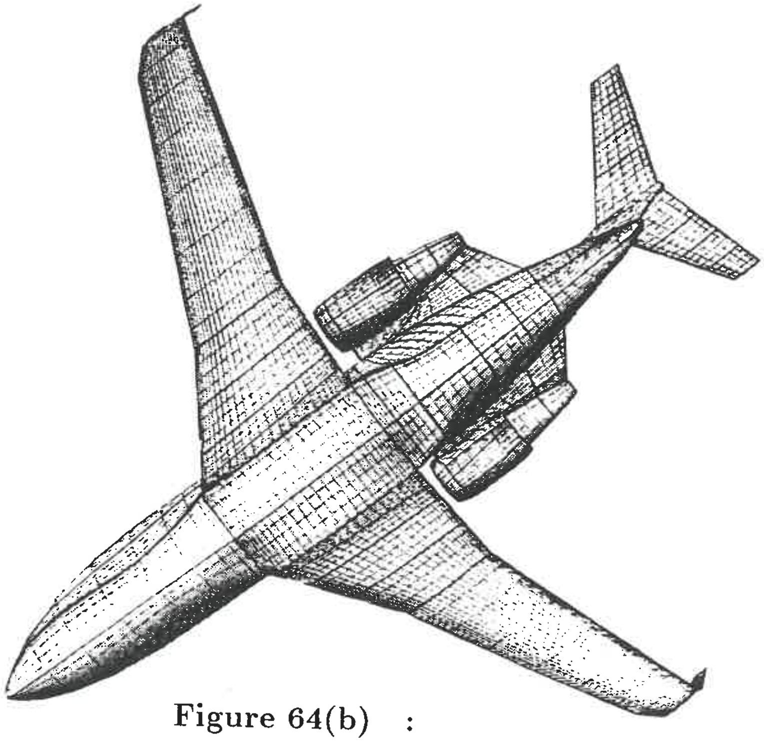


Figure 64(b) :

Challenger CL-601 aircraft lower surface grid (intermediate)

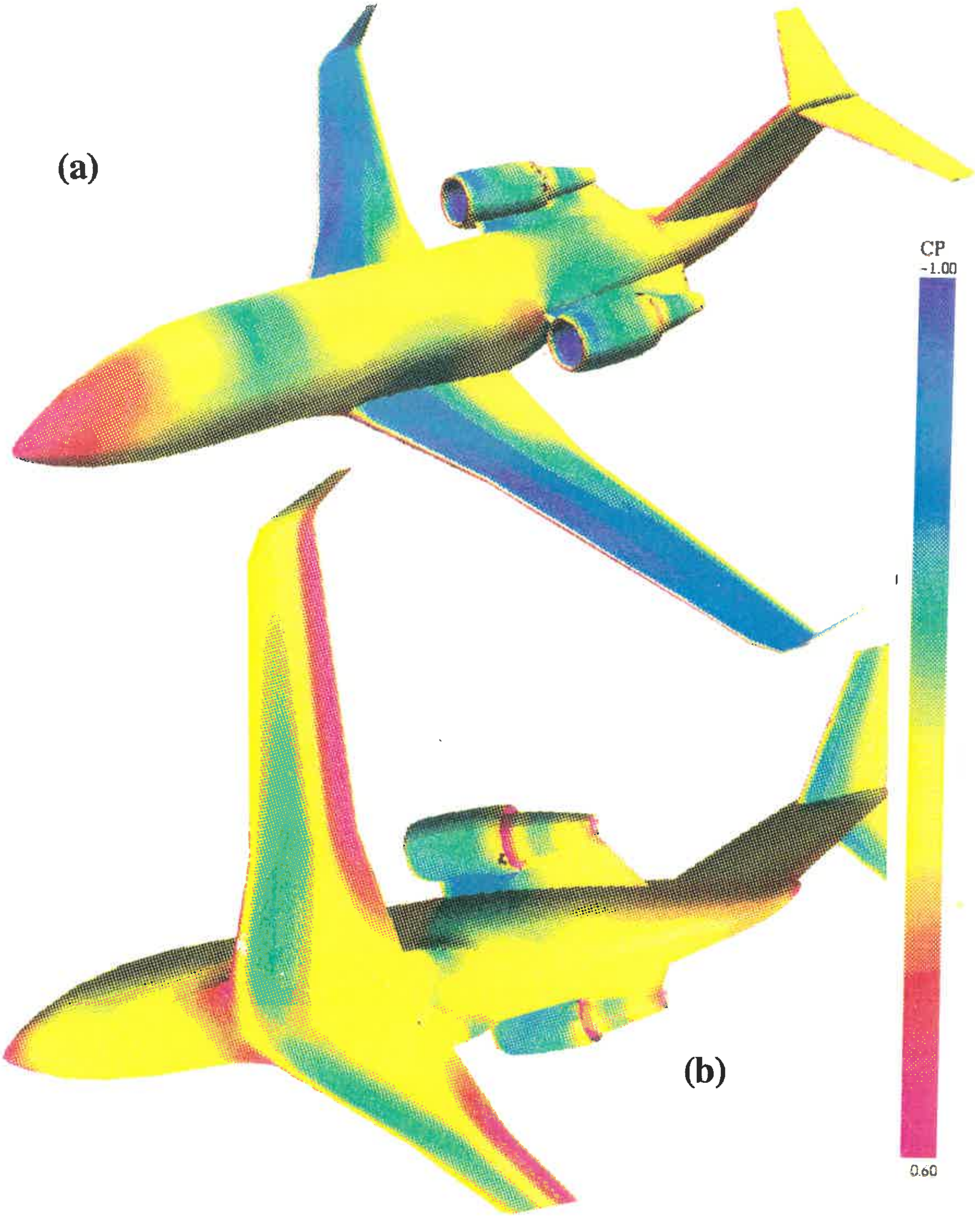


Figure 65 : Challenger complete configuration surface isobars at $M = 0.799$ and $\alpha = 0.93^\circ$. (a) upper surface isobars (b) lower surface isobars

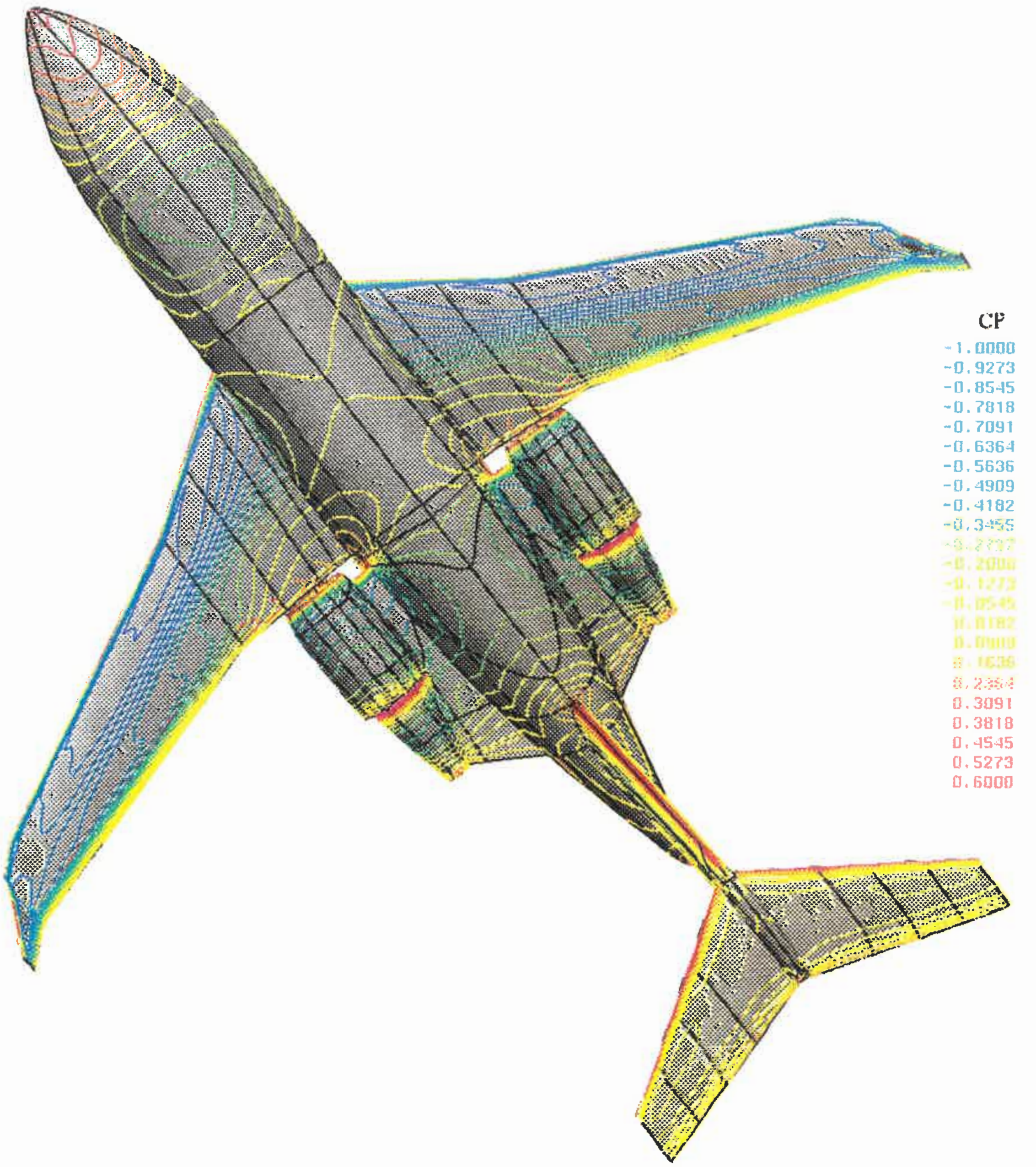


Figure 66 :

Challenger complete configuration upper surface isobars at $M = 0.799$ and $\alpha = 0.93^\circ$. With outlined block edges.

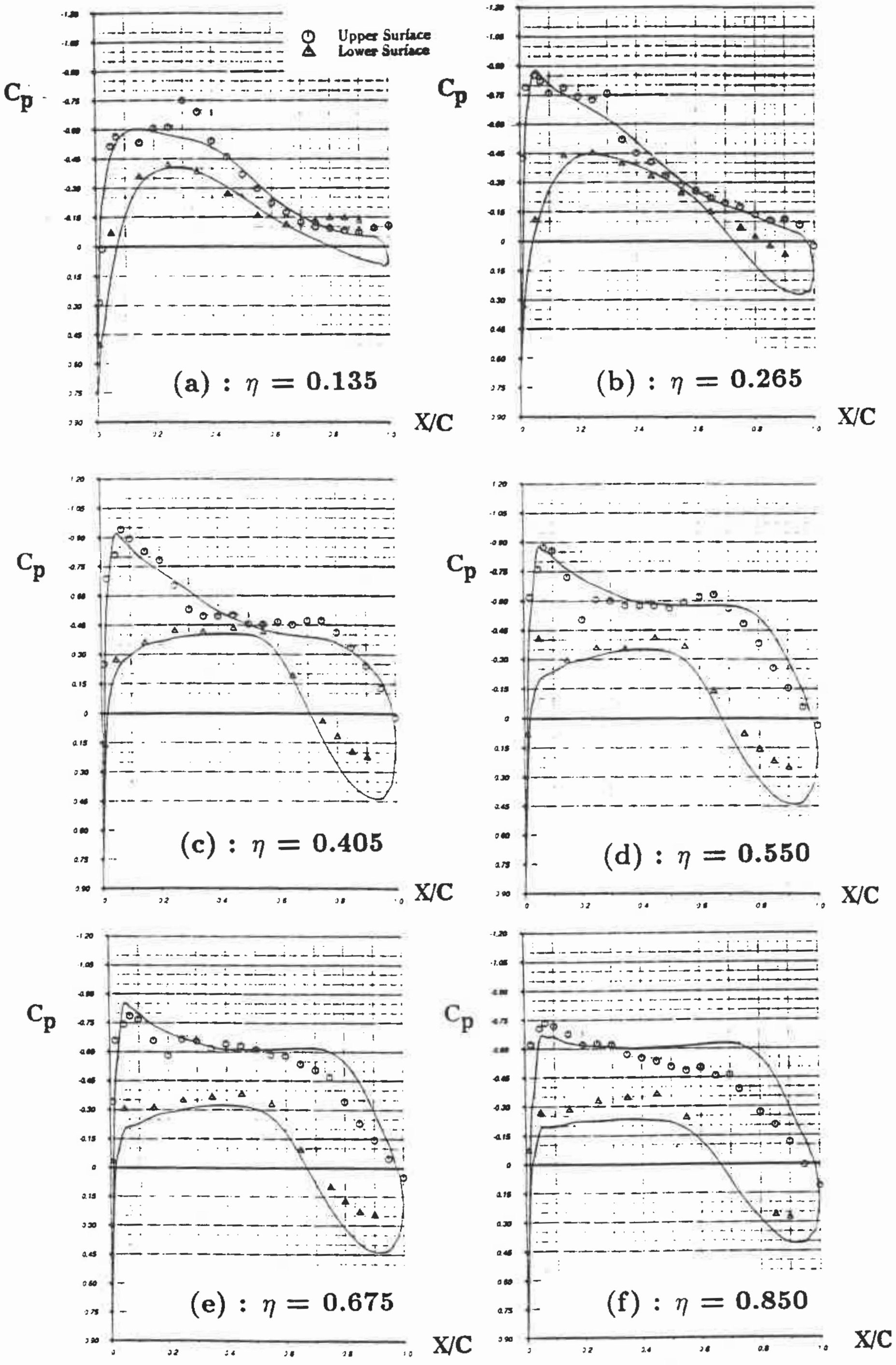


Figure 67 :
CL-601 complete configuration chordwise pressure distributions at six spanwise stations. M = 0.799 and $\alpha = 0.93^\circ$.

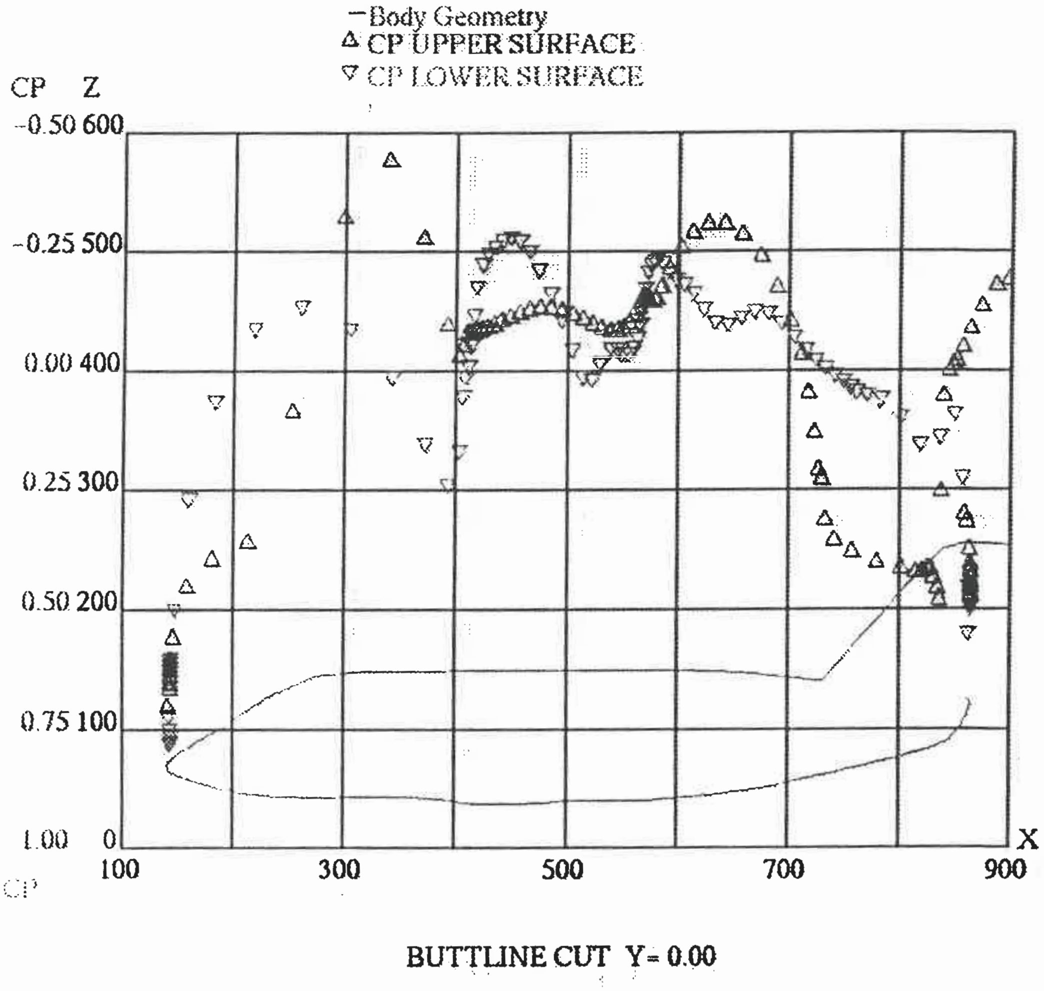


Figure 68 :

CL-601 complete configuration; fuselage centerline pressure distributions. M = 0.799 and $\alpha = 0.93°$.

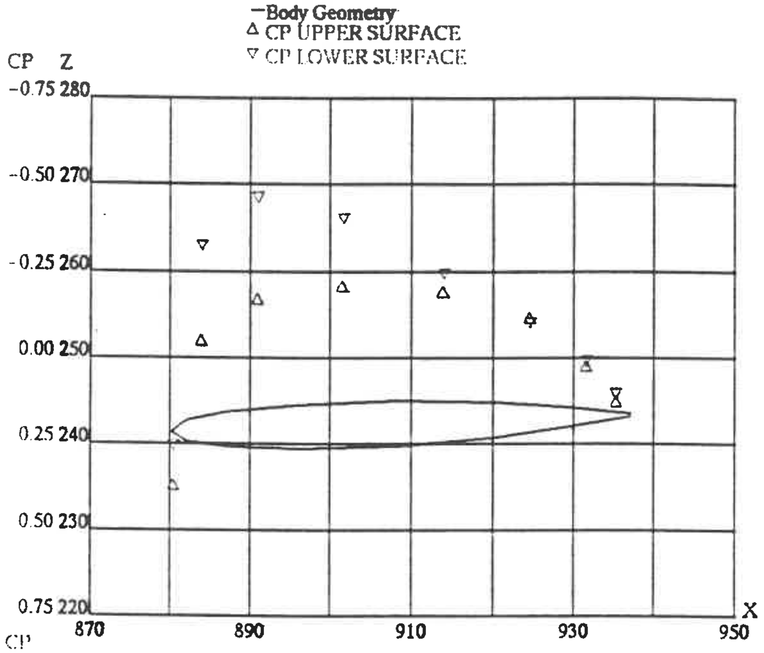


Figure 69 :

CL-601 complete configuration; pressure distribution at 45% semi-span of horizontal tailplane. M = 0.799 and $\alpha = 0.93°$.

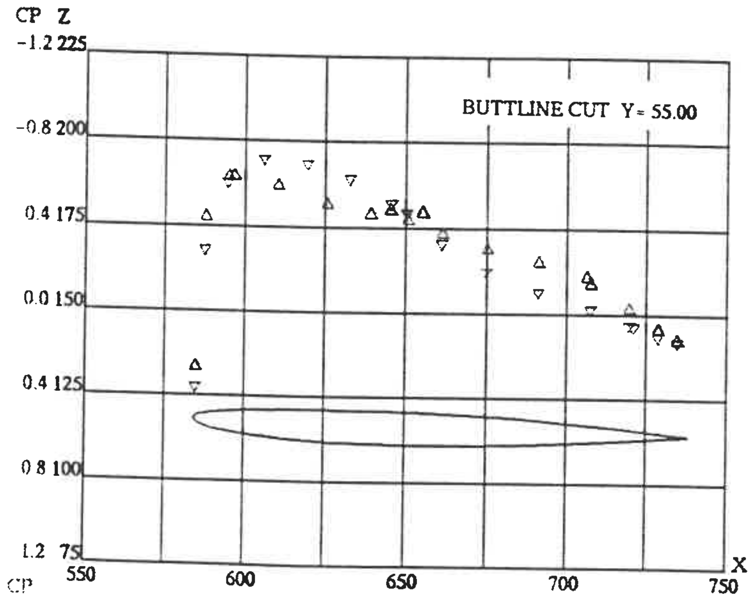


Figure 70 :

CL-601 complete configuration; pressure distribution at mid-span of nacelle pylon. M = 0.799 and $\alpha = 0.93°$.

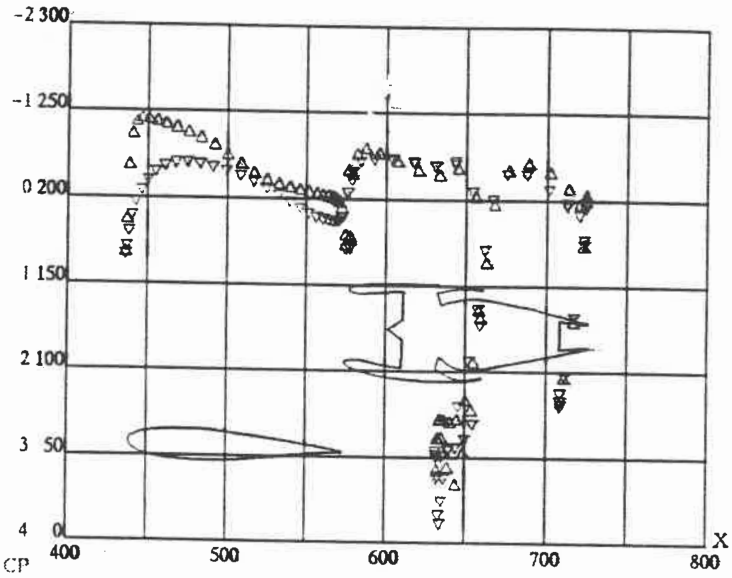


Figure 71 : CL-601 complete configuration; pressure distribution at buttline 88 on nacelle centerline and wing.
M = 0.799 and $\alpha = 0.93^\circ$.

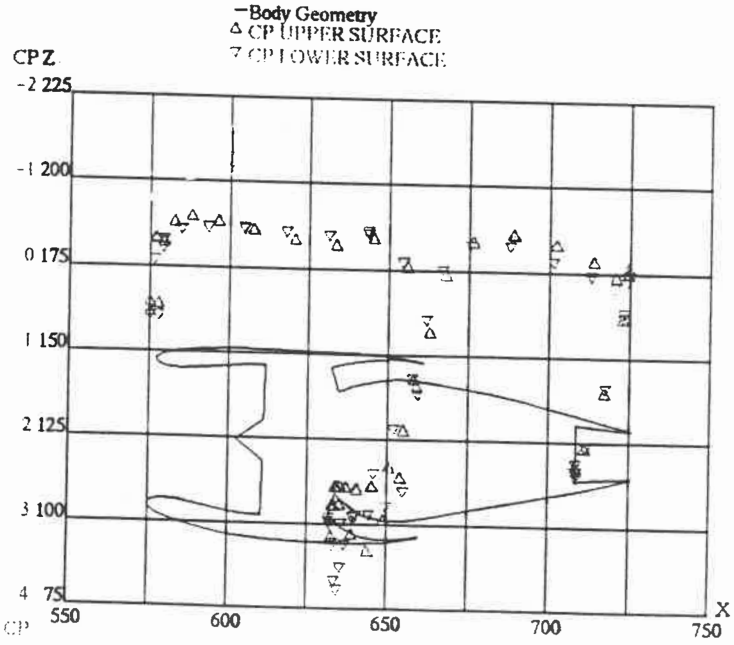


Figure 72 : CL-601 complete configuration; pressure distribution at buttline 88 on nacelle centerline surfaces.
M = 0.799 and $\alpha = 0.93^\circ$.

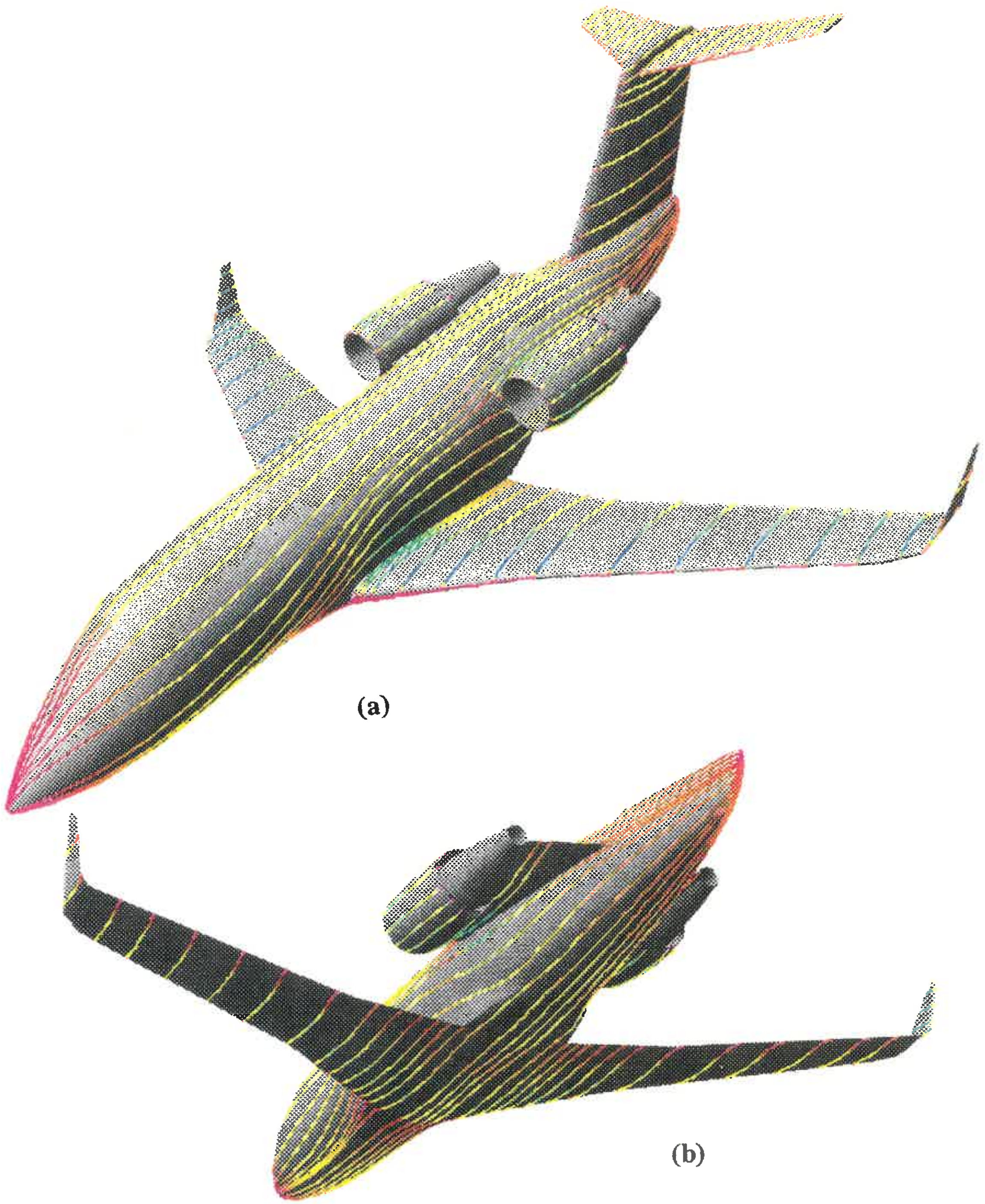


Figure 73 :

CL-601 complete configuration; surface streamlines predicted from Euler solution at $M = 0.799$ and $\alpha = 0.93°$.
(a) upper surface streamlines; (b) lower surface streamlines.

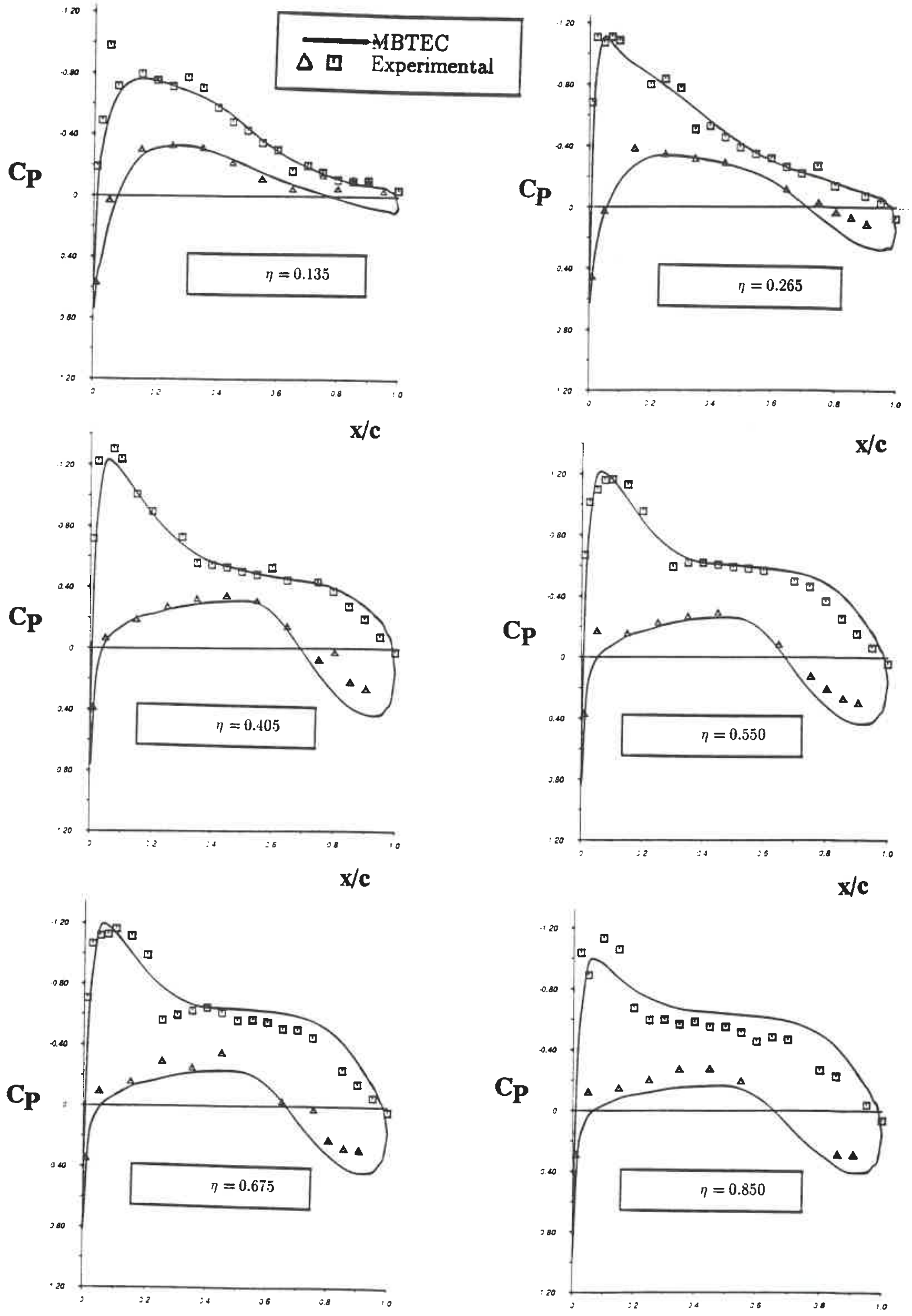


Figure 74 : CL-601 complete configuration chordwise wing pressure distributions at six spanwise stations. M = 0.76 and $\alpha = 2.10^\circ$.

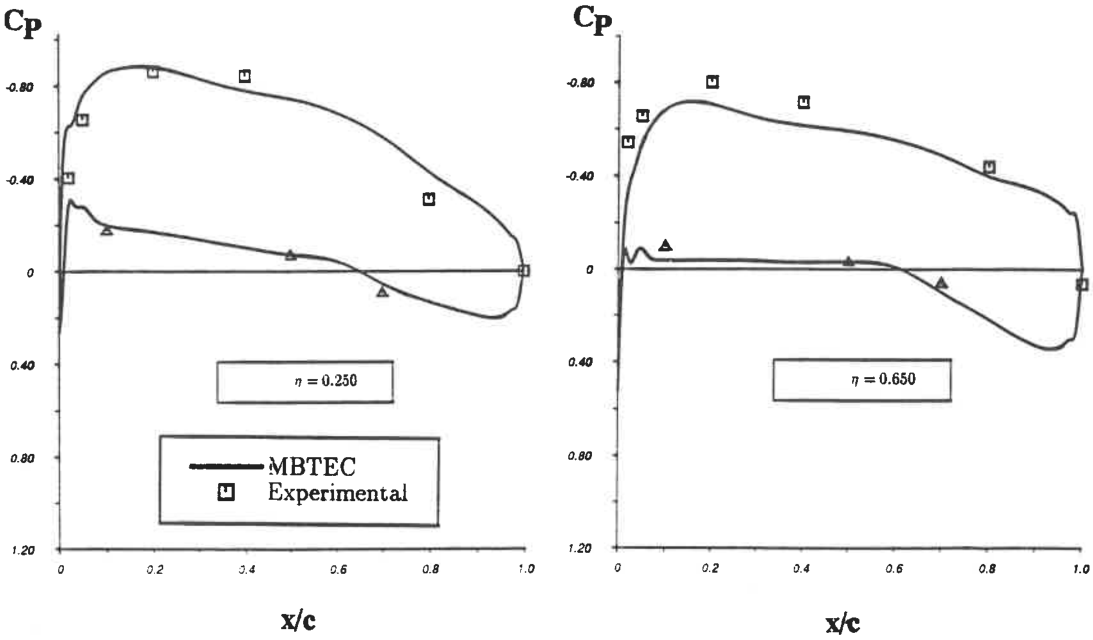


Figure 75 :

CL-601 complete configuration chordwise winglet pressure distributions at two spanwise stations. M = 0.76 and $\alpha = 2.10^\circ$.

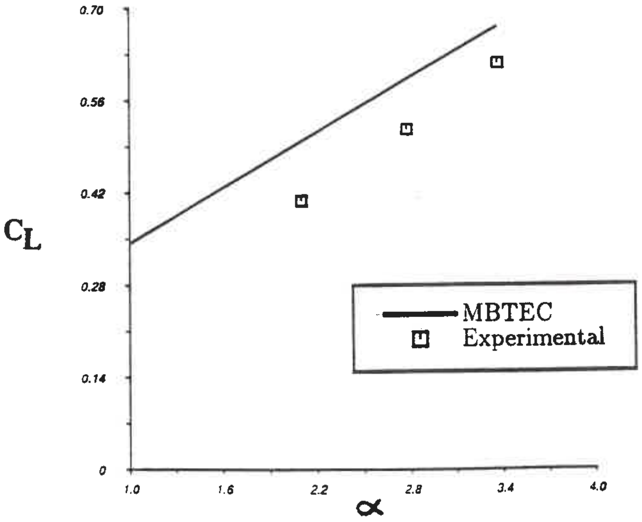


Figure 76 :

CL-601 complete configuration lift curve slope at Mach 0.76

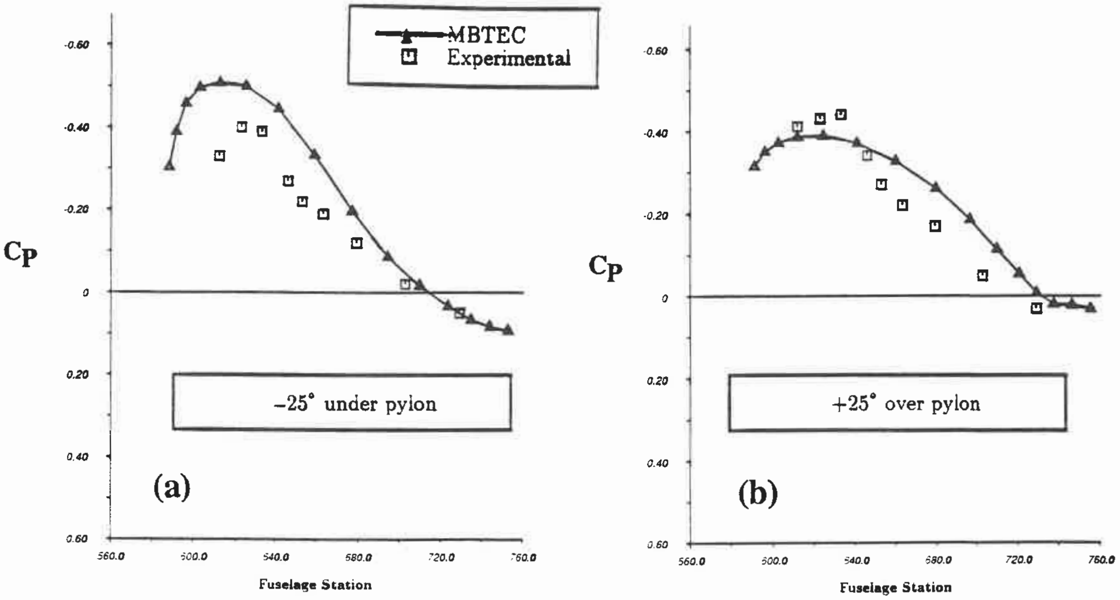


Figure 77 :

CL-601 complete configuration at Mach 0.76 and $\alpha = 2.10^\circ$**. Pressure distributions on fuselage near the nacelle.**

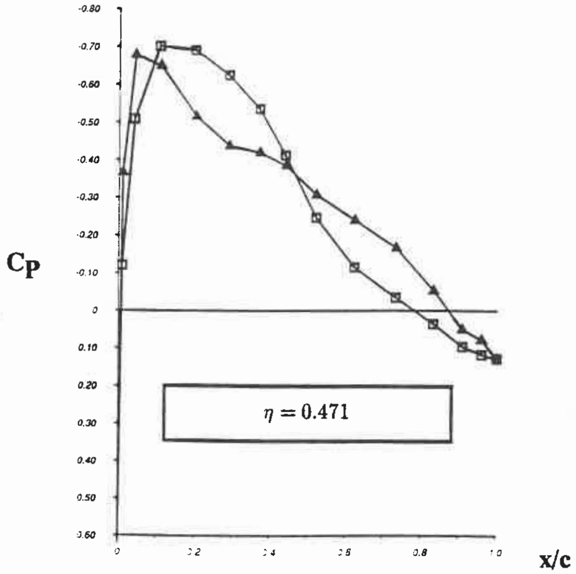


Figure 78 :

CL-601 complete configuration at Mach 0.76 and $\alpha = 2.10^\circ$**. Pressure distributions near the nacelle pylon centerline.**

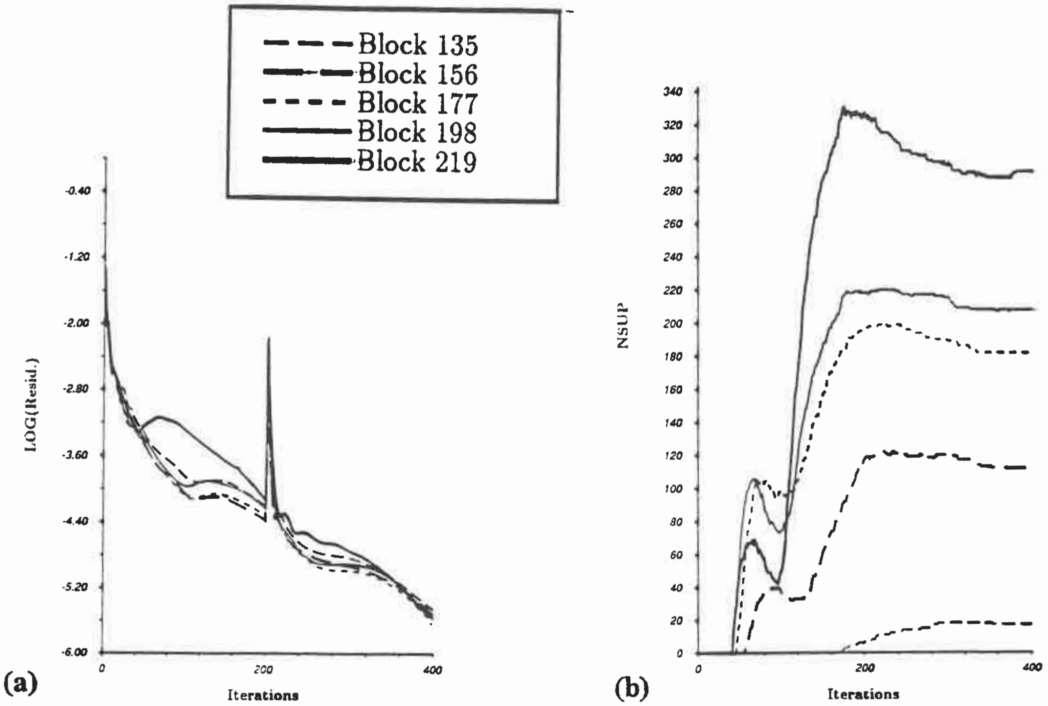


Figure 79 :

CL-601 complete configuration at Mach 0.76 and $\alpha = 2.10^{\circ}$. Convergence characteristics in blocks above the wing : (a) average residual, (b) number of supersonic points.

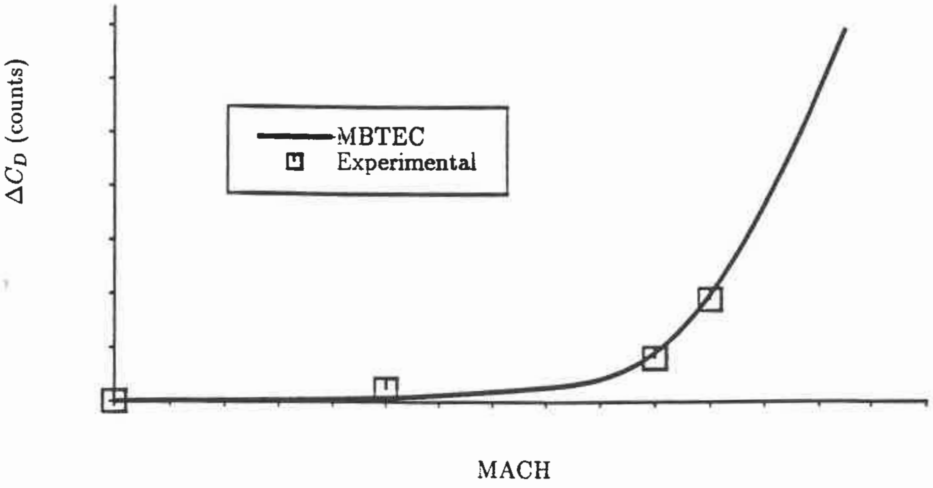


Figure 80 :

Drag rise curve for the CL-601 wing-body configuration at $C_L = 0.6$.

7.8 CF-18 COMPLETE CONFIGURATION

Another objective of the development of the **MBTEC** program was the ability to model accurately various types of flows around the Canadian Forces CF-18 aircraft. The engine inlets were covered by a streamlined surface for the exercise, which focussed on the ability to model the aircraft general aerodynamics. Results for this airplane are included here as a further validation of the method. The modelling of the CF-18 aircraft required the addition of several new topological capabilities to the **MBTEC** code. A sting mounted wing-body combination was considered first and used to tune the code to model accurately the wing strake. This strake is treated in the code as a closed wing tip. The wing-body model, made of 30 blocks, was used also to study the effect of the artificial viscosity coefficients on the stability of the solution at high angles of incidence. In the input data to the program, the parameters controlling the levels of the second and fourth order artificial viscosity coefficients are called VIS2 and VIS4. At low angles of incidence, values VIS2 = 1.0 and VIS4 = 1.0 are typically sufficient, in combination with a CFL (Courant, Friedrichs and Lewy) input value of 5.0. It was found during this exercise that for angles between 15 ° and 20 °, values of artificial viscosity coefficients VIS2 = 1.5 and VIS4 = 1.5 are needed. In a second step, the complete aircraft, including horizontal and vertical tailplanes, was modelled. In order to conform to the 6% scale model for which experimental data was obtained at the I.A.R. wind tunnel in Ottawa (private communication), the horizontal tailplane was set at an angle of -9°. This tail angle was kept constant for low angles of incidence as well, even though it may not be a typical flying configuration. The complete aircraft fine grid was constructed using 108 blocks. The domain decomposition of the space around the aircraft, performed on CADAM, is represented on figures 81(a) and 81(b). The fine grid generated for the CF-18 contains 784,168 nodes. A view of this grid is shown in figure 82. The intermediate grid used for the calculations discussed here contains 114,450 nodes. A view of the intermediate surface grid including an outline of the block edges is shown in figures 83(a) and 83(b). Calculations were made first at Mach 0.95 and $\alpha = 2.79^\circ$, for comparison with available wind tunnel test data. The surface isobars obtained after 200 iterations on the coarse grid and 400 iterations on the intermediate grid are shown in figures 84(a) and 84(b). An additional top view of these results is given in figure 85 to indicate that the location of the block edges is transparent to the solution. In this particular case, the shock wave on the upper surface of the wing appears to be located very near the junction of two blocks on the wing. The adequate convergence of the solution in this case is a further argument verifying the validity of interface boundary conditions used in the program. A comparison of these pressure distributions with wind tunnel test data is shown for six wing stations on figures 86 (a) to (f). The experimental data was obtained from the Institute of Aerospace Research through Canadair's Defence System Division. The locations of the shock waves on the upper and lower surfaces of the wing are predicted very well at the six spanwise stations (within 3% chord). The level of pressures is predicted fairly well on the lower surface, but there are some discrepancies on the upper surface, particularly near the leading edge. The theoretical results indicate higher suction levels than measured experimentally at a Reynolds number of about 4.19 million. High angle of incidence calculations were made at Mach 0.60 and alpha = 16.9°. The isobars obtained at the surface of the aircraft are shown on figures 87(a) and 87(b). At this angle of incidence, there are vortices generated along the strakes and going over the wing. The trace of these vortices can be seen on figure 88 where the normalized stagnation pressure is plotted at various fuselage stations. These calculations indicate the ability of the program to calculate stable solutions for highly vortical flow fields, as well as for high transonic

Mach numbers where strong shock waves are present. The Euler equations can capture primary vortices such as those originating from the leading edge extension, but secondary separation vortices and massively separated flows can only be captured by a Navier-Stokes program. Additional correlations with experimental data are still required for the CF-18 configuration.

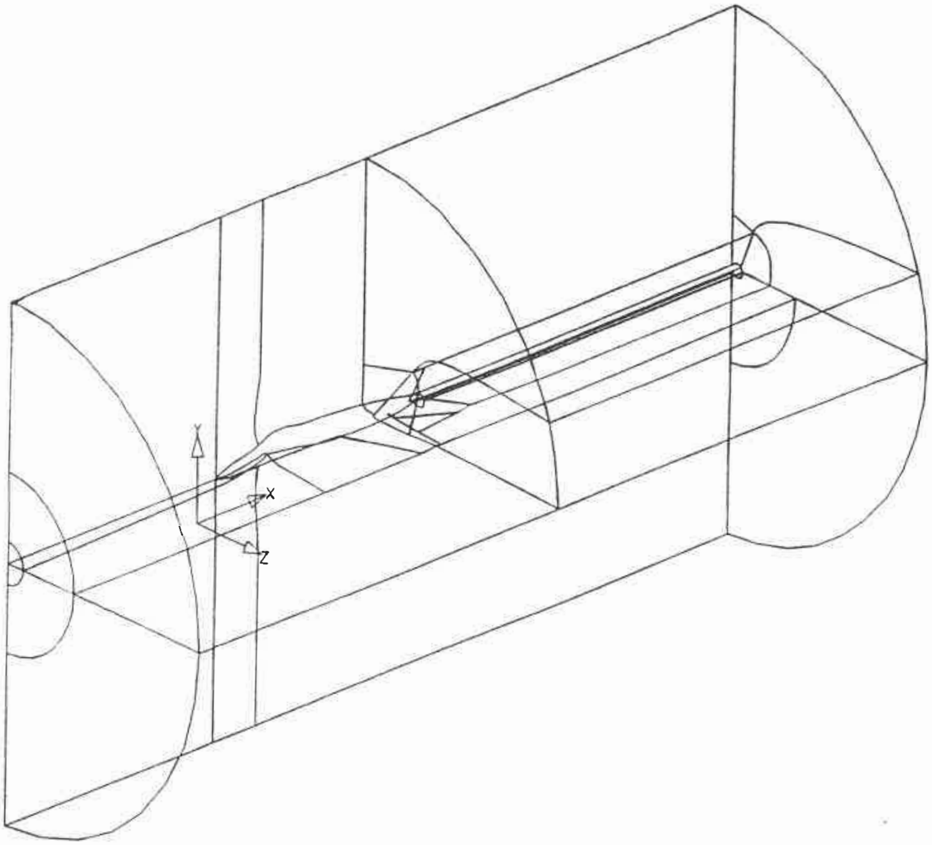


Figure 81(a): CADAM physical domain decomposition for the CF-18.

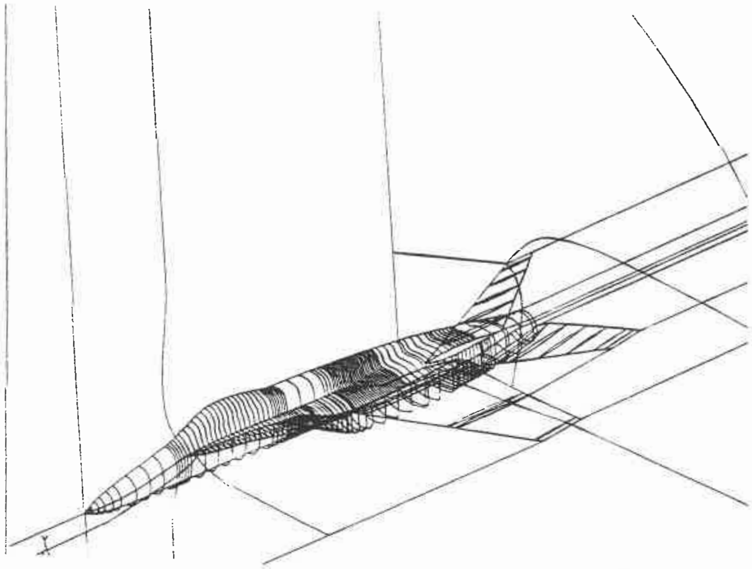


Figure 81(b): close-up view of domain decomposition and geometry modelling for the CF-18 aircraft.

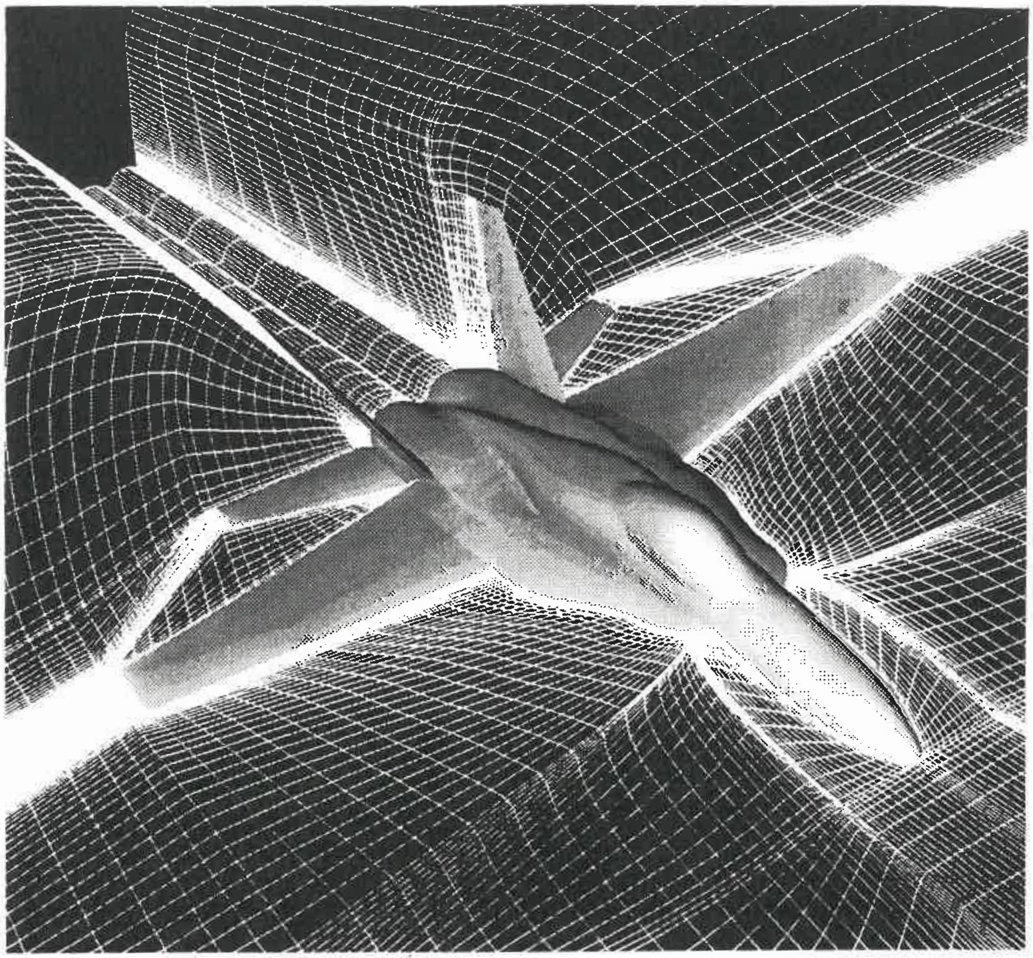


Figure 82 : Multi-block grid for the CF-18 clean aircraft.

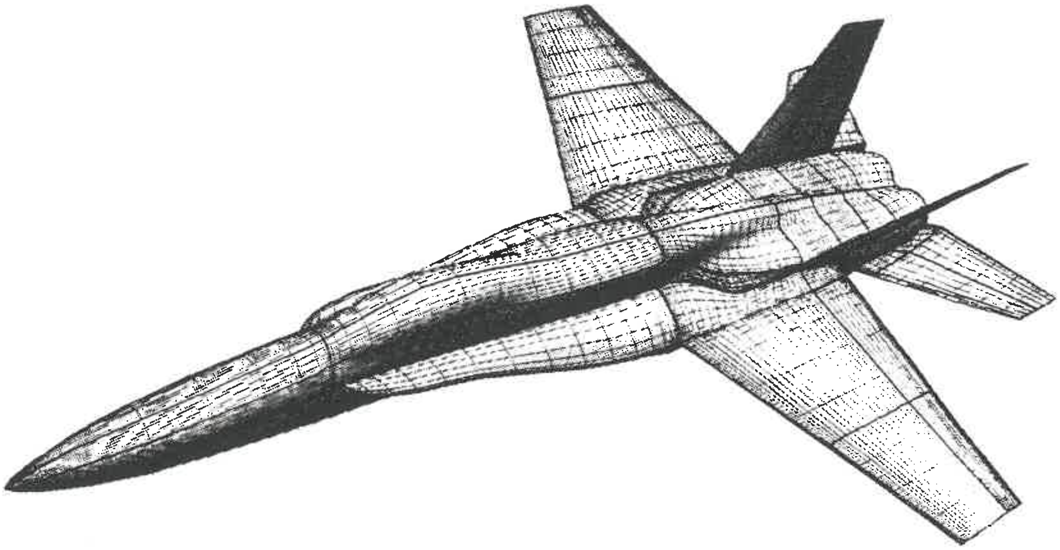


Figure 83(a): CF-18 aircraft upper surface grid (intermediate).

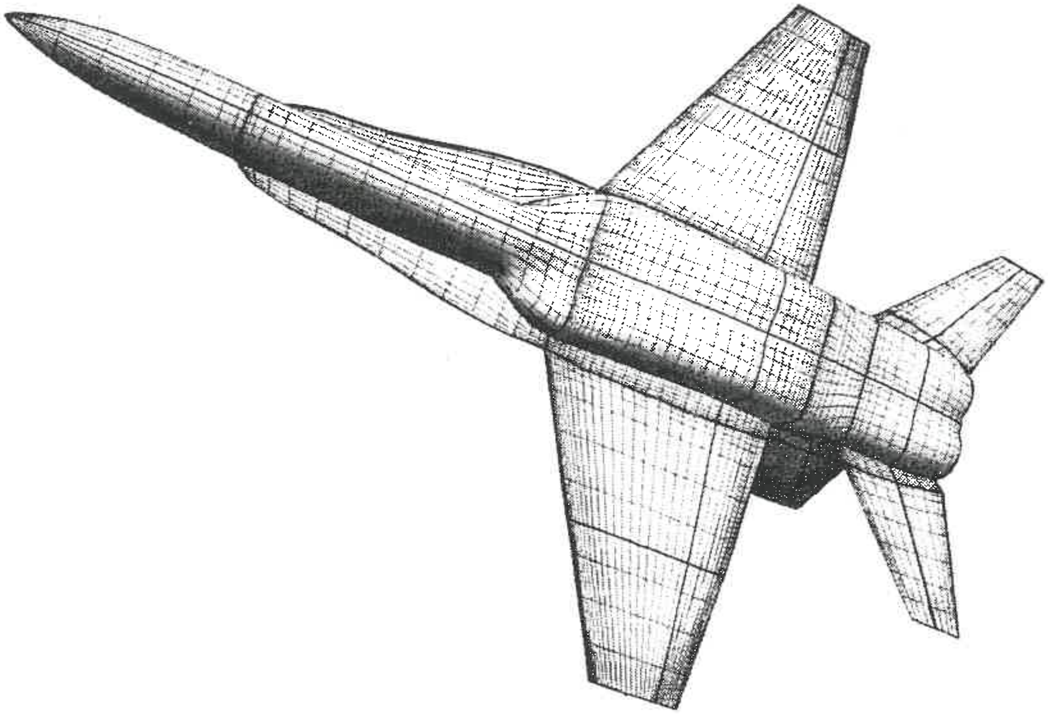


Figure 83(b): CF-18 aircraft lower surface grid (intermediate).

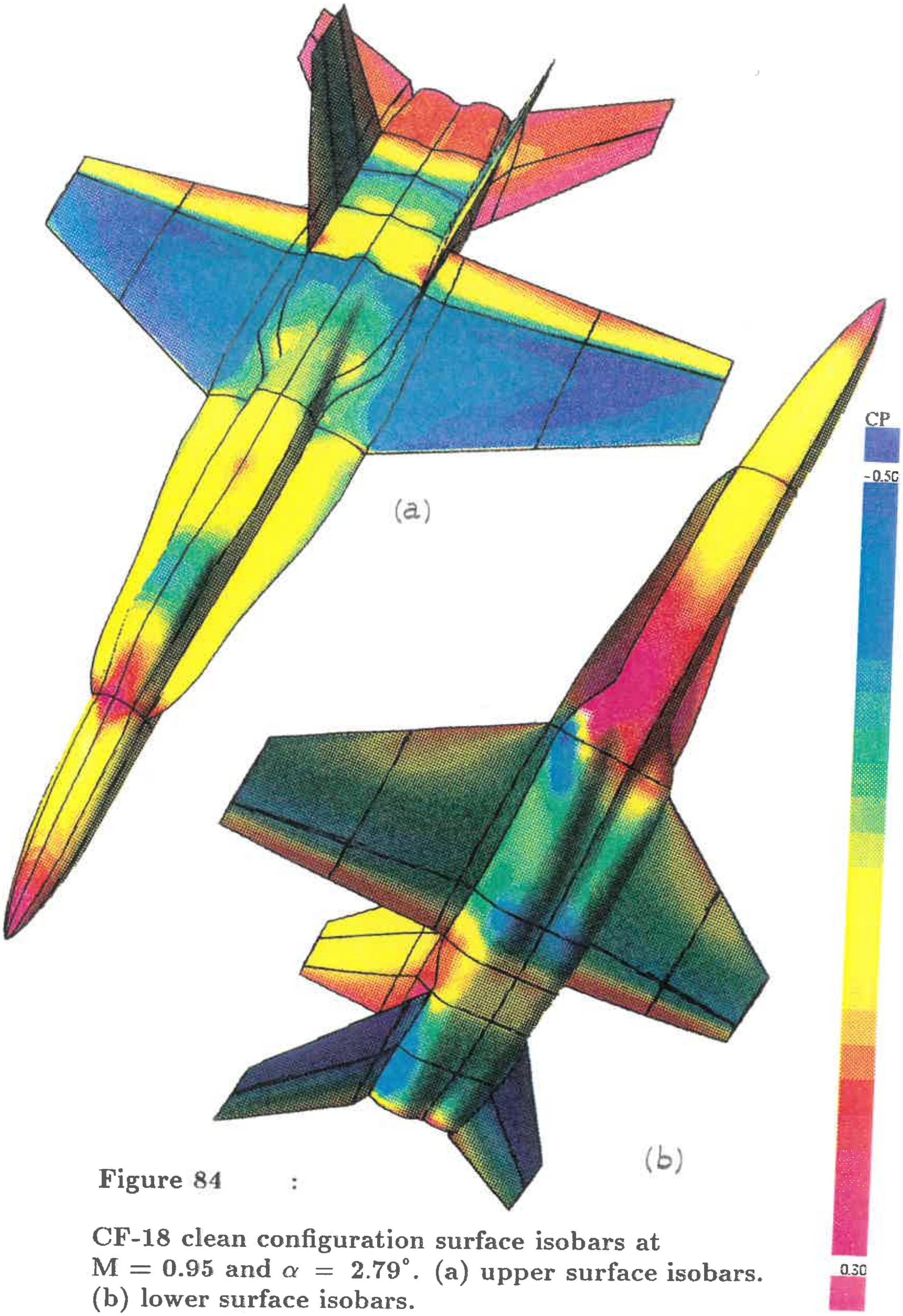


Figure 84 :

CF-18 clean configuration surface isobars at $M = 0.95$ and $\alpha = 2.79^{\circ}$. (a) upper surface isobars. (b) lower surface isobars.

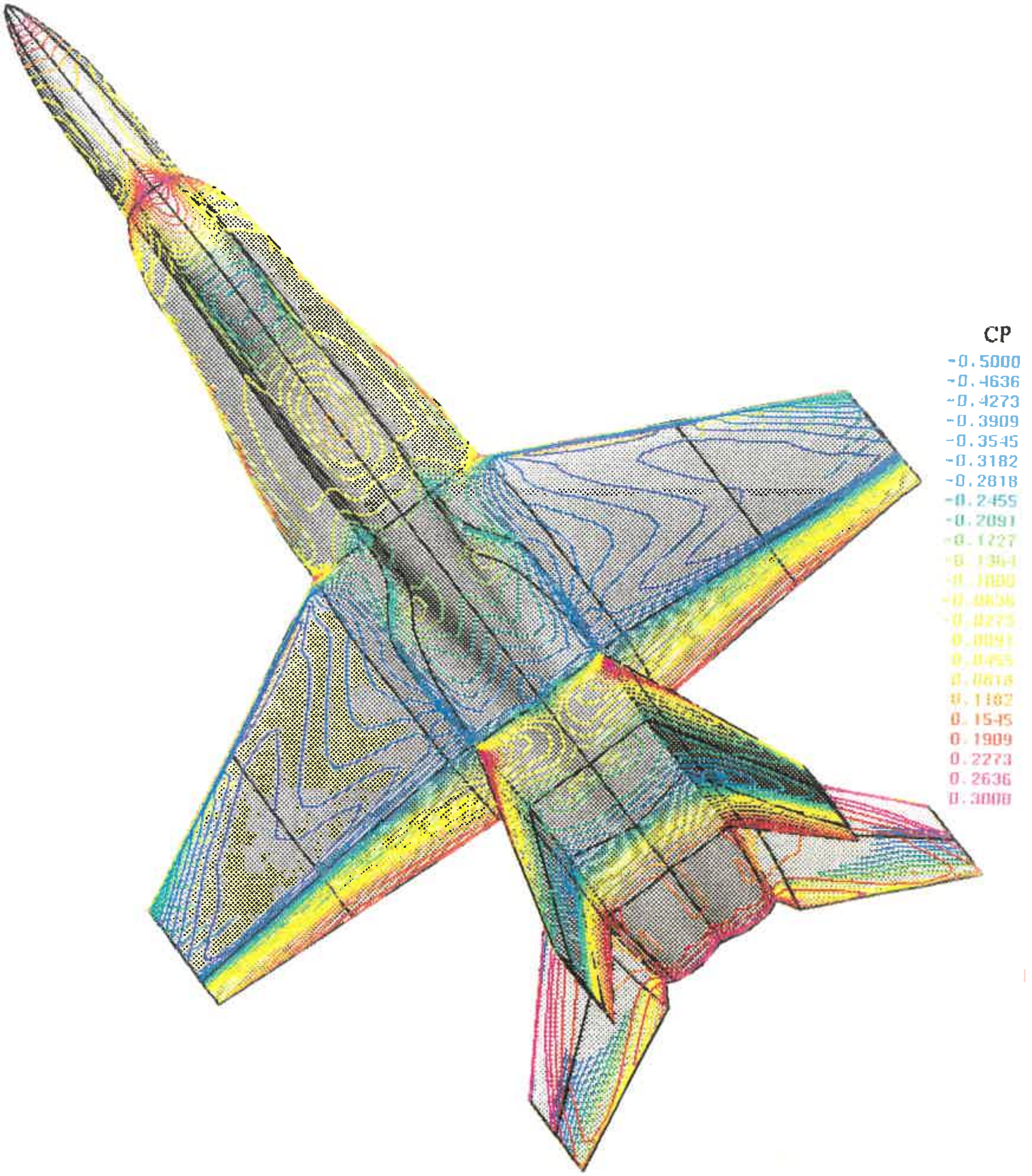


Figure 85 :

CF-18 clean configuration upper surface isobars at $M = 0.95$ and $\alpha = 2.79°$. With outlined block edges.

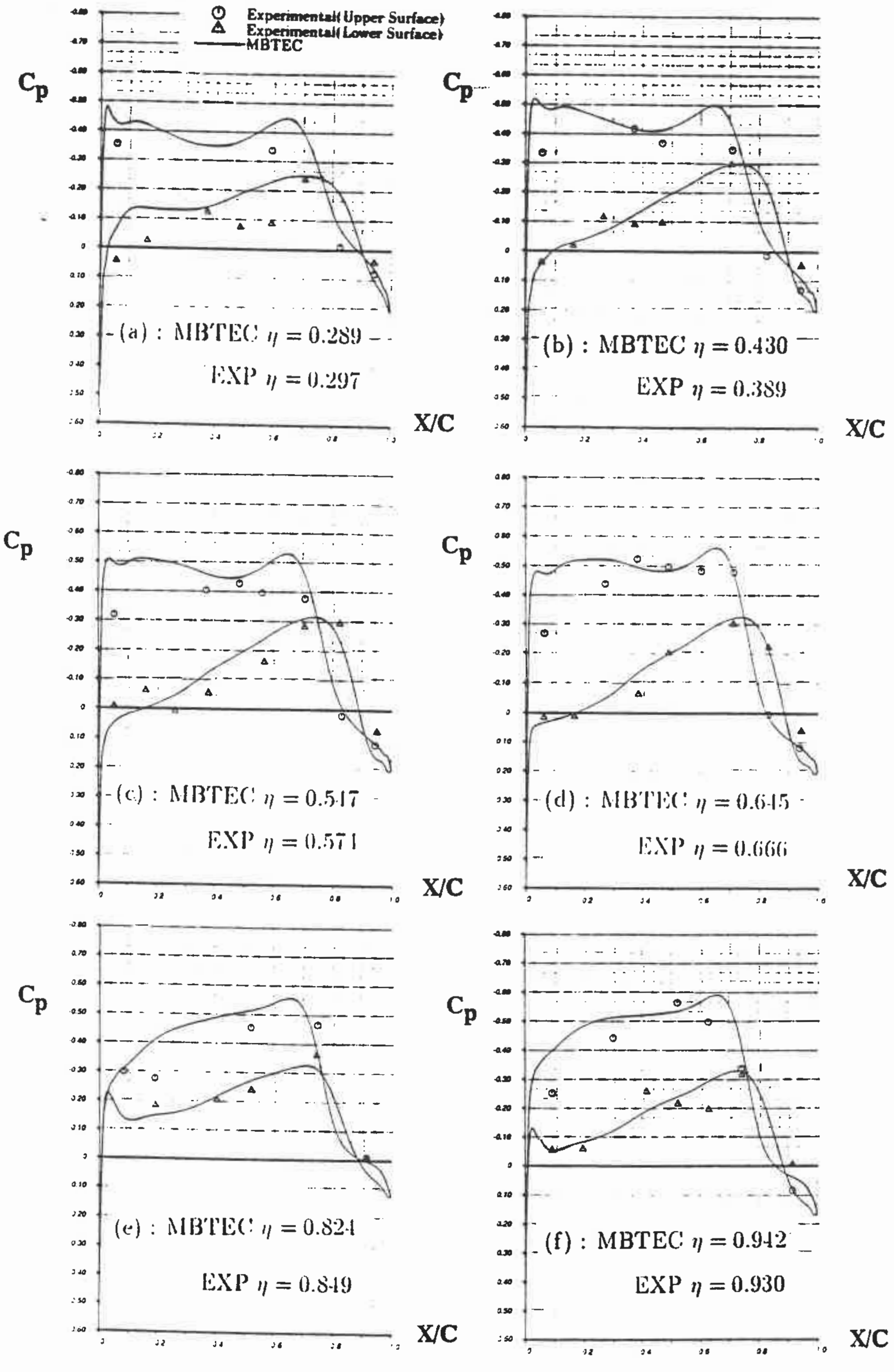


Figure 86 :

CF-18 clean configuration chordwise pressure distributions at six spanwise stations. $M = 0.95$ and $\alpha = 2.79^\circ$.

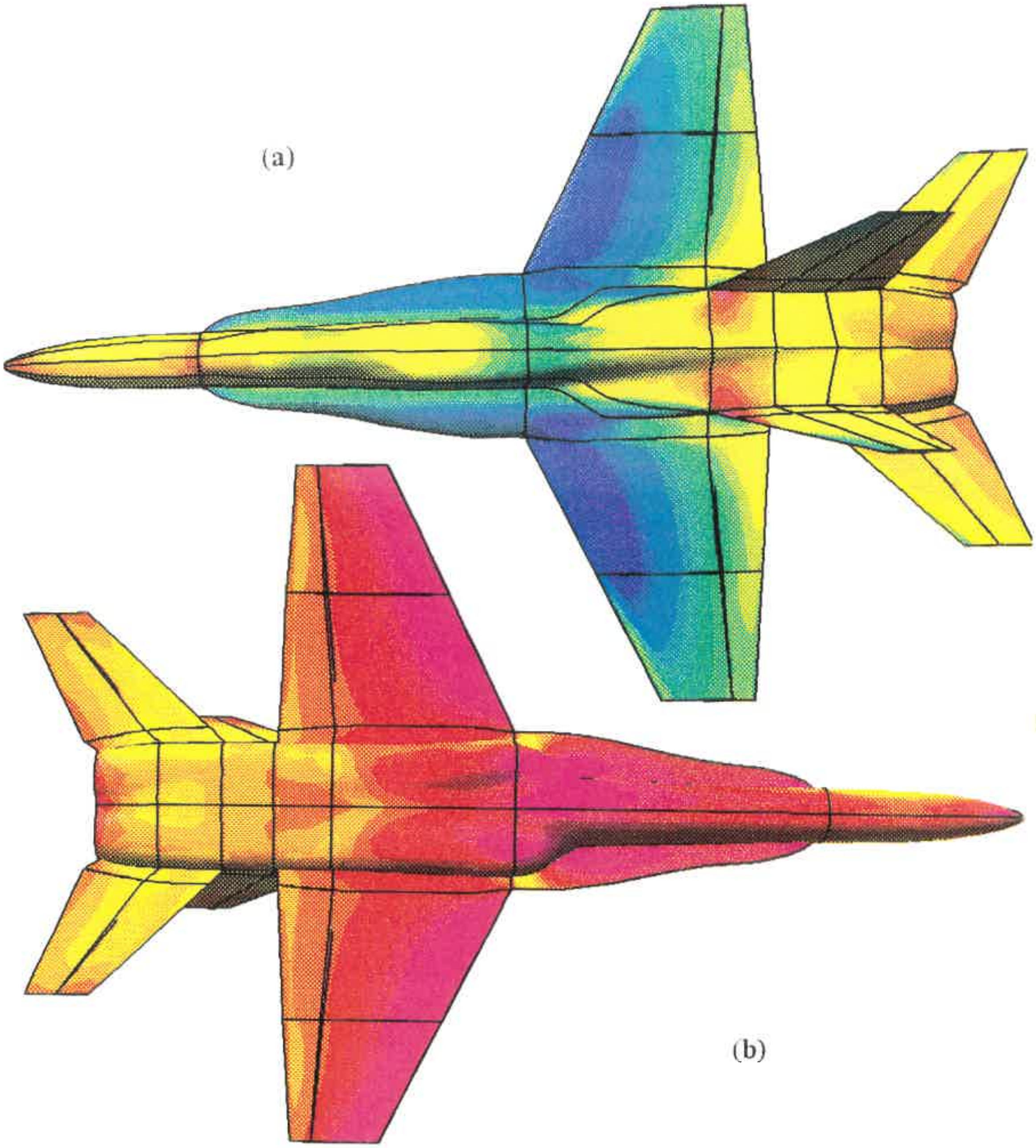
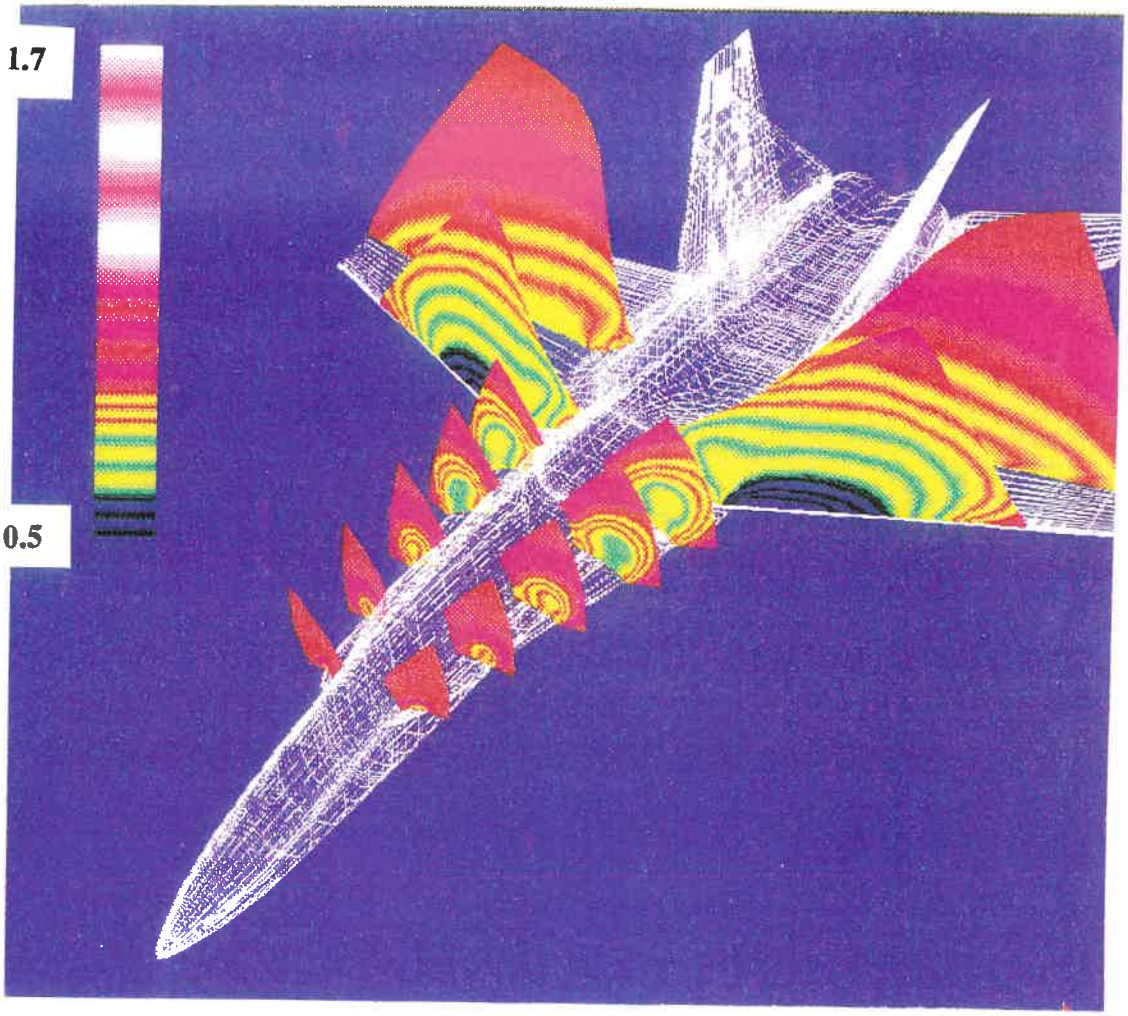


Figure 87 :

CF-18 clean configuration surface isobars at $M = 0.60$ and $\alpha = 16.9^{\circ}$. (a) upper surface isobars. (b) lower surface isobars.



CF-18 CLEAN CONFIGURATION
Mach 0.60 Alpha = 16.9 degrees
MBTEC solution 200 + 400 iterations
Normalized Stagnation Pressure

Figure 88 :

CF-18 clean configuration normalized stagnation pressures showing vortices emanating from the leading edge extensions. $M = 0.60$ and $\alpha = 16.9°$.

7.9 COMPUTING COSTS, PRE AND POST PROCESSING

The **MBTEC** 3-D multi-block Euler code requires 100 Megabytes of memory to compute the Euler solution for a grid with 800,000 nodes. The development of the program was supported by a significant increase of the computational capability of Canadair which occurred in mid-1990, when the company acquired the CONVEX C-220 vector computer with two 50-Megaflops CPU's with 128 Megabytes of central memory. This memory is large enough to allow core memory storage of all the variables. The code is written to take full advantage of the Convex Vectorizing FORTRAN compiler and has been modified for preliminary parallel computation on the two CPU's. The solution of the CF-34 turbofan flow field reported here required 8 hours and 30 minutes on a single CPU (100 + 100 + 500 iterations). A complete network was established around the CONVEX, as shown in figure 89. The work begins with the modelling of the surface geometry on a CADAM workstation attached to the AMDAHL 5890 scalar mainframe computer. The geometry and topology information is then retrieved and sent to the CONVEX via the company Ethernet backbone to perform the multi-block grid generation. The coordinates of the grid are then downloaded on one of the Silicon Graphics workstation connected to the Advanced Aerodynamics Laboratory local area network. The grid is visualized and inspected on the workstation using the Sterling Software SSV-4D program. The Euler calculations are performed on the CONVEX. The preparation of the data and the analysis of the numerical results is done on X-terminals connected to the Silicon Graphics 4D/310-R4000 working as a server and through it to the Ethernet network and eventually to the CONVEX. The **MBTEC** code outputs the coordinates and flow variables of all the node points in all blocks in a large data file. This file is transferred to a Silicon Graphics workstation and input to the SSV-4D program. It is thus possible to display the grid and any other aerodynamic characteristic of the flow field which can be computed from the conservative variables (velocity vectors, streamlines, Mach number, temperature, entropy, stagnation temperature and pressure, total enthalpy, total energy, etc.). Color hardcopy plots can be obtained with the four-pass thermal plotter connected to the network. The code also outputs a file with the surface geometry and surface pressure distribution. This file is used to visualize the results using the OMNI3D software from Analytical Methods Inc. OMNI3D allows isobars to be displayed as well as pressure distributions on arbitrary sections. A third output file containing sectional pressure data on wing and fuselage sections is sent to the AMDAHL computer where a variety of Canadair written programs is used to produce two-dimensional plots on an electrostatic machine.

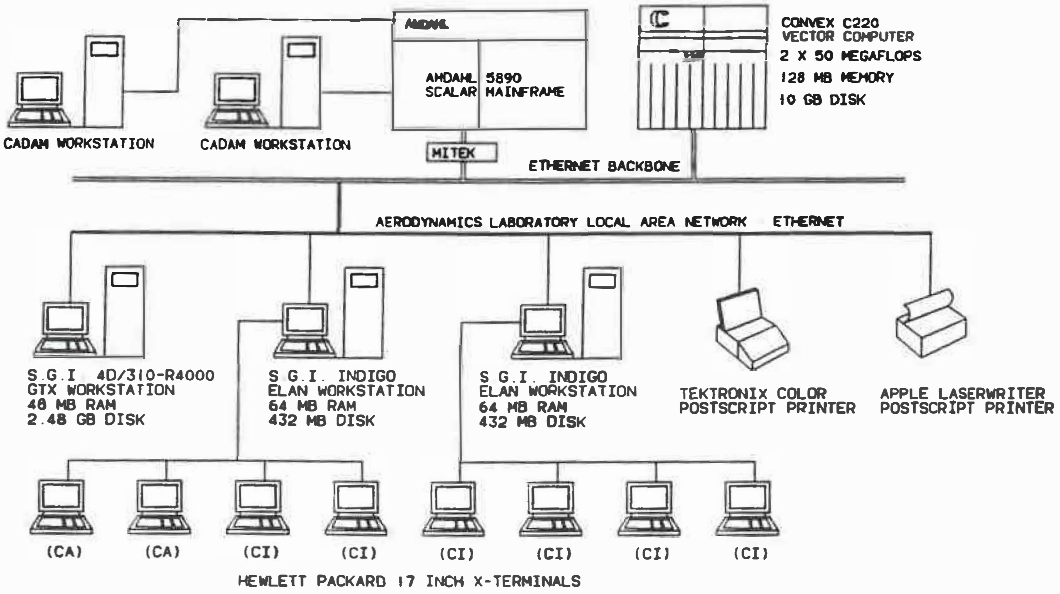


Figure 89 : Canadair CFD computing setup

CHAPTER 8 : Conclusions and Further Work

8.1 CONCLUSIONS

This work has demonstrated a new approach for efficient multi-block solutions of the Euler equations around arbitrary aircraft configurations. A complete method was proposed, using multi-block body-fitted structured grids, a finite volume discretization and a fully explicit solution. An original treatment of the block interface boundary conditions was proposed, designed to maintain the highest accuracy of the solution. This formulation, using two planes of halo cells in neighboring blocks, introduces an additional computational overhead but it renders block interfaces virtually transparent to the flow. The existence of these halo cells introduces difficulties at block corners near solid surface junctions. A careful treatment of the convective and dissipative fluxes in these areas of the flow field was necessary to ensure stable calculations. This formulation of the interface boundary conditions guarantees :

1. That the convergence characteristics of the scheme do not deteriorate with an increase in the number of blocks.
2. That the solution remains stable even with strong shock waves traversing block boundaries.

The CL-601 Challenger configuration has fuselage-mounted high by-pass turbofan nacelles located in close proximity to the wing and affecting the flow over its upper surface. In order to obtain accurate solutions for the complete aircraft, it was necessary to pay a particular attention to the formulation of the nacelle inlet and exhaust boundary conditions. Several formulations were examined, coded and tested, using a single-block version of the Euler code. An original way of specifying nacelle exhaust boundary conditions was proposed, using an approach suggested by non-reflecting far-field boundary conditions used at the outer boundaries of the domain. Extensive testing showed that our boundary conditions lead to stable and converging calculations in all cases except for combinations of an extremely low inlet Mass Flow Ratio and an unusually high nacelle incidence. The accuracy and versatility of the code was verified by comparing theoretical results with experimental data on several wings and with wind tunnel test and flight test data on two completely different aircraft configurations. This work has also produced the very first Euler solutions for the Canadair CL-601 complete aircraft configuration. These solutions provide details of the flow characteristics around the aircraft with an accuracy which was previously unattainable on complete configurations in transonic flow. The VSAERO panel method and the KTRAN Transonic Small Disturbance method previously used at Canadair had restrictions in this range of the flight envelope. With the body-fitted structured grid formulation, the use of conservative equations and discretizations and the high accuracy block-interface boundary conditions, the **MBTEC** code provides solutions for inviscid transonic flows of similar accuracy than those obtained on simpler geometries with single-block codes using central schemes with independent time integration. To our knowledge, the **MBTEC** program has also produced the first Euler solutions for a complete CF-18 aircraft obtained in Canada. This exercise indicated that the code can model vortical flows but additional correlations with experimental data are needed. In addition to the solution method for the Euler equations, we have established a complete procedure for analyzing transonic flows around complex configurations : modelling of the aircraft geometry on CADAM, generation of grids with programs written at Canadair specifically for the needs of this project, solution of the Euler equations and analysis of results on workstations. The code was tested on configurations with increasing levels of complexity, in a manner which ensured that most potential problems were addressed properly. As a result, the **MBTEC** program has consistently demonstrated accurate calculations on complex aircraft configurations including powered nacelles. The development of the 3-D Euler code was accompanied by the parallel development of 2-D and 3-D algebraic and elliptic multi-block grid generation tools, described in reference [100]. The requirements of the Euler codes have led Canadair to update significantly its computational and graphics facilities. This was done through the acquisition of a CONVEX vector computer and that of multiple workstations, printers and plotters, all connected through local and global high-speed networks. Efficient grid and flow visualization software were also brought into the company for this project. The Euler code has become the principal tool for verifying the design of the rear fuselage, the integration of fuselage-mounted nacelles and the design of nacelle pylons for all future Canadair airplanes. The code is used on a daily basis in the design of the Global Express high speed long range Business Jet.

8.2 FURTHER WORK

Further improvements in the rate of convergence of the code will be obtained by introducing a multigrid algorithm. In an experimental setup, the code was modified to run concurrent block solutions on the two parallel CPU's of the CONVEX C-220. In some cases a peak parallel factor of 1.8 was reached. Improvements in total turnaround time can be obtained by programming concurrent solutions of different blocks on multiple CPU's. A strip-wise interactive boundary layer was linked to the code and is presently being tested. This will offer the possibility of performing coupled Euler/Boundary Layer calculations on all surfaces of the aircraft. the MBTEC program is now being used as the starting point for the development of a Navier-Stokes program for complete aircraft configurations [12]. This will be done by the modelling of physical viscous fluxes and the introduction of a turbulence model for high Reynolds number viscous flow.

CHAPTER 9 : References

[1] B. MASKEW & F.A. WOODWARD : "Symmetrical Singularity Model For Lifting Potential Flow Analysis", Journal of Aircraft, Vol. 13, No. 9, September 1976.

[2] P. BANSOD & F. KAFYEKE : "The Development of an Improved First Order Panel Method to Calculate the Flow Over Wing-Body Combinations ", Canadair Report RAZ-000-497, March 1983.

[3] J.T. CONWAY : "Enhancement of the Canaero Computer code to Include Body Separation Vortices", Canadair Report RAZ-000-536, Vol. I & II, April 1988.

[4] L.A. CARLSON : "TRANDES : A Fortran Program for Transonic Airfoil Analysis or Design", NASA CR-2821, June 1977.

[5] F. BAUER, P. GARABEDIAN and D. KORN : "Supercritical Wing Sections", Lecture Notes in Economics and Mathematical Systems, No. 66, Springer-Verlag, 1972.

[6] F.KAFYEKE, P. PIPERNI & F. PÉPIN : "Development of Advanced CFD Methods for Analysis and Design of Aircraft Aerodynamic Configurations", Canadair Report RAZ-000-557, May 1991.

[7] A. JAMESON & D.A. CAUGHEY : "Numerical Calculation of the Transonic Flow Past a Swept Wing", NASA CR-153297, 1977.

[8] F. MAVRIPLIS : "Design and Development of a Transonic Wing for the Canadair Regional Jet", CASI Journal, Vol. 37, No. 4, December 1991.

[9] F. KAFYEKE : "An Analysis Method for Transonic Flows About Three Dimensional Configurations", Canadair Report RAZ-000-516, February 1986.

[10] P. PIPERNI & F. KAFYEKE : "Application of the KTRAN Transonic Small Disturbance Code to the Complete CF-18 With Stores", Proceedings of the First Canadian Symposium on Aerodynamics, December 1989.

[11] F. KAFYEKE, P. PIPERNI & S. ROBIN : "Application of KTRAN Transonic Small Perturbation code to the Challenger Business Jet Configuration with Winglets", ASME Paper 881483, published in SAE Conference Proceedings SP-757, October 1988.

[12] F. KAFYEKE, F. MOKHTARIAN, P. PIPERNI : "A proposal for the Development of Advanced Navier-Stokes Methods for the Analysis of Aircraft Aerodynamic Configurations", Canadair Report RAZ-000-567, October 1992.

[13] C.HIRSCH : "Numerical Computation of Internal and External Flows", Vol I. & II, John Wiley & Sons, 1988.

[14] Klaus. A. HOFFMAN : "Computational Fluid Dynamics for Engineers", Engineering Education System, Austin, Texas, 1989.

[15] R. MAGNUS & H. YOSHIHARA : "Inviscid Transonic Flow Over Airfoils", AIAA Journal, 8, pp. 2157-2162, 1970.

[16] R.W. MACCORMACK : "The effect of Viscosity in Hyper-Velocity Impact Cratering", AIAA Paper 69-354, 1969.

[17] A. JAMESON : "Successes and Challenges in Computational Aerodynamics", AIAA Paper 87-1184-CP, 1987.

[18] M. GILES, M. DRELA and W.T. THOMPKINS : "Newton Solutions of Direct and Inverse Transonic Euler Equations", AIAA Paper 85-1530, 1985.

[19] M.O. BRISTEAU, R. GLOWINSKI, J. PERIAUX, P. PERRIER, O. Pironneau, G. Poirier : "On the Numerical Solution of Nonlinear Problems in Fluid Dynamics by Least Square and Finite Element Methods", in "Application to Transonic Flow Simulations,

Computer Methods in Applied Mechanics and Engineering", 51, 1985, pp. 363-394.

[20] P.D. LAX : "Weak Solutions of Non Linear Hyperbolic Equations and their Numerical Computations", Comm. Pure and Applied Mathematics, No.7, pp. 159-193, 1954.

[21] P.D. LAX & B. WENDROFF : "Difference Schemes for the Hyperbolic Equations with High Order of Accuracy", Comm. in Pure and Applied Mathematics, No. 17, pp. 381-398, 1964.

[22] F. KAFYEKE : "Calcul numérique de grilles d'aubes en régime transsonique (par une méthode quasi-naturelle)", Université de Liège (Belgium), July 1980.

[23] J.A. ESSERS & F. KAFYEKE : "Application of a Fast Pseudo Unsteady Method to Steady Transonic Flows in Turbine Cascades", ASME Paper 81-GT-124, March 1981.

[24] K.M. PEERY & C.K. FORRESTER : "Numerical Simulation of Multistream Nozzle Flows", AIAA Journal, Vol. 18, No. 9, September 1979, or AIAA Paper 79-1549, July 1979.

[25] C. KOECK : "Computations of Three-dimensional Flow Using the Euler Equations and a Multiple-grid Scheme", International Journal for Numerical Methods in Fluids, Vol. 5, pp. 483-500, 1985.

[26] R.H. NI : "A Multiple-Grid Scheme for Solving the Euler Equations", AIAA Paper 81-1025, June 1981.

[27] R.V. CHIMA & Gary JOHNSON : " Efficient Solution of the Euler and Navier-Stokes Equations with a Vectorized Multiple-Grid Algorithm", AIAA Journal Vol. 23, No. 1, January 1985.

[28] A. RIZZI : "Damped Euler-Equation Method to Compute Transonic Flow around Wing-Body Combinations", AIAA Journal, Vol. 20, No. 2, October 1982.

[29] A. LERAT : "Implicit methods of second-order accuracy for the Euler Equations", AIAA

Journal, 23, 33-40, 1985.

[30] R.M. BEAM & R.F. WARMING : "An Implicit Finite-Difference Algorithm for Hyperbolic Systems in Conservation Law Form", Journal of Computational Physics, Vol. 22, pp. 87-110, Sept 1976.

[31] R.M. BEAM & R.F. WARMING : "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations", AIAA Journal, Vol.16, pp. 393-402, April 1978.

[32] R.M. BEAM & R.F. WARMING : "Implicit Numerical Methods for the Compressible Navier-Stokes and Euler equations", Von Karman Institute Lecture Series 1982-04, Rhode Saint Genèse, Belgium, 1982.

[33] T. H. PULLIAM and J.T. BARTON : "Euler Computations of AGARD Working Group 07 Airfoil Test Cases", AIAA Paper 85-0018, January 1985.

[34] T. H. PULLIAM and D.S. CHAUSSEE : "A Diagonal Form of an Implicit Factorization Algorithm", Journal of Computational Physics, No. 39, pp. 347-363, 1981.

[35] J. VADYAK : "Simulation of Transonic Three-Dimensional Nacelle/Inlet Flowfields Using an Euler/Navier-Stokes Algorithm", AIAA Paper 85-0084, January 1985.

[36] A. JAMESON & E. TURKEL : "Implicit schemes and LU Decompositions", Math. Comp., 37, pp. 385-397, 1981.

[37] A. JAMESON, W. SCHMIDT and E. TURKEL : "Numerical Simulation of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes", AIAA Paper 81-1259, 1981.

[38] P. RAJ : "An Euler Code for Nonlinear Aerodynamic Analysis : Assessment of Capabilities", SAE Paper 881486, October 1988.

[39] C.R. OLLING anf K.KURIAN MANI : "Navier-Stokes and Euler Computations Around Complete Aircraft", SAE Paper 881488, October 1988.

[40] A. SHMILOVICH & K.C. CHANG : "An algorithm for Predicting the Flow Past Fuselage-Mounted Engine Arrangements", AIAA Paper 92-0151, January 1992.

[41] J. BUSCH, Jr., M. JAGER & B. BERGMAN : "The Application of CFD to Aircraft Design", AIAA Paper 86-2651, October 1986.

[42] P.A. SHEPHERD and G.R. TOD : "Development and Application of a Weapons Multiblock Suite., AGARD CP-??? paper 23."

[43] B. EPSTEIN, A.L. LUNTZ & A. NACHSHON : "Cartesian Euler Method for Arbitrary Aircraft Configurations", AIAA Journal, Vol. 30, No. 3, March 1992.

[44] H.C. CHEN, N.Y. YU, P. RUBBERT & A. JAMESON : "Flow Simulation for General Nacelle configurations Using Euler Equations", AIAA Paper 83-0539, January 1983.

[45] J.C. VASSBERG, K.B. DAILEY : "Airplane : Experiences, Benchmarks and Improvements", AIAA Paper 90-2998-CP, August 1990.

[46] J.L. STEGER & R.F. WARMING : "Flux Vector Splitting of the Inviscid Gas Dynamics Equations with Applications to Finite Difference Methods", Journal of Computational Physics, 40, pp. 263-293, 1981.

[47] B. VAN LEER : "Flux Vector Splitting for the Euler Equations", Proc. 8th Internal Conference on Numerical Methods in Fluid Dynamics, Berlin; Springer-Verlag, 1982.

[48] S.K. GODUNOV : "A Difference Scheme for Numerical Computation of Discontinuous Solution of Hydrodynamic Equations", Math. Sbornik, No.47, pp. 271-306, 1959 (in Russian) - Translated US Joint Publ. Res. Service, JPRS 7226 (1969).

[49] P.L. ROE : "Approximate Riemmann Solvers, Parameter Vectors and Difference Schemes", Journal of Computational Physics, 43 pp. 357-372, 1981.

[50] S. OSHER : "Numerical Solution of Singular Perturbation Problems and Hyperbolic Systems of Conservation Laws", in O. Axelsson et al (eds) : "Mathematical Studies",

North Holland No. 47, pp. 179-205, 1981.

[51] R. COURANT, E. ISAACSON and M. REEVES : "On the Solution of Nonlinear Hyperbolic Differential Equations by Finite Differences", Comm. Pure and Applied Mathematics, No. 5, pp. 243-255, 1952.

[52] W.K. ANDERSON & J.T. BATINA : "Accurate Solutions, Parameter Studies, and Comparisons for the Euler and Potential Flow Equations", AGARD CP-437, Vol. I, 1988.

[53] B. VAN LEER, J.L. THOMAS, P. ROE and R. NEWSOME : "A Comparison of Numerical Flux Formulas for the Euler and Navier- Stokes Equations", AIAA Paper 87-1104-CP, 1987.

[54] P.L. ROE : "The Use of the Riemann Problem in Finite Difference Schemes", in "Lecture Notes in Physics", No. 141, pp. 354-359, 1980.

[55] J.Y. TRÉPANIER, M. REGGIO, H. ZHANG and R. CAMARERO : "A Finite Volume Method for Solving Euler Equations Using Arbitrary Lagrangian-Eulerian Grids",Computers and Fluids, 1991.

[56] M. PARASCHIVOIU : "Unsteady Euler Solution for Oscilatory Airfoil and Oscillating Flap", AIAA Paper 92-0131, January 1992.

[57] A. EBERLE : "3D-Euler Calculations Using Characteristic Flux Extrapolation", AIAA paper 85-0119, January 1985

[58] A. HARTEN & P.D. LAX : "A Random Choice Finite Difference Scheme for Hyperbolic Conservation Laws", SIAM Journal of Numerical Analysis, No. 18, pp. 289-315.

[59] A. HARTEN : " High Resolution Schemes for Hyperbolic Conservation Laws", Journal of Computational Physics, 49 pp. 357-393, 1983.

[60] B. VAN LEER : "Towards the Ultimate Conservative Difference Scheme. I. The Quest

of Monotonicity", Lecture Notes in Physics, Vol. 18, pp. 163-168, Springer-Verlag, Berlin, 1973.

[61] J.P. BORIS and D.L. BOOK : "Flux Corrected Transport : I. SHASTA, a Fluid Transport Algorithm that Works", Journal of Computational Physics, No. 11, pp. 38-69, 1973.

[62] Y. TAKAKURA, T. ISHIGURO & S. OGAWA : "On the Recent Difference Schemes for the Three-dimensional Euler Equations", AIAA paper 87-1151, June 1987.

[63] M.Y. HUSSAINI, D.A. KOPRIVA, M.D. SALAS and T.A. ZANG : " Spectral Methods for the Euler Equations : Part I - Fourier Methods and shock capturing", AIAA Journal Vol. 23 No. 1, January 1985.

[64] M.Y. HUSSAINI, D.A. KOPRIVA, M.D. SALAS and T.A. ZANG : " Spectral Methods for the Euler Equations : Part II - Chebyshev Methods and Shock fitting", AIAA Journal Vol. 23 No. 2, January 1985.

[65] J. Von NEUMANN & R.D. RICHTMYER : "A method for the numerical calculation of hydrodynamical shocks", Journal of Mathematical Physics, No. 21, 1950.

[66] T.H. PULLIAM & J.L. STEGER : "Recent Improvements in Efficiency, Accuracy and Convergence for Implicit Approximate Factorization Algorithms", AIAA Paper 85-0360, January 1985.

[67] D.A. CAUGHEY & E. TURKEL : "Effects of Numerical Dissipation on Finite Volume Solutions of Compressible Flow Problems", AIAA Paper 88-0621, January 1988.

[68] David A. CAUGHEY : "Diagonal Implicit Multigrid algorithm for the Euler Equations, AIAA Journal, Vol. 26, pp-841-851, July 1988.

[69] R.C. SWANSON & E. TURKEL : "Artificial Dissipation and Central Difference Schemes for the Euler and Navier-Stokes Equations", AIAA Paper 87-1107, June 1987.

[70] E. TURKEL : "Accuracy Versus Convergence Rates for a Three-dimensional Multistage Euler Code", ICAS Paper 88-4.6.1, 1988

[71] E. TURKEL & V.N. VATSA : "Effect of Artificial Viscosity on Three Dimensional Flow Solutions", AIAA Paper 90-1444, June 1990.

[72] L. MARTINELLI : "Calculations of Viscous Flows with a Multigrid Method", Ph. D. Thesis, Princeton University, 1987.

[73] B. ENGQUIST and A. MAJDA :"Radiation Boundary Conditions for the Numerical Simulation of Waves", Comm. Pure and Applied Mathematics, No. 32, pp. 629-651.

[74] A. BAYLISS & E. TURKEL : "Outflow Boundary Conditions for Fluid Dynamics", SIAM Journal of Scientific and Statistical Computing, Vol. 3, No. 2, June 1982.

[75] D.L. MARCUM & J.D. HOFFMAN : "Numerical Boundary Condition Procedures for Euler Solvers", AIAA Paper 86-0107, January 1986.

[76] J.L. THOMAS & M.D. SALAS : "Far-field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations", AIAA Journal, Vol. 24, No. 7, july 1986.

[77] M.B. GILES : "Non reflecting Boundary Conditions for Euler Equation Calculations", AIAA Journal Vol. 28, No. 2, 1989.

[78] A. RIZZI and H. VIVIAND (eds) : Proceedings of the GAMM workshop on Numerical Methods for the Computation of Inviscid Transonic Flows with Shock Waves. Notes on Numerical Fluid Dynamics, Vol. 3, Braunshweig/Wiesbaden : Vieweg, 1981.

[79] H.C. CHEN, K. KUSUNOSE and N.J. YU : Flow Simulation for Detailed Nacelle-Exhaust Flow Using Euler Equations", AIAA Paper 85-5003, October 1985.

[80] N. HIROSE, K. ASAI, K. IKAWA & R. KAWAMURA : "3-D Euler Flow Analysis of Fanjet Engine and Turbine Powered Simulator with Experimental Comparison in Transonic Speed", AIAA Paper 89-1835, June 1989.

[81] E. KRAMER, J. HERTEL & S. WAGNER : "Euler Procedure for the Calculation of the Steady Rotor Flow with Emphasis on Wake Evolution", AIAA Paper 90-3007, August 1990.

[82] E.J. HALL & R.A. DELANEY : "3-D Euler Analysis of Ducted Propfan Flowfields. AIAA Paper 90-3034, August 1990.

[83] J.L. KUIJVENHOVEN : "Validation of Propeller Slipstream Calculations Using a Multi-Block Euler Code", AIAA Paper 90-3035, August 1990.

[84] A. AMENDOLA, R. TOGNACCINI, J.W. BOERSTOEL and A. KASSIES : "Validation of a Multi-Block Euler Flow Solver with Propeller Slipstream Flows", AGARD CP-437-P1, May 1988.

[85] S. AGRAWAL, R.M. BARNETT & B.A. ROBINSON : "Numerical Investigation of Vortex Breakdown on a Delta Wing", AIAA Journal, Vol. 30, No. 3, March 1992.

[86] J.I. VAN DEN BERG, H.W.M. HOEIJMAKERS & H.A. SYTSMA : "Numerical Investigation Into High-Angle-of-Attack Leading Edge Flow", AIAA Paper 92-2600, June 1992.

[87] A. HILGENSTOCK : "Validation of Transonic Turbulent Flows Past Delta Wing Configurations", The Aeronautical Journal, Vol. 95, No. 947, August/September 1991.

[88] A. RIZZI & B. MULLER : "Comparison of Euler and Navier-Stokes Solutions for Vortex Flows over a Delta Wing", The Aeronautical Journal, Vol. 92, No. 914, April 1988.

[89] A. JAMESON, T.J. BAKER & N.P. WEATHERILL : "Calculations of Inviscid Transonic Flow over a Complete Aircraft", AIAA Paper 86-0103, January 1986.

[90] D.M. TIDD, D.J. STRASH, B. EPSTEIN, A. LUNTZ, A. NACHSHON and T. RUBIN : "Application of an Efficient 3-D Multigrid Euler Method (MGAERO) to Complete Aircraft Configurations", AIAA Paper 91-3236, September 1991.

[91] R. COLLERCANDY : "An Improved Approach for the Computation of Transonic/Supersonic Flows with Applications to Aerospace Configurations", AIAA Paper 92-2613, June 1992.

[92] J. REUTHER, S.E. ClIFF, R.M. HICKS & C.P. VAN DAM : "Practical Design Optimization of Wing/Body Configurations Using the Euler Equations", AIAA Paper 92-2633.

[93] J. SCHONE : "Design of Supersonic Wings Using an Optimization Strategy Coupled with a Solution Scheme for the Euler Equations", AIAA Paper 90-3060, August 1990.

[94] W.F. LIN, A.W. CHEN and E.N. TINOCO : "3D Transonic Nacelle and Winglet Design", AIAA Paper 90-3064, August 1990.

[95] M.G. HALL & S.P. FIDDES : " Computational Fluid Dynamics in the United Kingdom", AIAA Paper 87-1132, June 1987.

[96] P.L.ROE : " Error Estimates for Cell-Vertex Solutions of the Compressible Euler Equations', ICASE report 87-6, 1987.

[97] J.F. THOMPSON, Z.U.A. WARSI and C.W. MASTIN : "Numerical Grid Generation, Foundations and Applications", North-Holland, New-York, 1985.

[98] J.F. THOMPSON, Z.U.A. WARSI and C.W. MASTIN : "Boundary Fitted Coordinate Systems for Numerical Solution of Partial Differential Equations - A Review", Journal of Computational Physics, 47, 1, 1982.

[99] F. KAFYEKE, N. DJILALI, F. MOKHTARIAN, P. PIPERNI : "Development of Advanced CFD Methods for Analysis and Design of Aircraft Aerodynamic Configurations - First Milestone Report", Canadair Report RAZ-000-549, Volume 2, April 1990.

[100] P. PIPERNI : "Multi-Block Grid Generation for Complete Aircraft Configurations", Canadair Report RAZ-000-572, Vol. III, March 1993.

[101] A. JAMESON : "Transonic Flow Calculation for Aircraft", Lecture Notes in Mathematics, Numerical Methods in Fluid Dynamics, edited by F. Brezzi, Springer-Verlag, 1127, 1985, pp. 156-242.

[102] A. JAMESON : "Solution of the Euler Equations by a Multigrid Method ", Applied Math and Computation, Vol. 13, 1983, pp 327-356.

[103] S.S. SAMANT, N.J. YU :"Flow Prediction for Propfan Installation Effects on Transport Aircraft at Transonic Speeds", NASA Contactor Report CR-3954, January 1986.

[104] P. RAJ, C.R. OLLING and S.W. SINGER : "Application of Multizone Euler/Navier-Stokes Aerodynamic Methods to Aircraft Configurations", ICAS Paper 90-6.4.4, September 1990.

[105] A. KASSIES, R. TOGNACCINI : "Boundary Conditions for Euler Equations at Internal Block Faces of Multi-Block Domains Using Local Grid Refinement", AIAA Paper 90-1590, June 1990.

[106] S.M. HITZEL : "Wing Vortex Flows Up Into Vortex-Breakdown; a Numerical Simulation", IAAA Paper 88-2518-CP, June 1988.

[107] D.L. LOVING & B.B. ESTABROOKS : "Transonic Wing Investigation in the Langley 8-foot High-Speed Tunnel at High Subsonic Mach Numbers and at Mach Number of 1.2", NACA Research Memorandum RM L51F07, September 1951.

[108] V. SCHMITT and F. CHARPIN : "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers", AGARD Advisory Report No. 138, May 1979.

[109] F. MOKHTARIAN : "On the Coupling of Euler/Boundary-Layer Equations", CASI Journal, Volume 38, No. 3, September 1992.