NASA Contractor Report 4085

Supersonic Flow Computations Over Aerospace Configurations Using an Euler Marching Solver

Kuo-Yen Szema, Sukumar Chakravarthy,
and Vijaya Shankar
Rockwell International Science Center
Thousand Oaks, California

Prepared for
Langley Research Center
under Contract NAS1-15820

National Aeronautics
and Space Administration

**Scientific and Technical
Information Office**

1987

Foreword

This final report was prepared by the Science Center of Rockwell International, Thousand Oaks, California, for the Langley Research Center, National Aeronautics and Space Administration, Hampton, Virginia. The work was performed under Contract No. NAS1-15820, "Development of Full Potential and Euler Aero Prediction Methods for Supersonic/Hypersonic Configuration Design." Mr. Noel Talcott and Mr. Kenneth Jones were the contract monitors of this contract.

Mr. E. Bonner of the Los Angeles Division, Rockwell International, was the Program Manager; Drs. K.-Y. Szema, S.R. Chakravarthy, and V. Shankar of the Rockwell International Science Center were the Principal Investigators.

Summary

An Euler marching algorithm for computing supersonic flows was developed by Dr. Chakravarthy as part of a NASA-Langley Research Center contract (NAS1-17492). The objective of the present contract (NAS1-15820) is to apply that Euler methodology to compute supersonic flows over realistic fighter-like configurations using the geometry/grid generation package developed for a similar full potential capability known as the SIMP (Supersonic Implicit Marching Potential) code, whose development was also funded by Contract NAS1-15820.

The Euler marching capability is termed "EMTAC" (Euler Marching Technique for Accurate Computation). The EMTAC code and the SIMP code have been extensively validated against each other in the Mach number range where the isentropic assumption is valid. The EMTAC code, being based on the exact inviscid gasdynamic equations, is valid for low and high supersonic Mach number computations exhibiting strong shocks and rotational effects. However, the use of Euler methods for computing vortex dominated flows is still unresolved and needs further investigation.

Several AIAA papers have been written describing the EMTAC methodology with comparisons of Euler results with the SIMP code and experimental data. The Appendix section of this report includes several of these papers.

Contents

1. INTRODUCTION

For fully supersonic flows, an efficient strategy for obtaining numerical solutions is to employ space–marching techniques. At low supersonic Mach numbers, realistic fighter configurations give rise to subsonic pockets near the canopy, wing–body junction, wing leading edge, and wing tip regions. A full potential marching technique[1-4] capable of handling such embedded subsonic regions was developed as part of a NASA–Langley Research Center contract (NAS1–15820). The full potential method though very efficient for treating low supersonic Mach number flows (Mach number normal to a shock front is less than 1.3) is not capable of handling strongly shocked flows with rotational and vortex effects due to the underlying isentropic assumptions.

The objective of the present contract is to extend the full potential approach to the Euler equations which model the exact nonlinear inviscid gasdynamic flow processes. Within the assumption of an inviscid flow, such an Euler marching solver can be applied to a wide class of shocked flows including the hypersonic range. The intent of the Euler contract is to maintain some of the basic features of the full potential SIMP code[4] within the Euler solver in dealing with geometry input, gridding techniques, and input/output routines including post processing of results.

The algorithm for the Euler marching solver was developed by Chakravarthy[5] under a NASA contract, NAS1–17492. An Euler marching capability known as the "EMTAC" code ensuring compatibility with the full potential SIMP code has been developed. Results obtained for a variety of configurations involving canard, wing, horizontal tail, flow–through inlet, and fuselage using both the EMTAC and SIMP codes are reported in Refs. 5–9. Many of these papers are included in the Appendix of this report. For shocked cases satisfying the isentropic assumption ($M_n < 1.3$) with negligible entropy effects, the EMTAC and the SIMP codes produced practically identical results even for complex geometry configurations. In terms of execution time, the EMTAC code is about 5 to 10 times slower than the SIMP code since the Euler formulation solves five equations involving block tridiagonal inversions.

2. EULER METHOD

The Euler marching solver is described in detail in Ref. 5 and a copy of that paper is included in Appendix B.

Some of the salient features of the method are:

- Efficient space-marching technique based on unsteady Euler equations
- Finite volume upwind-biased scheme (modified Roe's approximate Riemann solver)
- High accuracy TVD formulation (up to third order)
- Approximate factorization in cross plane; forward marching for purely supersonic regions; Gauss-Seidel relaxation in marching direction for subsonic regions
- Proper treatment of wake-like grid topology
- Numerical grid generation (marching plane by marching plane)
- Nacelle treatment
- Code can also be easily used for inviscid 3-D flows which are fully subsonic or transonic (subsonic with supersonic pockets).

The EMTAC code is a single zone code just like the SIMP code. At present, the EMTAC code doesn't include the yaw capability for computing combined yaw and angle of attack cases (the SIMP code does). A multizone version of the EMTAC known as the EMTAC-MZ[9] is currently under development which will accommodate any number of computational zones with proper flux balancing treatment at zonal boundaries. Treatment of combined yaw and angle of attack cases can be handled with ease using the EMTAC-MZ multizonal capability. The EMTAC code is currently operational on the VPS-32 at NASA-Langley Research Center.

3. RESULTS

The geometry input format for the EMTAC code is the same as that for the SIMP code[4]. See Appendix A for details.

Results obtained using the EMTAC code for a number of configurations are reported in Refs. 5–9 and some are included in Appendix B. The following configurations have been successfully computed using both the EMTAC and the SIMP codes:

1) Forebody geometry with a subsonic canopy region (Fig. 1)

2) Fighter configuration with vertical tail and flow–through nacelle (Fig. 2)

3) Shuttle Orbiter (Fig. 3)

4) Waverider (Fig. 4)

5) Shuttle–like configuration (Fig. 5)

6) Canard–wing fighter with nacelle (Fig. 6)

7) Wing–horizontal tail fighter with nacelle (Fig. 7)

8) Wing–body–strake configuration (Fig. 8).

The results for Cases 1–3 are reported in Ref. 5. Cases 4 and 5 are presented in AIAA Paper 86–0244. Cases 6–8 are included in AIAA Paper 87–0592.

In addition to these results, the Euler code was also tested for computing flows with vortex features. Numerical issues in computing supersonic vortex flows over conical delta wings are discussed in Ref. 10 (AIAA Paper 86–0440). Appendix B includes this paper also. References 11 and 12 also report discussions relevant to the use of an Euler solver for computing vortex flows. Figure 9 shows results for a conical flat plate delta wing at $M_\infty = 2$, $\alpha = 10°$, $\Lambda = 70°$. Though Euler codes seem to produce the vortex features emanating from a sharp leading edge, computation of vortex flows around rounded leading edges still needs further study to understand the influence of numerical viscosity in predicting the correct location of the separation point.

AIAA Paper 86–1834[8], included in Appendix B, includes SIMP code results for combined yaw and angle of attack cases.

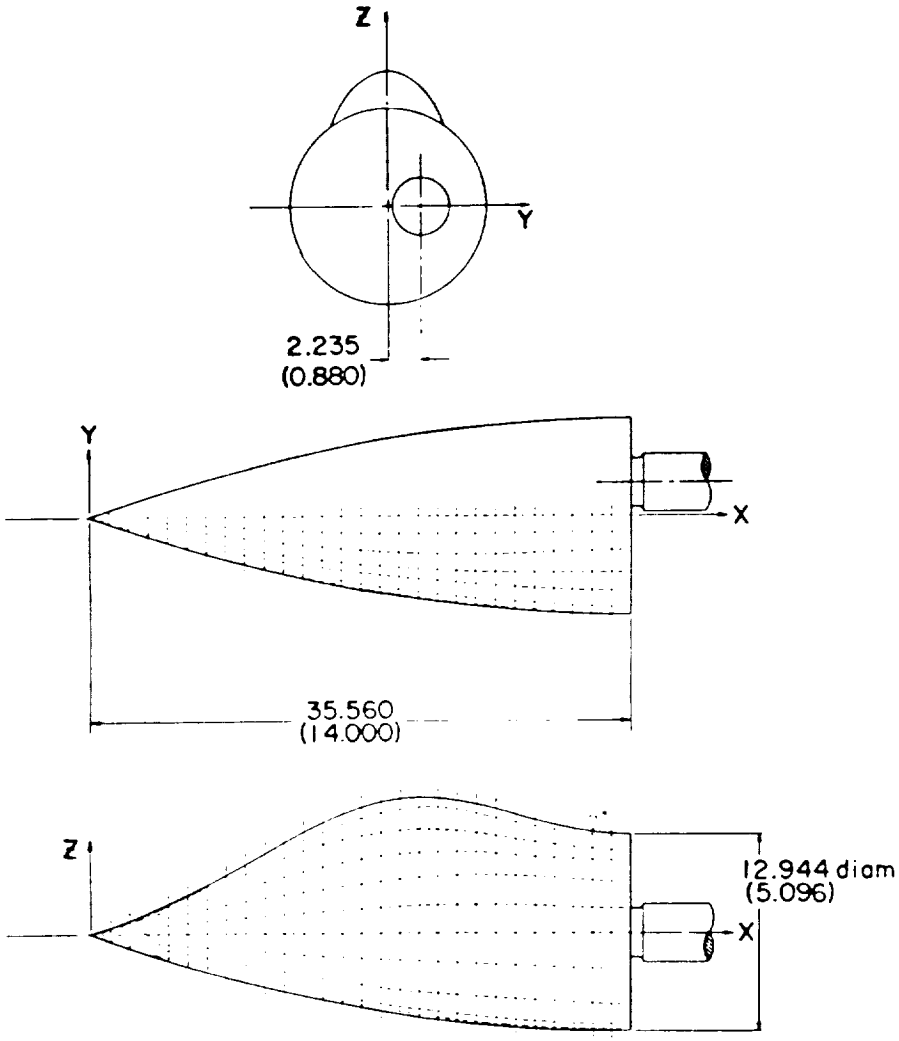


Fig. 1. Forebody canopy geometry.

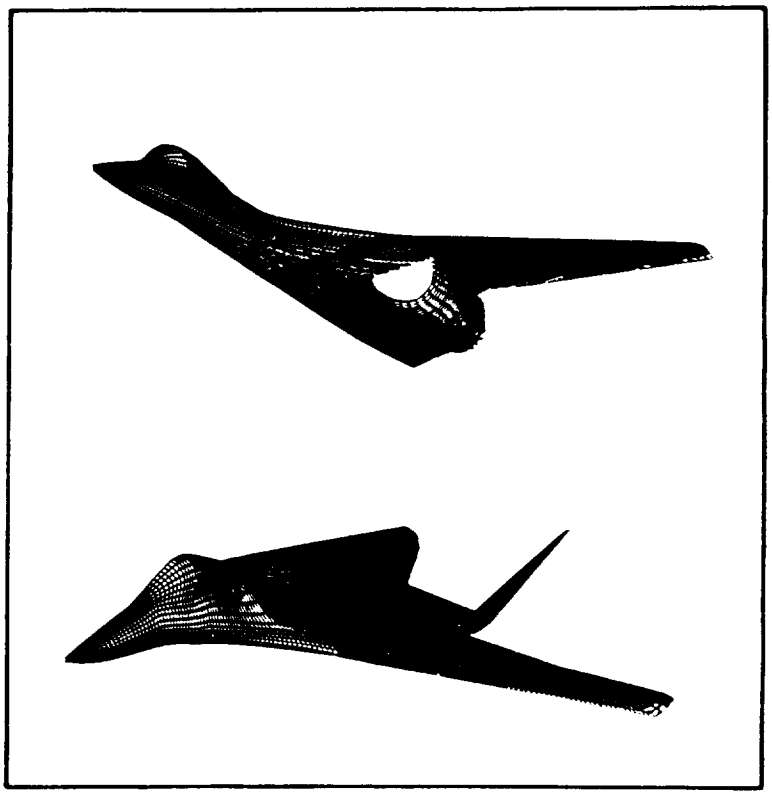


Fig. 2. Geometry and surface grid for a fighter with vertical tail and nacelle.

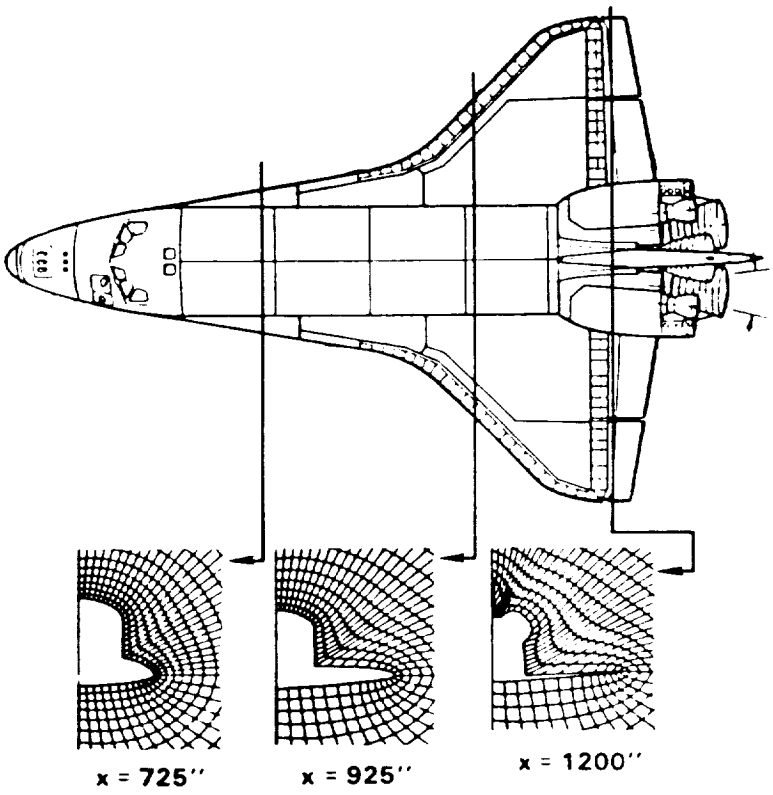


Fig. 3. Shuttle Orbiter configuration.

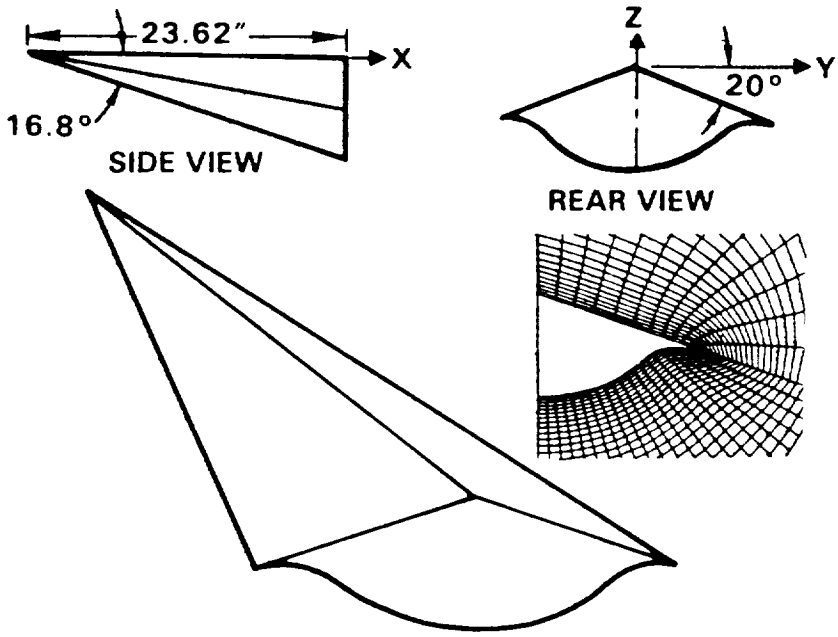


Fig. 4. Waverider geometry and grid.

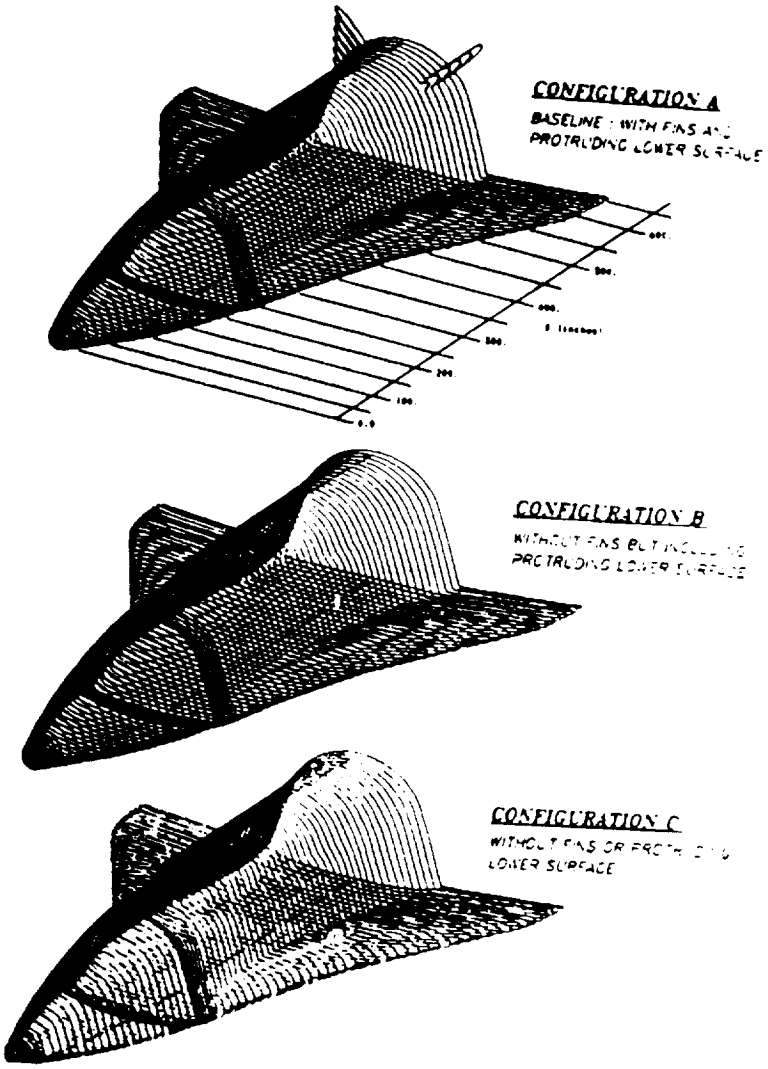


Fig. 5. Shuttle–like configuration.

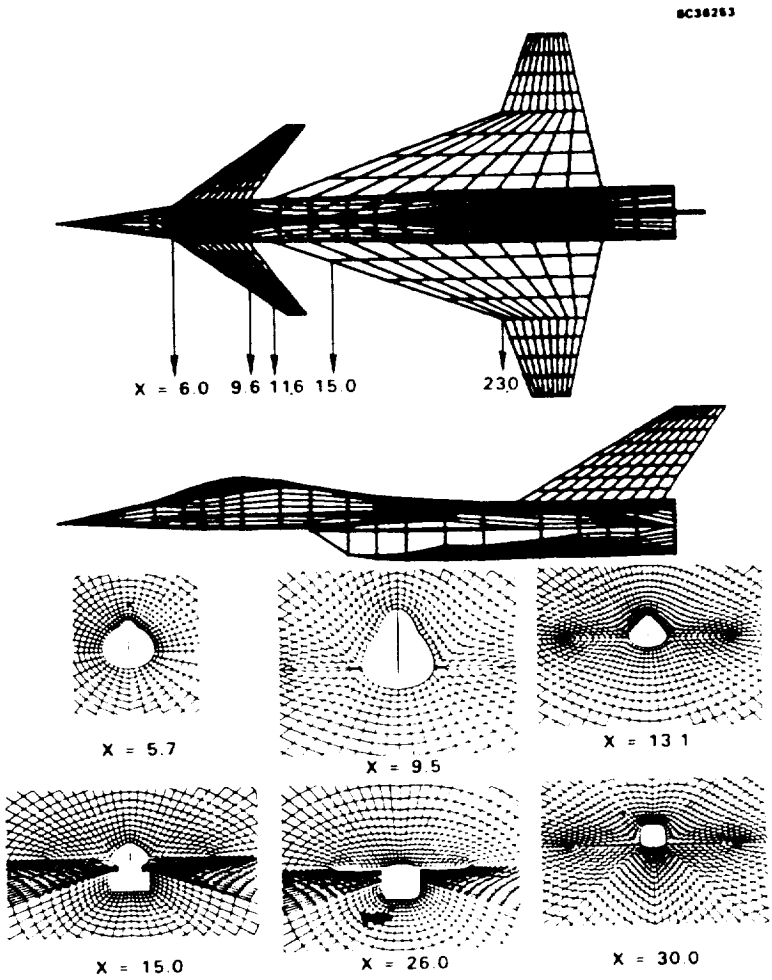


Fig. 6. Canard–wing fighter with nacelle.

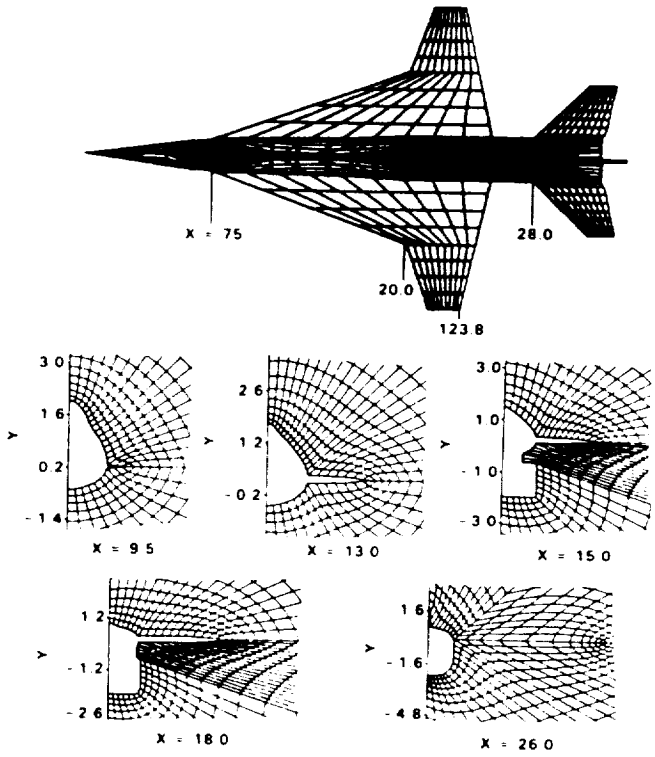


Fig. 7. Wing–horizontal tail fighter with nacelle.

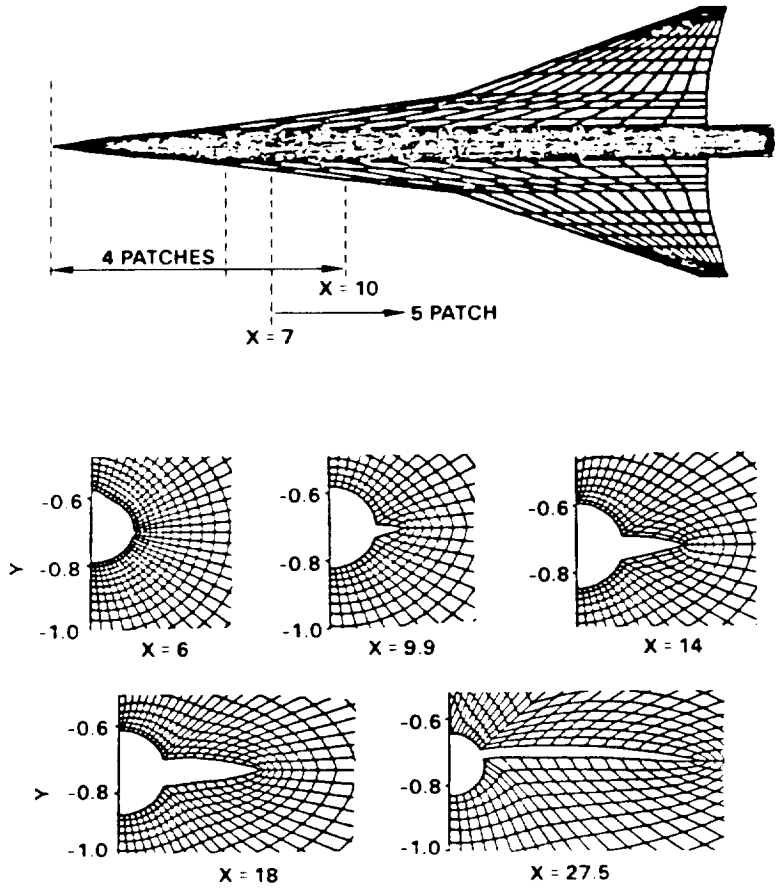


Fig. 8. Wing-body-strake configuration.

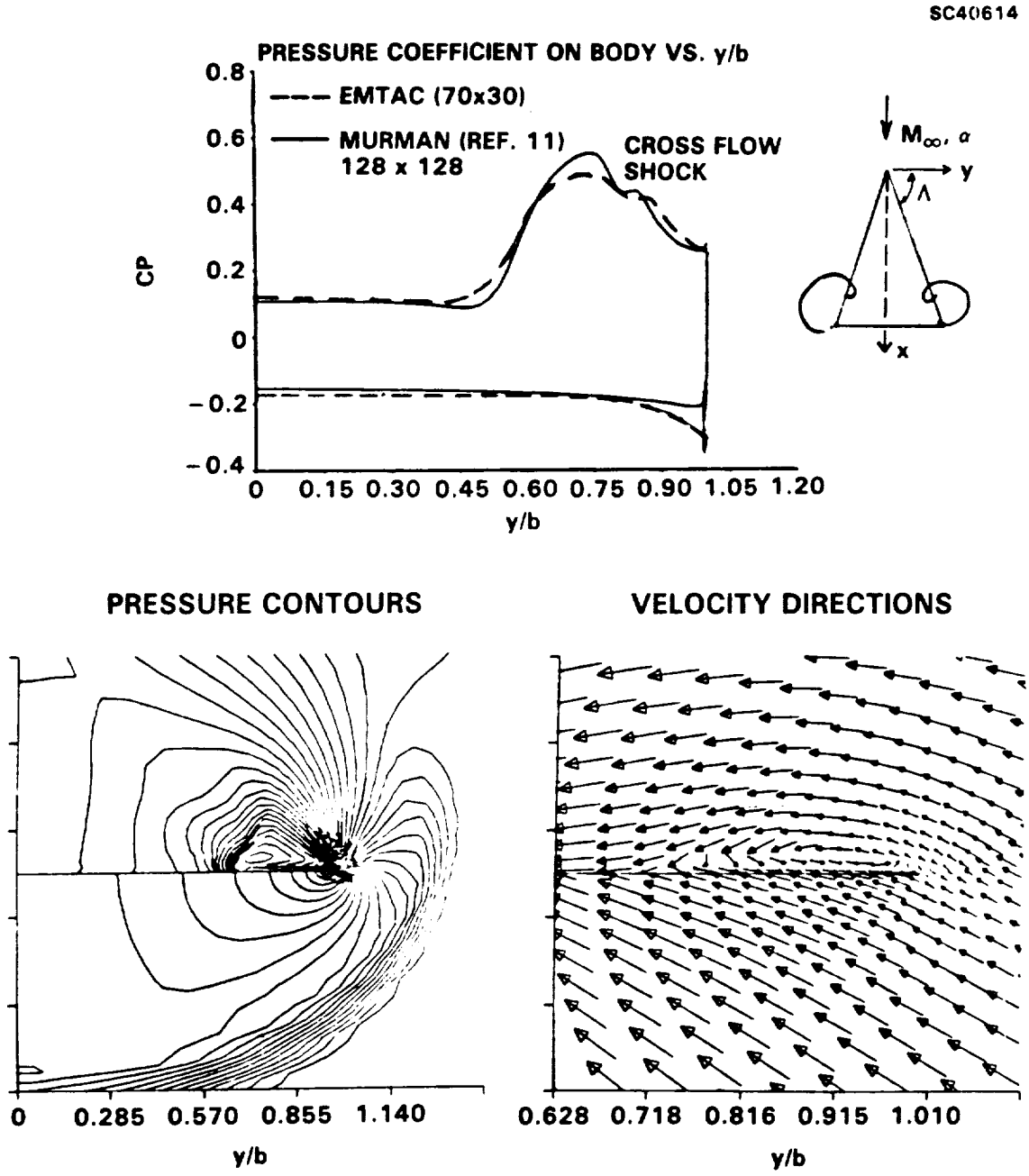


Fig. 9. Conical flat plate delta wing results. $M_\infty = 2$, $\alpha = 10°$, $\Lambda = 70°$.

4. CONCLUSIONS

The full potential SIMP code and the EMTAC Euler code have matured into powerful nonlinear tools for computing supersonic flows over complex aerospace configurations with canard, wing, tail, fuselage, and flow–through nacelle. The geometry setup and grid generation are common to both the codes. Several configurations have been computed using both the SIMP and the EMTAC codes over a wide range of Mach number and angle of attack. For cases with weaker shocks (satisfying the isentropic assumption) the codes agreed very well with each other. The real use of the EMTAC code is in computing high Mach number flows with strong shock, rotational and vortex effects.

The codes are operational on the CRAY–XMP and the VPS–32 supercomputers. The SIMP code runs 5 to 10 times faster than the EMTAC code.

5. REFERENCES

1. V. Shankar and S. Osher, "An Efficient Full Potential Implicit Method Based on Characteristics for Analysis of Supersonic Flows," AIAA Paper 82–0974, June 1982; AIAA J. 21 (9), 1262 (1983).

2. V. Shankar, K.–Y. Szema, and S. Osher, "A Conservative–Type–Dependent Full Potential Method for the Treatment of Supersonic Flows with Embedded Subsonic Regions," AIAA Paper 83–1887; AIAA J. 23 (1), 41–48 (1985).

3. K.–Y. Szema and V. Shankar, "Full Potential Solutions of Three–Dimensional Complex Geometries and Multibody Configuration," AIAA Paper 85–0272, Reno, Nevada, January 1985.

4. V. Shankar, K.–Y. Szema, and E. Bonner, "Full Potential Methods for Analysis/ Design of Complex Aerospace Configurations," NASA Contractor Report 3982, May 1986.

5. S.R. Chakravarthy and K.–Y. Szema, "An Euler Solver for Three–Dimensional Supersonic Flows with Subsonic Pockets," J. of Aircraft, Vol. 24, February 1987, pp. 73–83.

6. K.–Y. Szema, S.R. Chakravarthy, V. Shankar, and J. Byerly, "Comparison of Euler and Full Potential Marching Techniques for Flows over Complex Configurations," AIAA Paper 86–0244, Jan. 6–9, 1986.

7. S.R. Chakravarthy, "Development of Upwind Schemes for Euler Equations," NASA Contract Report 4043, January 1987.

8. K.–Y. Szema and V. Shankar, "Validation of a Full Potential Method for Combined Yaw and Angle of Attack," AIAA Paper 86–1834, June 9–11, 1986.

9. K.–Y. Szema, S.R. Chakravarthy, W.T. Riba, J. Byerly, and H.S. Dresser, "Multizone Euler Marching Techniques for Flows over Single and Multibody Configurations," AIAA Paper 87–0592, Jan. 12–15, 1987.

10. S.R. Chakravarthy and D.K. Ota, "Numerical Issues in Computing Inviscid Supersonic Flow over Conical Delta Wings," AIAA Paper 86–0440, Jan. 6–9, 1986.

11. E.M. Murman and A. Rizzi, "Applications of Euler Equations to Sharp Edge Delta Wings with Leading Edge Vortices," paper presented at the AGARD Symposium on

Applications of Computational Fluid Dynamics in Aeronautics, Aux–En–Provence, France, April 7–10, 1986.

12. O.A. Kandil, A.H. Chuang, and J.M. Shifflette, "Finite–Volume Euler and Navier–Stokes Solvers for Three–Dimensional and Conical Vortex Flows over Delta Wings," AIAA Paper 87–0041, Jan. 12–15, 1987.

13. Raymer, D.P., "Configuration Development System," Rockwell International Report TFD–78–755–4, April 1983.

APPENDIX A — CODE STRUCTURE

CODE ORGANIZATION

The EMTAC analysis code is applicable to arbitrary wing–body–nacelle–tail arrangements from moderate supersonic Mach numbers ($M_\infty \sim 1.2$) to values of the hypersonic range ($M_\infty \sim 40$). The lower code limit is governed by the extent of the embedded subsonic flow while the upper limit results from a breakdown in the perfect gas assumption for the flow.

The program is written in FORTRAN V language. It can be executed on supercomputers such as the CRAY–XMP and CYBER 205, as well as on superminicomputers such as the VAX and ELXSI. The program consists of a main routine (UDRIVE) and several subroutines. A brief description of the code along with input instructions needed to execute the code are given in this Appendix.

Program UDRIVE

Program UDRIVE coordinates the entire operation. A flowchart and subroutines describing the various operations performed by the UDRIVE program are given in Fig. A1. The UDRIVE program sets up the initial (known) data plane and the body–fitted grid system and performs the marching procedure to advance the solution. The various read and write tapes used in the calculation are listed below.

TAPE1 — Disk data input file containing starting solution to be read in for restart

TAPE2 — Disk data output file containing final solution to be stored in current run for later use

TAPE5 — Disk data input file containing input data needed (including the geometry data)

TAPE7 — Disk data output file to output solution in the form needed by plotting program and postprocessing

SC40616

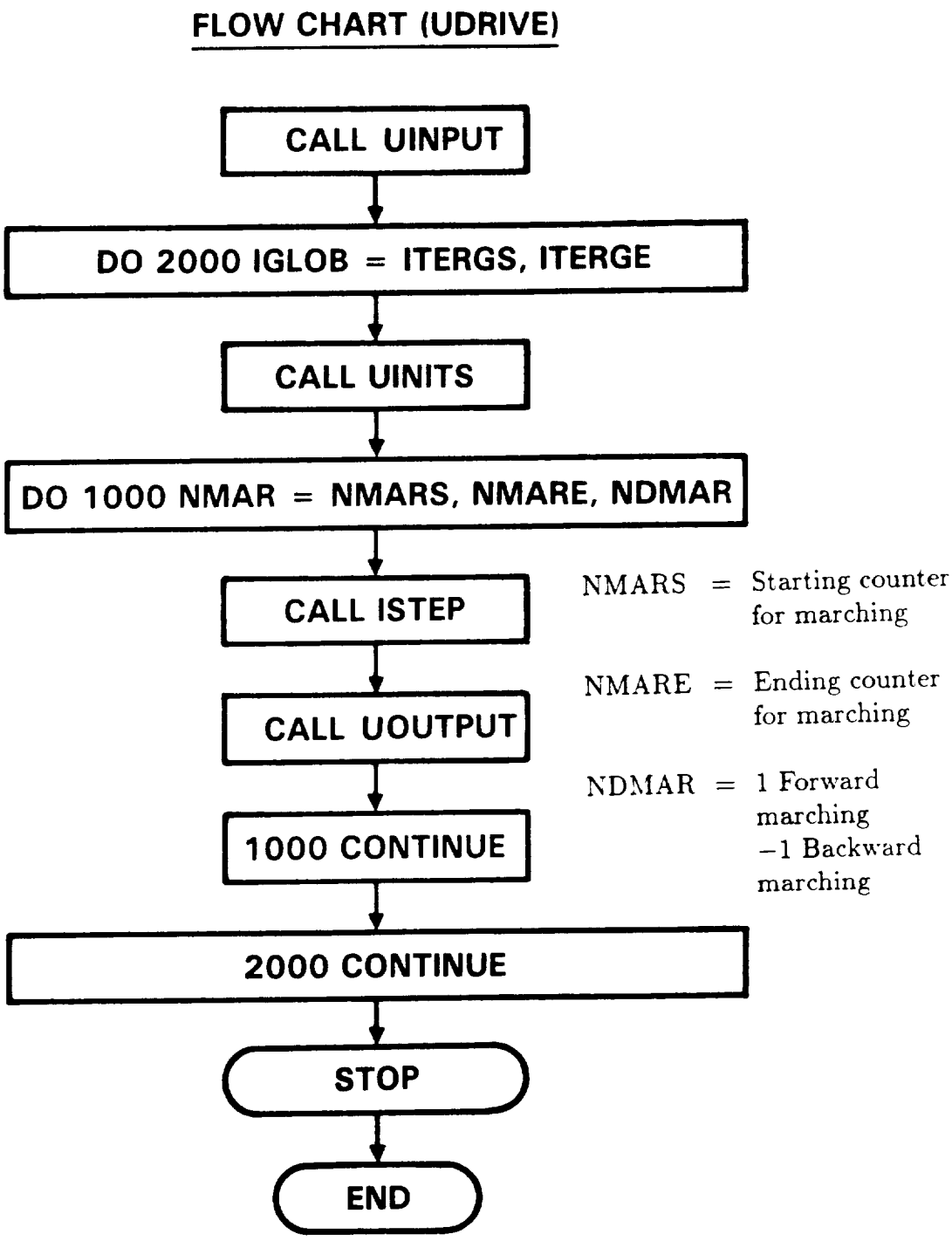


Fig. A1. Flow chart for EMTAC code.

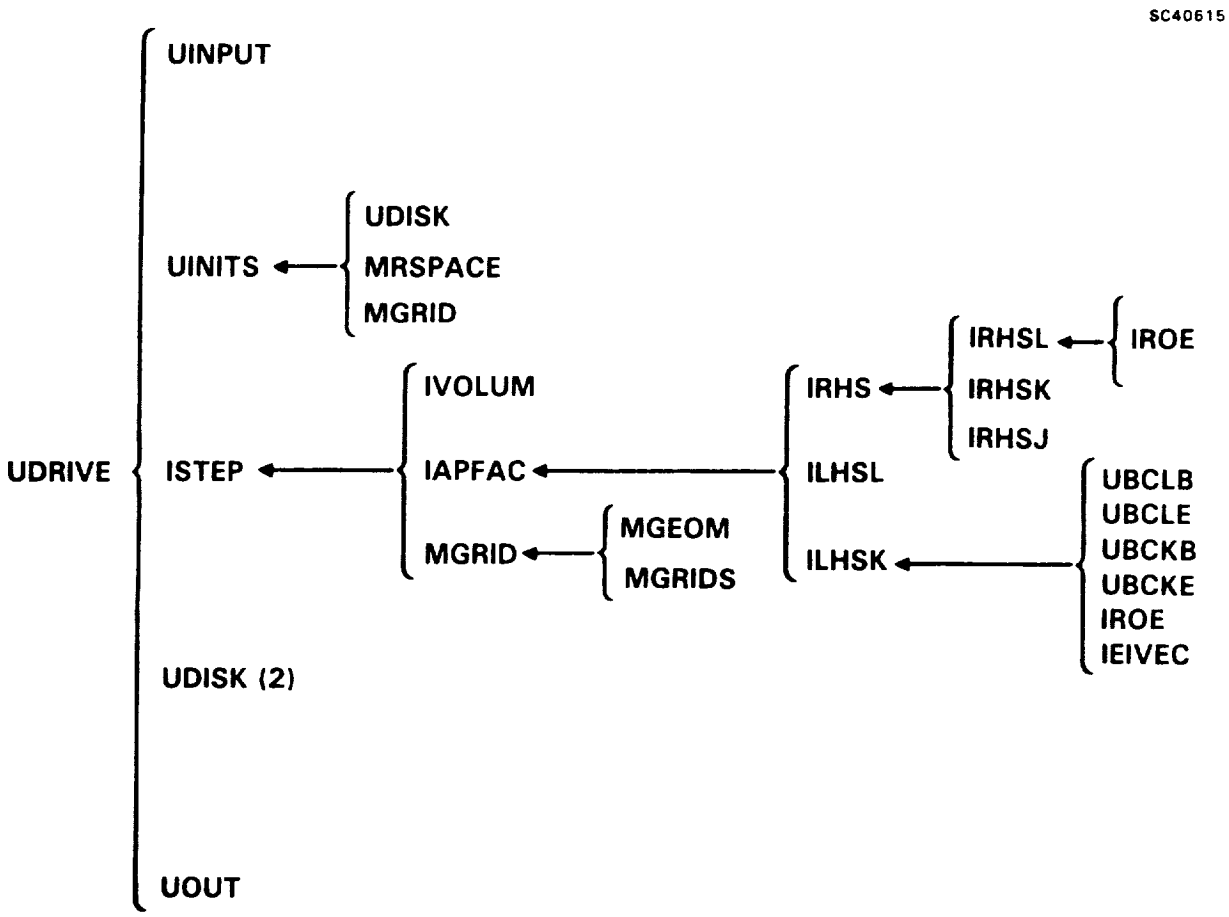


Fig. A1. Flow chart for EMTAC code (concluded).

TAPE8	—	
TAPE9	—	Data needed for subsonic global iteration
TAPE10	—	

Subroutine ISTEP

Subroutine ISTEP performs the marching procedure and updates the solution one step at a time.

Subroutine IAPFAC

The factored implicit scheme for the governing Euler equations can be written as

$$\left[I + \frac{1}{V}\widehat{A}^{-1}\left\{B^{+}_{k-1/2}\Delta_{k-1/2} + B^{-}_{k+1/2}\Delta_{k+1/2}\right\}\right]$$
$$\left[I + \frac{1}{V}\widehat{A}^{-1}\left\{C^{+}_{l-1/2}\Delta_{l-1/2} + C^{-}_{l+1/2}\Delta_{l+1/2}\right\}\right]\Delta^{s}q$$
$$= \frac{1}{V}\widehat{A}^{-1}\ [\text{Right Hand Side}]$$

The subroutine IAPFAC calls Subroutines ILHSL (left hand side L–direction), ILHSK (left hand side K–direction) and IRHS (right hand side) to calculate the solution by using the approximate factorization method.

Subroutine IROE

The numerical flux at cell surface $m + 1/2$ is given as

$$h_{m+1/2} = \frac{1}{2}\left[f\left(Q_{m+1}, N_{m+1/2}\right) + f\left(Q_{m}, N_{m+1/2}\right)\right]$$
$$- \frac{1}{2}\left[\sum_{i}\left(\lambda^{i+}_{m+1/2} - \lambda^{i-}_{m+1/2}\right)\alpha^{i}_{2}r^{i}_{m+1/2}\right]$$
$$= f\left(Q_{m}, N_{m+1/2}\right) + \sum_{i}\lambda^{i-}_{m+1/2}\alpha^{i}_{2}r^{i}_{m+1/2}$$
$$= f\left(Q_{m+1}, N_{m+1/2}\right) - \sum_{i}\lambda^{i+}_{m+1/2}\alpha^{i}_{2}r^{i}_{m+1/2}$$

where $\alpha_i = \ell_i dQ$.

The right eigenvector (r), left eigenvector (ℓ), and parameter α are calculated in this subroutine.

Subroutines UBCLB, UBCLE, UBCKB, UBCKE

UBCLB: Apply boundary conditions at $L = 1$.
UBCLE: Apply boundary conditions at $L = LGRD$ (end of L).
UBCKB: Apply boundary conditions at $K = 1$.
UBCKE: Apply boundary conditions at $K = KGRD$ (end of K).

Subroutine MGEOM (N9, NRP)

$N9 = 0$, geometry data at X_1 and X_2 are read in
> 0, geometry data at X_1 is updated and X_2 is read in
$NRP = 0$, constant x marching plane geometry calculation
$= 1$, spherical marching plane geometry calculation

Subroutine MGEOM sets up the body grid points from a prescribed geometry shape. From the input geometry points, a key point system is established using cubic splines. These key points are then joined from one prescribed geometry station to the next to provide the geometry at any intermediate marching plane[12].

Subroutine MGRID

Once the body points are obtained at a marching plane from MGEOM, subroutine MGRID sets up the entire crossflow plane grid using an elliptic grid solver that satisfies certain grid constraints.

Subroutine NFORCE (PX, PY, PM, AREA, KFG)

At the end of each marching plane calculation, this subroutine computes the axial force, PX, vertical force, PY, and the side force, PZ, by integrating the pressure force acting on an elemental area, dA.

$KFG = 0$, conical or blunt body nose force calculation
$= 1$, rest of the body force calculation.

The program also prints the force coefficients, C_L and C_D, information based on a prescribed reference area, and moment coefficients, C_M, about a given reference point $(X_0,\ Y_0)$.

Header Data

A typical analysis of a complete configuration requires several regions of marching calculations for a complete analysis. Each region calculation has a different set of header instructions for describing grid parameters, wake information if pertinent, restart directions, and number of mesh points for each patch of the region. A sample input is given in Fig. A2, and a brief description of each variable is given in this section.

Symbol	Format	Description
NMARCH	I5	Number of axial marching steps. If NMARCH = 0, and XSTART = $\bar{\zeta}$ and DISKIN = F the code generates geometry and grid data at $x = \bar{\zeta}$ for plotting. For NMARCH $\neq$ 0, the code will march for NMARCH steps unless XEND is encountered first. NMARCH must include NCON iterations if applicable. (NMARCH = 0 option for grid plot is provided to allow the user to review the quality of grid at various axial stations before the flow solver is turned on.)
KMAX	I5	Mesh points in the normal direction (η). Present maximum is 30. This can be increased by increasing the dimension.
LMAX	I5	Mesh points in circumferential direction (ξ) (maximum value: 80). If this number is incorrectly specified, the code will reset LMAX properly using the LMAX = 1 + (IPT1–1) + (IPT2–1) + (IPT3–1) + $\cdots$ + (IPTn–1) + 1 (definition of IPT follows in the next section). "n" is the number of patches.

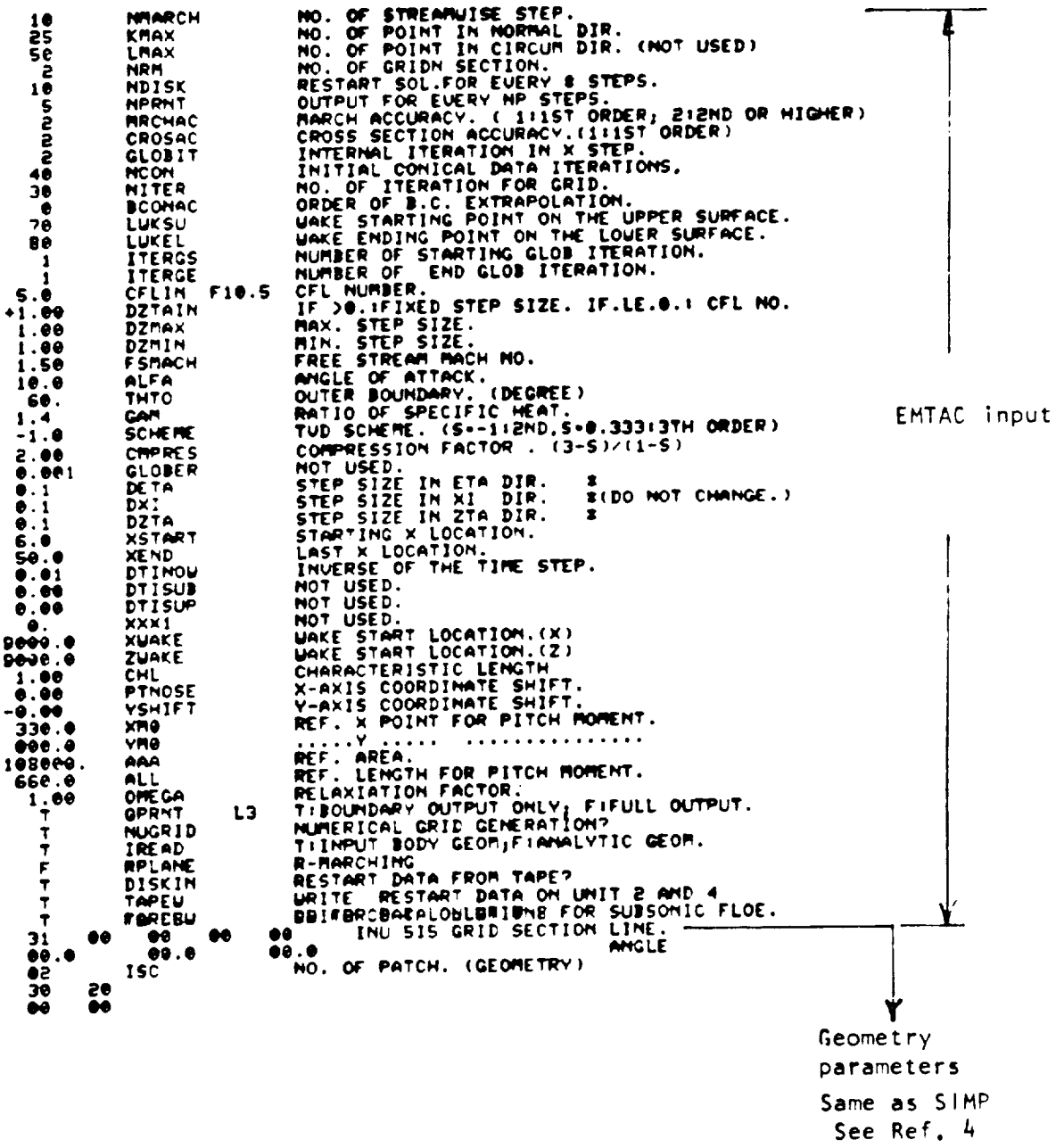


Fig. A2. Sample input data for the EMTAC code.

NRM	I5	Number of grid regions (separated by dashed lines in Fig. A3). Maximum of 6 allowed.
NDISK	I5	Write restart data for every NDISK marching step.
NPRNT	I5	Print out solution for every NPRNT step.
MRCHAC	I5	Accuracy parameter in marching direction. 1: first order accuracy 2: second or higher order accuracy (Also see SCHEME) Recommended value: 1
CROSAC	I5	Accuracy parameter in L and K direction 1: first order accuracy 2: second or higher order accuracy (Also see SCHEME) Recommended value: 2
GLOBIT	I5	Number of internal iterations to perform before proceeding to next marching step. Recommended value: 2

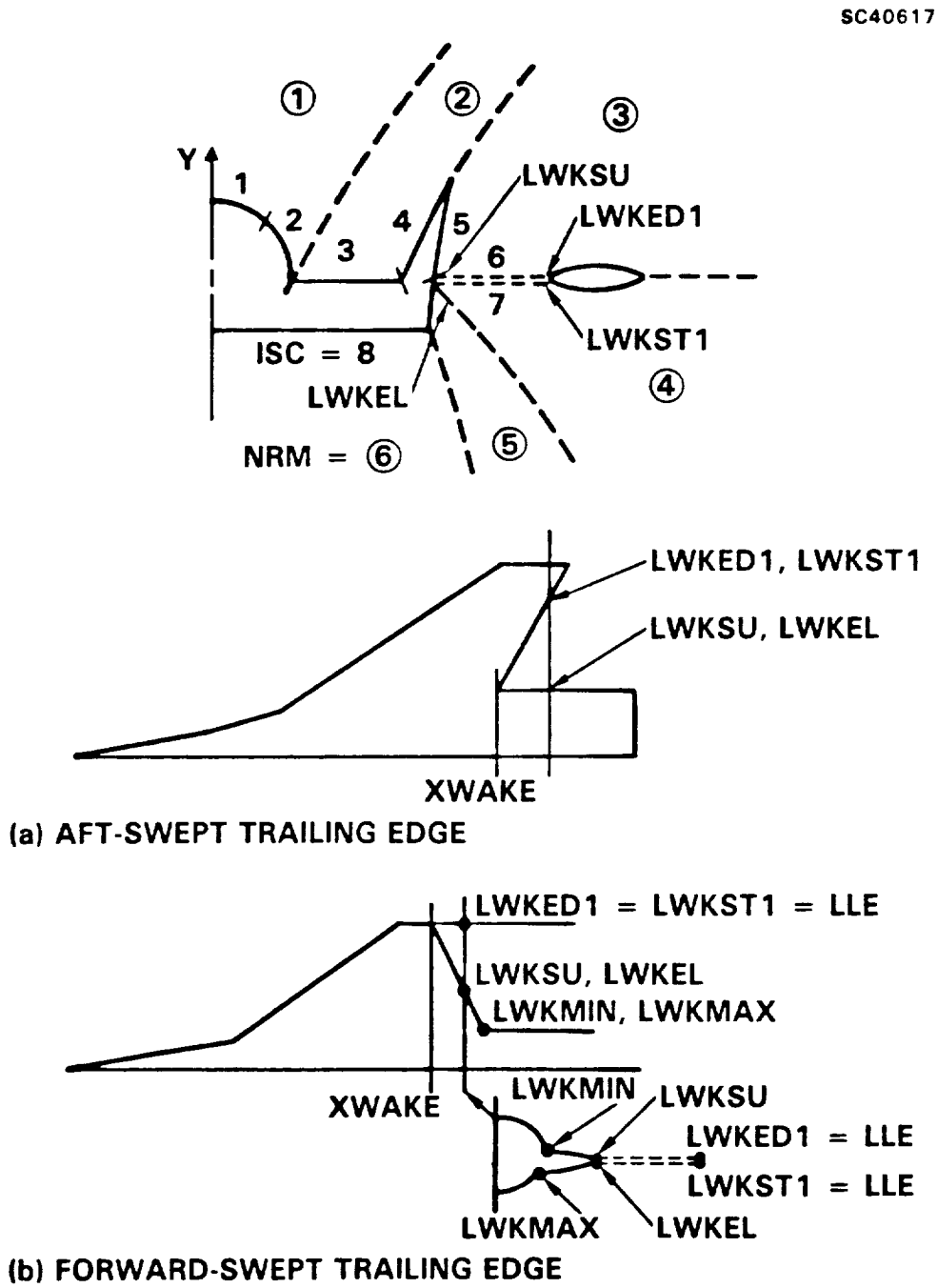
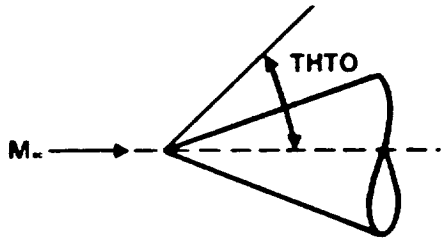


Fig. A3. Cross section patches and nomenclature.

NCON	I5	Number of iterations for conical starting solution (usually set to 30). To establish this starting solution, the geometry is initially assumed to be conical. The geometry at XSTART is projected forward conically to a point at (0,0,0). (The nose of the configuration is assumed to be at (0,0,0). If the geometry is not input this way, shift the geometry using PTNOSE and YSHIFT.) The solution is then obtained for this conical geometry based on NCON iterations. The conical solution is used as a starting solution for the nonconical case, beginning at XSTART. The user should be aware that NCON is included in the NMARCH total. Also, XSTART output values have no physical significance during conical calculation.
NITER	I5	Number of iterations to generate the marching grid using an elliptic grid solver. Usually set to 30. If grid routine fails, set this to 0 to analyze the geometry and the initial grid generated before grid relaxation (this is for debugging purposes). Set NITER back to 30 for flow field analysis. NMARCH should be set to zero for analyzing the grid quality.
BCONAC	I5	On the B.C. surface, the solution is extrapolated. 0: set surface value equal to the first cell centroid values. 1: the value is obtained by using 2 points extrapolation

		2: the value is obtained by using 3 points extrapolation. Recommended value: 1 or 2
LWKSU	I5	L value of starting point of a patch containing wake (Fig. A3).
LWKEL	I5	L value of ending point of a patch containing wake (Fig. A3).
ITERGS	I5	Starting number of global iterations for subsonic region calculations. Set to 1 for supersonic marching case.
ITERGE	I5	Ending number of global iterations for subsonic region calculations. Set to 1 for supersonic marching case. The number of global iterations = ITERGE=ITERGS.
CFLIN	F10.5	Not used.
DZTAIN	F10.5	Initial step size. For nonconical geometry calculations, DZTAIN is chosen to be either DZMIN or DZMAX. If DZTAIN is set to less than DZMAX, then during marching calculation, $\Delta\zeta$ will be slowly increased to DZMAX.
DZMAX	F10.5	Maximum step size.
DZMIN	F10.5	Minimum step size.
		(DZMAX and DZMIN depend on the complexity of the geometry. Suggested value: DZMAX = total length/400 and DZMIN = DZMAX/2.) If DZMIN is set equal to DZMAX, then constant step size is used.

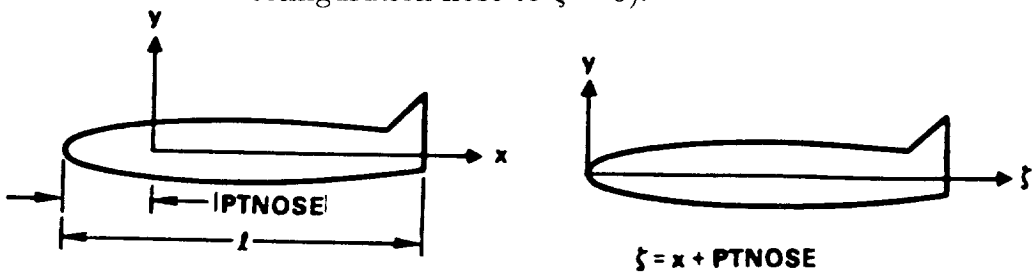
FSMACH	F10.5	Freestream Mach number.
ALFA	F10.5	Angle of attack (degrees).
THTO	F10.5	Angle of outer boundary (degrees). This angle must be larger than the bow shock wave in order for the code to capture the bow shock. Often the best way to choose this value is to calculate the bow shock wave half angle and add 10°.



GAM	F10.5	Ratio of specific heat.
SCHEME	F10.5	Parameter to pick particular TVD scheme. $\frac{1}{3}$: third order accurate scheme −1: fully second order upwind scheme 0: Fromm's second order scheme $\frac{1}{2}$: low truncation error second order scheme Recommended value: −1
CMPRES	F10.5	Compression factor. Choose in the range $1 < \text{CMPRES} < \frac{3-\text{SCHEME}}{1-\text{SCHEME}}$ Normally pick (3-SCHEME)/(1-SCHEME)
GLOBER	F10.5	Not used.
DETA DXI DZTA	F10.5	Set to 0.1 (do not change).

XSTART	F10.5	Starting X location. If DISKIN = TRUE, this value is overwritten by stored restart value.
XEND	F10.5	Final X location for this run.
DTINOW	F10.5	Inverse of the time step. Set to 0.01 for supersonic flow. For the subsonic flow region, set to $\sim$ 10.0 and gradually decreasing to 0.01. The user provides the necessary update changes in Subroutine UDRIVE or through input variable ITERGS and ITERGE for this variation. Usually the variation from 10 to 0.01 can be imposed in ten time–relaxation sweeps.
DTISUB	F10.5	Not used.
DTISUP	F10.5	Not used.
XXX1	F10.5	Not used.
XWAKE	F10.5	Wake starting location in the axial direction (see Fig. A3).
ZWAKE	F10.5	Not used.
CHL	F10.5	Geometry scale factor. If set to total length, X will be scaled from 0 to 1. If set to 1, actual dimensions of the geometry are used. Use of dimensional (CHL = 1) or nondimensional (CHL = ℓ) option is left to user's choice.

PTNOSE	F10.5	Axial geometry shift. Equal to negative of apex of the forebody (*i.e.*, shifts configuration nose to $\zeta = 0$).



YSHIFT	F10.5	Vertical geometry shift (*i.e.*, shifts configuration nose to $\eta = 0$).
XMO	F10.5	Moment reference X location (unit $\sim$ length).
YMO	F10.5	Moment reference Y location (unit $\sim$ length).
AAA	F10.5	Reference area to compute aerodynamic force coefficients (unit $\sim$ length^2).
ALL	F10.5	Reference length to compute aerodynamic moment coefficients (unit $\sim$ length). XO, YO, AAA, and ALL are to be chosen (dimensional or nondimensional) based on CHL.
OMEGA	F10.5	Overrelaxation parameter for grid generation. Suggested value: 1.0 (for vectorized code) 1.75 (for scalar code).
OPRNT	L5	T: boundary output only F: full output
NUGRID	L5	T: Numerical grid generation (normally used). F: User must adapt code for his particular need.

IREAD	L5	T: Read body geometry input which must be supplied in the format described in the next section titled "Geometry Data". F: Analytic geometry (which must be supplied by the user and inserted in subroutine GRID).
RPLANE	L5	Not used.
DISKIN	L5	T: Restart the calculation. F: Start the calculation from freestream.
TAPEW	L5	T: Write restart data on Tape 2. F: No data storage for restart.
TAPE8W	L5	T: Write entire flow field data for subsonic iterations on Tape 8. F: No flow field data saved.
FORCE	L5	T: Compute aerodynamic forces and moments. F: No force computation.
THTU(5)	5I5	Grid region terminal points (k) (see Fig. A3). These values are the K values of the points where the dashed lines intersect the body.
INU(5)	5F10.4	Polar angle (degrees) at respective terminal point.
ISC	I5	Number of patches (geometry) that define the cross-sectional shape of the configuration for this region of the configuration (see Fig. A4). (Maximum number of patches = 15.)

NPT(15)	15I5	Number of output points on each patch (maximum number of points per patch is 30).

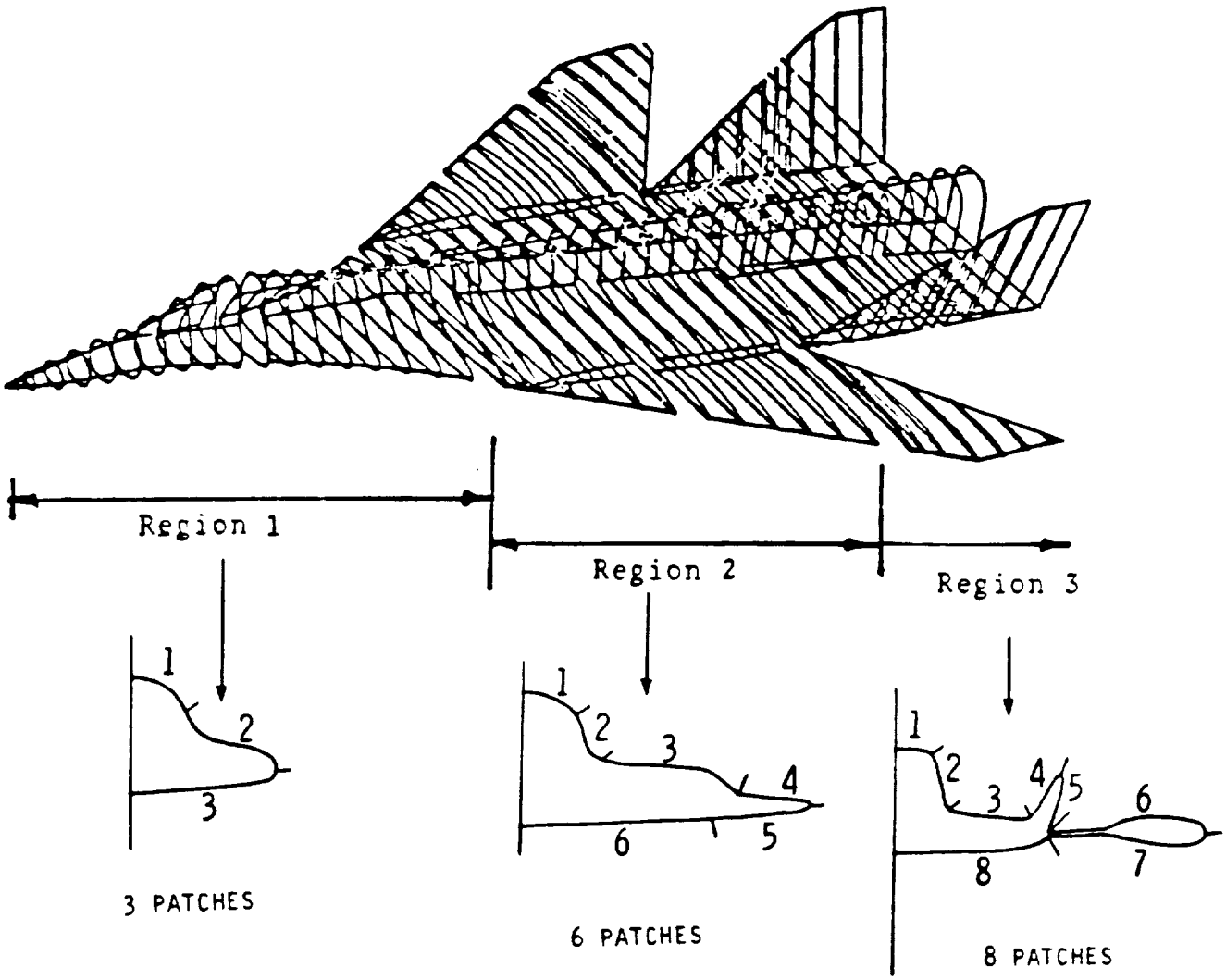


Fig. A4. Sample problem.

Geometry Data

The cross–sectional geometry of a typical aircraft changes considerably in the axial direction due to emergence of various components such as canopy, wing, nacelle, and tail, etc. The marching computation, as it sweeps along the marching direction ξ, has to account for this geometry variation to set up the proper body–fitted coordinate system to aid in the application of body boundary conditions. To treat complex geometry cross sections, patches are introduced to define the geometry as indicated in Fig. A4. Using patches, a configuration is defined by several regions of cross sections. The number of patches defining a section is constant for a given region (Fig. A4).

A complete computation over a configuration such as the one in Fig. A4 is usually done in segments rather than in one shot. The calculation starts from the nose and proceeds along ξ. Even within a region (defined by the same number of patches), the calculation might be done in segments using the restart option in the code. Restart is used any time the calculation is halted and then continued with another run that picks up where the previous run left off. Pure restart is performed only when there is no alteration to the number of points along η and along ξ, and no change in the number of grid points per patch between the previous run and the current restart run. If there is any alteration to the grid structure, the restart run will automatically perform a respace operation to interpolate the solution from the previous solution grid to the current grid. Respace is used whenever the following situations are encountered:

1) Number of patches defining the cross section is changed. This situation occurs when the cross–sectional geometry becomes more complex. This is illustrated in Fig. A4.

2) Number of KGRD (KGRD = KMAX−1) and/or LGRD (LGRD = LMAX−1) points is changed (even if the number of patches defining the cross section is kept the same as before). This situation often occurs for cases where a patch length is increasing with ξ. For example, a swept wing is very small when it first appears in the cross section of the geometry and only requires a few grid points for accurate computation of the flow field. However, as the analysis is continued in the ξ direction, the wing patches grow and will require more points for accurate flow field analysis.

3) Number of grid points per patch is changed (even if KGRD is kept the same as before).

Any time a respace is required, the code must be stopped. The code will automatically do a respace if KGRD or LGRD is different from the previous values of KGRD and LGRD.

One may be able to compute the entire configuration using the same number of patches and same KGRD and LGRD values throughout to avoid the respace requirement. This will

mean even in the forebody region of a configuration, where the cross–sectional geometry is usually simple, more grid points and more patches are to be used than necessary to adequately resolve the flow field. Use of the same number of patches and grid points for throughout the length of the configuration is generally not recommended. This can substantially increase the total execution time.

Transitioning from one region to the next (number of patches is changed) requires an overlapped zone, as illustrated in Fig. A5, to allow for increased or decreased number of patches in the next region. The extent of this overlapped zone must be sufficient to include at least the final three marching data planes of the prior region. In the overlapped region, the data from the previous region is interpolated onto the grids of the new region. For the example of Fig. A5, the results from the 3–patch region are interpolated onto a 4–patch region grid at the same x location. This is required in order to continue marching along the body with the new patch definition.

Figure A5 illustrates how to transition from a fuselage computation to a wing–fuselage computation. First, the calculation is performed for the fuselage section denoted by REGION1 which ends just prior to the starting point of the wing. This calculation might involve, say, three patches. Then, to introduce the wing, a four patch representation is used in REGION2. In the overlapped zone, the fuselage which is defined using a three patch representation in REGION1 is represented by a four patch representation as part of REGION2. The second and third patch locations on the fuselage in REGION2 within the overlapped zone are chosen in the vicinity of where the leading edge of the wing is expected to emerge from the fuselage.

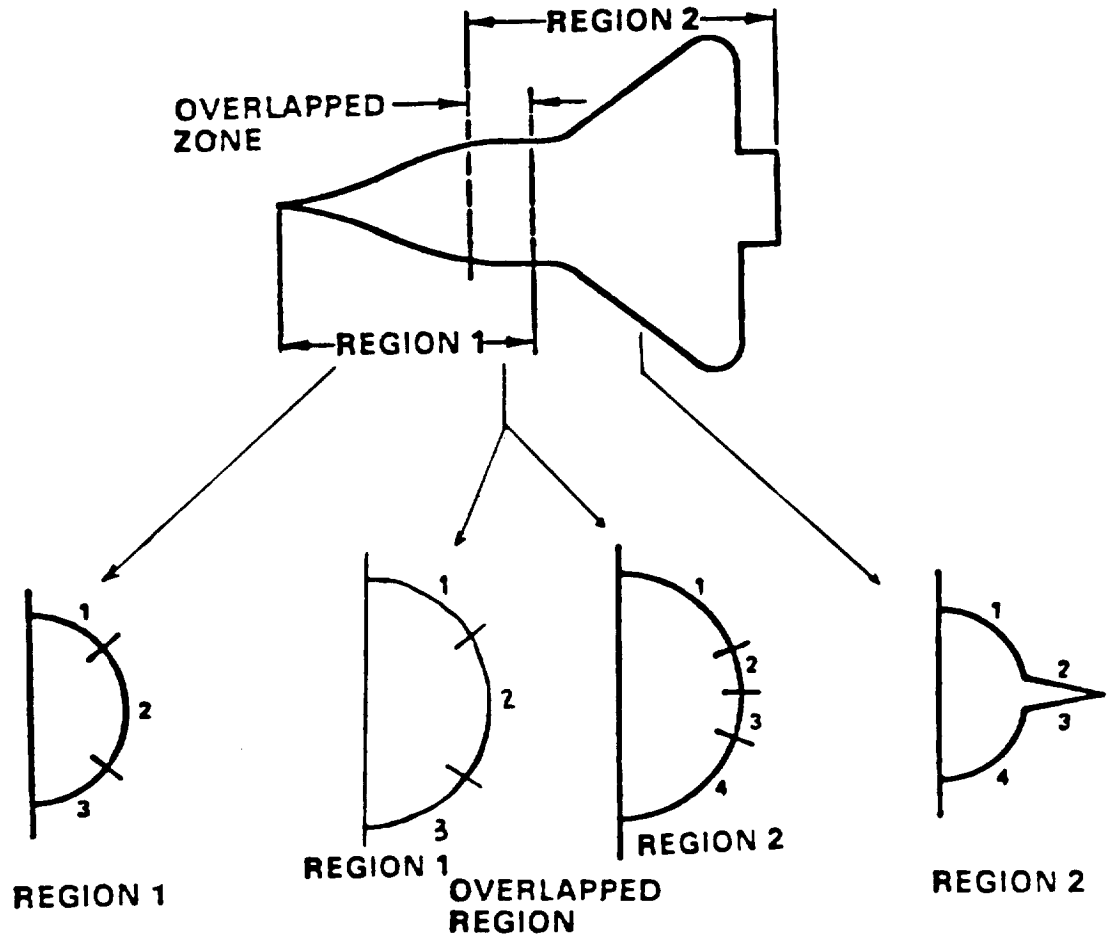


Fig. A5. Cross section patches in overlap region.

Wake Geometry

Behind the trailing edge of a lifting surface, a wake cut is introduced (see Fig. A3). The treatment of wake cut within the code requires the knowledge of starting and ending L index values of the upper wake cut and the lower one. Depending on the sweep of the trailing edge, the wake cut is appropriately modeled. This is illustrated in Fig. A3. The user has to define the shape of the trailing edge and also the starting x value in Subroutine MGRID where the wake begins to appear in the cross-sectional geometry (XWAKE). The wake cut is part of a patch which contains the wing also as illustrated in Fig. A3. As marching proceeds along the axial direction, the extent of the wake cut grows within that patch. The nomenclature for the starting and ending points of the wake cut are also indicated in Fig. A3. The number of points in the patch containing the wake cut is not allowed to change during the calculation. Thus, while exercising the respace option in the region containing the wake, the user has to ensure that the number of points in the wake patch (usually there are two wake patches; one corresponding to the upper cut and one for the lower cut) is not altered.

The shape of the trailing edge is provided by the user using the update option.

For the wing–body–vertical case of Fig. A4, a 3–patch initial region, a 6–patch center region, and an 8–patch final region was used. Zero length patches are <u>not</u> permissible. Since the analysis is marching in nature, a complete geometry data set is not required to begin and partially process a problem. Appropriate use of restart solutions allows continuation of the analysis as new or modified geometry becomes available.

The format for a typical station is shown below. The group of cards is repeated for each station of a region. The last point of each patch (except for the last patch of a station) should have the same coordinates as the first point of the next patch.

Card No.	Format	Field	Name	Description
A1	F15.6,I5	1	X1	The x value (longitudinal) of this station.
		2	ISC1	The number of patches for this section. $1 \leq$ ISC1 ≤ 15.

The group of cards A2 through A3 are repeated ISC1 times.

A2	3I5	1	ITH	Patch number ≤ 15.

		2	IPT	Number of points in this patch. $2 \leq \text{IPT} \leq 30$.
		3	ND	Mesh spacing parameters*. Typically the same for all stations of a region.

The A3 card is repeated IPT times.

A3	2F15.6	1	YK	Vertical location of point (positive upwards). Points start at top centerline.
		2	ZK	Spanwise location of point.

Cubic spline interpolation is performed on input patch data to derive the geometry. Linear interpolation is performed to define the geometry at a marching plane between input stations.

Sample geometry data for the problem of Fig. A4 is presented in Table 1 and was developed using CDS[13].

*For Segment AB: 0 equal space; 1 cluster near A; 2 cluster near B.

Table 1. Geometry Data

ORIGINAL PAGE IS
OF POOR QUALITY

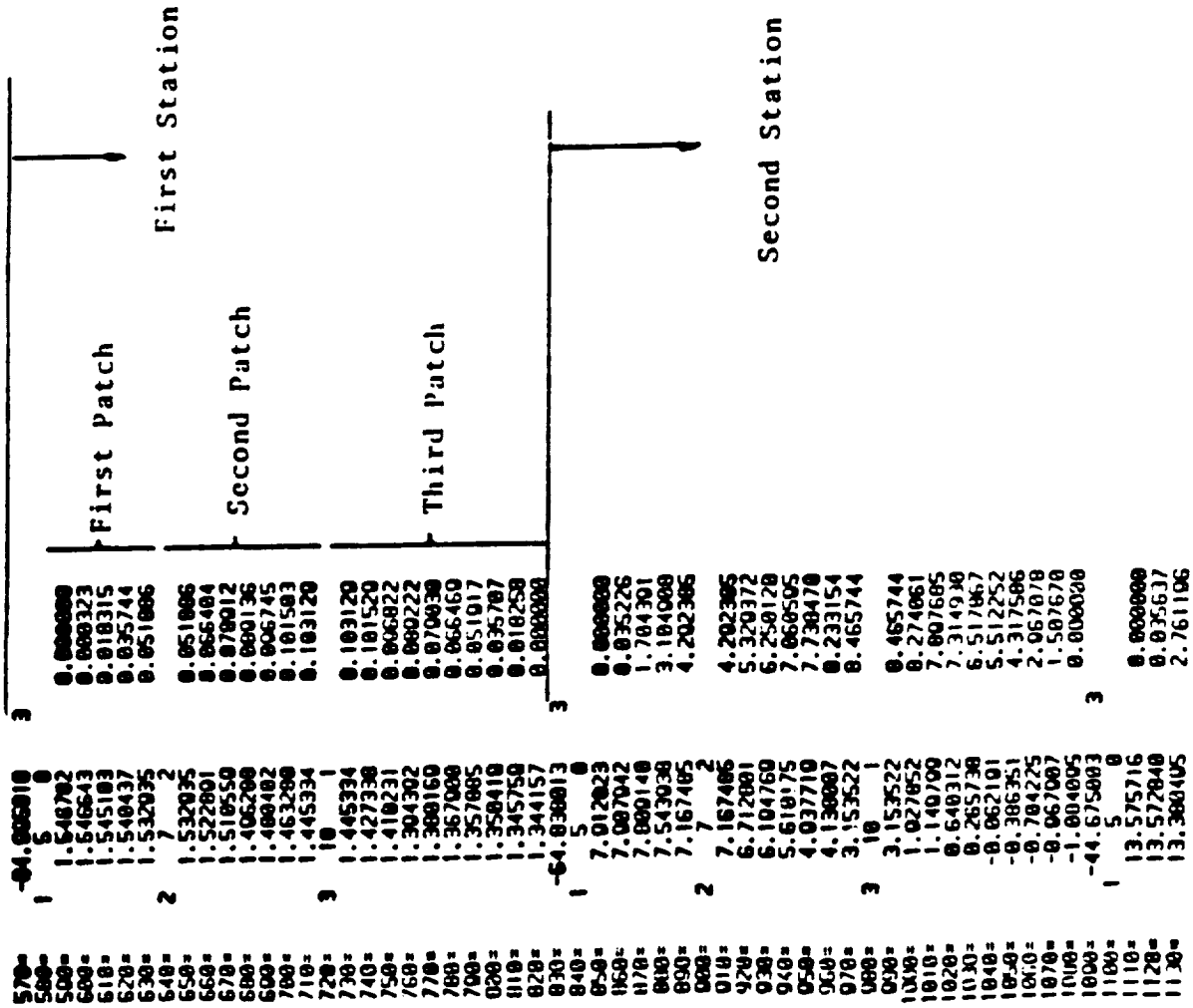


Table 1. Geometry Data (continued)


```
0300:     280.670050    3          <-- End of Region 1
0310=  1   13    0
0320=       40.958420        0.000000
0330=       40.958382        1.037543
0340=       40.958420        4.388307
0350=       40.680511        7.428100
0360=       40.879784       10.266657
0370=       47.604164       12.876740
0380=       45.809048       15.224077
0390=       43.811752       17.267170
0400=       41.382515       18.957400
0410=       38.653473       20.230323
0420=       35.664105       21.040073
0430=       32.452553       21.310206
0440=       30.713201       21.205403
0450=  2   30    2
0460=       30.713201       21.205403
0470=       29.147600       23.800067
0480=       26.650154       28.400600
0490=       24.242340       33.620456
0500=       22.121180       38.922207
0510=       20.874866       42.500350
0520=       19.201040       47.515030
0530=       17.635311       53.405407
0540=       16.076260       61.000318
0550=       15.763630       65.007015
0560=       16.355867       68.560077
0570=       14.472210       73.730820
0580=       13.400101       78.001434
0590=       12.402630       84.404002
0600=       11.402655       90.847702
0610=       10.037710       91.700420
0620=       10.246025       94.368530
0630=        9.555044       97.232040
0640=        8.031576       99.526506
0650=        8.207584      102.105400
0660=        7.425758      104.684341
0670=        6.573520      107.263100
0680=        5.626501      109.842041
0690=        4.722771      112.048233
0700=        4.134672      113.304123
0710=        3.615775      114.227844
0720=        3.348063      114.597412
0730=        3.147207      114.798508
0740=        2.910736      114.946838
0750=        2.690031      114.993576
0760=  3   29    1
0770=        2.690031      114.993756
0780=        2.570846      114.980423
0790=        2.459620      114.933670
0800=        2.363230      114.838547
0810=        2.245280      114.634401
0820=        2.062772      113.952515
0830=        1.905408      112.702000
0840=        1.762813      111.130202
0850=        1.631411      109.000453
0860=        1.515947      106.502197
0870=        1.424664      103.704922
0880=        1.356829      101.146477
0890=        1.284087       96.947678
0900=        1.005898       94.368591
0910=        0.917631       90.622636
0920=        0.640761       86.630768
0930=        0.257428       81.430713
0940=       -0.130081       76.311234
0950=       -0.450283       71.150040
0960=       -0.645550       65.907915
0970=       -0.904200       61.978348
0980=       -1.161518       57.469460
0990=       -1.284488       52.036806
1000=       -1.417318       45.258186
1010=       -1.456881       36.650615
1020=       -1.600638       27.401102
1030=       -1.781206       18.323032
1040=       -1.914505        9.163094
1050=       -1.960518        0.000000
1060=      257.762939    6          <-- Beginning of Region 2
1070=  1    6    0
1080=       50.010070        0.000000   Note overlap
1090=       50.010070        0.611856   in x
1100=       49.824493        3.408121
1110=       49.187645        6.642347
1120=       48.033119        9.803850
1130=       46.350802       12.845306
1140=  2    6    0
1150=       46.350802       12.845306
1160=       44.232117       15.282759
1170=       41.700586       17.005585
1180=       39.201430       18.276600
1190=       36.756043       18.921902
1200=       34.530315       19.328766
1210=  3   10    0
1220=       34.530315       19.328766
1230=       32.000067       21.765604
1240=       29.693047       26.516281
1250=       26.462524       31.830555
1260=       23.469040       37.253181
1270=       21.147877       42.000042
1280=       20.279148       43.921326
1290=       19.470787       46.044807
1300=       18.717270       48.341698
1310=       17.326115       53.300514
1320=  4   18    2
1330=       17.326115       53.300514
1340=       15.954680       59.162170
1350=       14.366322       65.990017
1360=       13.750040       68.714081
1370=       13.102150       71.150070
1380=       12.580235       73.730709
1390=       11.930618       76.311203
1400=       11.212303       78.891495
1410=       10.451151       81.471466
```

Table 1. Geometry Data (continued)

Table 1. Geometry Data (concluded)

ORIGINAL PAGE IS OF POOR QUALITY

25510=	1	7	0
25520=		50.007126	0.000000
25530=		50.007005	2.742030
25540=		50.005126	4.056260
25550=		49.737724	7.001527
25560=		48.050752	10.803637
25570=		47.682236	13.590180
25580=		45.074167	16.014576
25590=	2	8	0
25600=		45.074167	16.014576
25610=		43.074220	18.105961
25620=		41.436036	19.800280
25630=		38.710070	21.075272
25640=		35.700315	21.860430
25650=		32.715111	22.127040
25660=		26.350161	22.127010
25670=		18.414303	22.127010
25680=	3	10	0
25690=		18.414303	22.127010
25700=		17.014026	26.774406
25710=		16.023122	34.597084
25720=		16.013002	43.651665
25730=		15.325804	52.031616
25740=		14.666430	60.607010
25750=		13.910023	68.560077
25760=		13.272402	76.311203
25770=		12.510040	84.051117
25780=		11.722006	91.780300
25790=	4	4	2
25800=		11.722006	91.780300
25810=		10.950051	99.526520
25820=		10.160640	107.263123
25830=		9.650802	112.420068
25840=	5	3	1
25850=		9.650802	112.420068
25860=		7.128304	115.000000
25870=		5.126203	117.056802
25880=	6	20	2
25890=		5.126203	117.056802
25900=		5.863470	125.842810
25910=		6.825004	133.837067
25920=		8.132015	145.820040
25930=		8.700452	153.823050
25940=		9.262057	161.816005
25950=		9.645007	173.806427
25960=		9.760035	181.708708
25970=		9.833300	193.786194
25980=		9.705547	201.776581
25990=		9.630101	209.766571
26000=		9.360663	217.755157
26010=		8.956104	226.046530
26020=		8.400652	233.006222
26030=		7.640726	239.003317
26040=		6.840140	245.704103
26050=		5.740037	251.073320
26060=		5.230540	254.410126
26070=		4.662115	256.886719
26080=		4.369453	258.021057
26090=		4.007445	259.004211
26100=		3.846460	259.842285
26110=		3.616040	260.540894
26120=		3.404075	261.104058
26130=		3.210036	261.537354
26140=		3.102206	261.720047
26150=		3.030006	261.841614
26160=		2.861572	262.010060
26170=		2.697000	262.070618
26180=	7	30	1
26190=		2.697000	262.070618
26200=		2.627072	262.051607
26210=		2.560114	262.012207
26220=		2.513620	261.944702
26230=		2.450703	261.845093
26240=		2.410055	261.742126
26250=		2.370328	261.570125
26260=		2.310113	261.204773
26270=		2.270214	260.722534
26280=		2.253210	260.120967
26290=		2.242231	259.410250
26300=		2.243005	258.501707
26310=		2.257326	257.642151
26320=		2.282346	256.567322
26330=		2.360320	254.016083
26340=		2.521813	250.454346
26350=		2.700336	246.293015
26360=		3.116043	235.725586
26370=		3.460006	221.033502
26380=		3.830403	206.437592
26390=		4.006307	195.451019
26400=		4.351102	184.462091
26410=		4.600018	173.473604
26420=		4.703060	166.146042
26430=		4.687814	158.819366
26440=		4.501020	151.102020
26450=		3.955367	140.490260
26460=		3.023705	127.743134
26470=		2.402436	116.711746
26480=		1.313310	114.000817
26490=	8	13	0
26500=		1.313310	114.000817
26510=		0.154016	112.420014
26520=		0.021608	107.263168
26530=		-0.209001	99.526550
26540=		-0.567024	89.210120
26550=		-0.840183	81.471390
26560=		-1.263398	71.150055
26570=		-1.606522	59.600020
26580=		-2.104604	50.220080
26590=		-2.726031	40.460147
26600=		-3.224504	28.843316
26610=		-3.657440	16.530712
26620=		-3.791033	0.000000
26630=		510.704012	0
26640=	1	7	0
26650=		50.007103	0.000000

APPENDIX B — PUBLICATIONS

This Appendix contains the following papers:

1. J. of Aircraft, Vol. 24, February 1987, pp. 73–83.
2. AIAA Paper 86–0244
3. AIAA Paper 86–1834
4. AIAA Paper 87–0592
5. AIAA Paper 86–0440

Permission to reprint the papers appearing in this Appendix was granted by the AIAA.

At the 24th Aerospace Sciences Meeting, where this paper was presented, we learnt from R. W. Newsome that 1) Newsome had tried fixed time steps with his MacCormack scheme formulation and still observed large cross-flow separation, and 2) Newsome had also tried spatially varying time steps with the upwind-biased formulation and yet failed to observe large separation. In this paper, we have shown that variable time steps can lead to a different solution for the delta-wing problem considered. But this conclusion may be valid for only certain ranges of recipes for varying the time step spatially and over the sequence of time steps. While this may serve as a good counter-example (to the argument that the solution with large cross-flow separation is only due to numerical diffusion on the coarse grid), more research must surely be performed to understand other possible mechanisms.

8.0 References

[1] R. W. Newsome, "A Comparison of Euler and Navier-Stokes Solutions for Supersonic Flow Over a Conical Delta Wing," AIAA Paper 85-0111.

[2] S. R. Chakravarthy and S. Osher, "A New Class of High Accuracy TVD Schemes for Hyperbolic Conservation Laws," AIAA Paper 85-0363.

[3] S. R. Chakravarthy and K. Y. Szema, "An Euler Solver for Three-Dimensional Supersonic Flows with Subsonic Pockets," AIAA Paper 85-1703.

[4] R. W. Newsome and J. L. Thomas, "Computation of Leading-Edge Vortex Flows," presented at the Vortex Flow Aerodynamics Conference, NASA Langley Research Center, October 8-10, 1985, Hampton, Virginia.

[5] W. Kordulla and M. Vinokur, "Efficient Computation of Volume in Flow Predictions," *AIAA Journal*, Vol. 21, No. 6, June 1983.

[6] R. G. Burman, Naval Weapons Center, China Lake, California, private communication.

[7] S. R. Chakravarthy and S. Osher, "Computing With High-Resolution Upwind Schemes for Hyperbolic Equations", to appear in the Proceedings of the 1983 AMS-SIAM Summer Seminar on Large-Scale Computations in Fluid Mechanics, published by American Mathematical Society in *Lectures in Applied Mathematics*, Volume 22, 1985.

[8] P. L. Roe, "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357-372.

[9] S. R. Chakravarthy, "Relaxation Methods for Unfactored Implicit Upwind Schemes," AIAA Paper 84-0165, 1984; also to appear in the AIAA Journal.

[10] S. R. Chakravarthy, "Numerical Implementation of Boundary Conditions for Hyperbolic Systems of Conservation Laws," in preparation.

[11] P. Kutler, "Computation of Three-Dimensional, Inviscid Supersonic Flows," *Lecture Notes in Physics*, Springer-Verlag, No. 41, Progress in Numerical Fluid Dynamics, 1975, pp. 287-374.

Standard Bibliographic Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-4085		

4. Title and Subtitle	5. Report Date
Supersonic Flow Computations Over Aerospace Configurations Using An Euler Marching Solver	July 1987
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Kuo-Yen Szema, Sukumar Chakravarthy, and Vijaya Shankar	
9. Performing Organization Name and Address	10. Work Unit No.
Rockwell International Science Center Thousand Oaks, CA 91360	505-68-91-02
	11. Contract or Grant No.
	NAS1-15820
12. Sponsoring Agency Name and Address	13. Type of Report and Period Covered
National Aeronautics and Space Administration Washington, DC 20546	Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Langley Technical Monitors: Noel A. Talcott, Jr., and Kenneth M. Jones

16. Abstract

For fully supersonic flows, an efficient strategy for obtaining numerical solutions is to employ space-marching techniques. At low supersonic Mach numbers, realistic fighter configurations give rise to subsonic pockets near the canopy, wing-body junction, wing leading edge, and wing tip regions. A full potential marching technique, known as the SIMP code and capable of handling such embedded subsonic regions, has achieved some success analyzing low supersonic Mach number flows. The SIMP code, however, is not capable of handling strongly shocked flows with rotational and vortex effects due to the underlying isentropic assumptions.

This paper presents the extension of the full potential approach to the Euler equations which model the exact nonlinear inviscid gas dynamic flow processes. Within the assumption of an inviscid flow, such an Euler marching solver can be applied to a wide class of shocked flows including the hypersonic range. The intent is to maintain some of the basic features of the full potential SIMP code within the Euler solver in dealing with geometry input, gridding techniques, and input/output routines including post processing of results. The Euler marching code known as EMTAC has been developed. Results obtained for a variety of configurations involving canard, wing, horizontal tail, flow-through inlet, and fuselage using both the EMTAC and SIMP codes are reported. For shocked cases satisfying the isentropic assumption, the EMTAC and SIMP codes produced practically identical results. In terms of execution time, the EMTAC code is 5 to 10 times slower than the SIMP code since the Euler formulation solves five equations involving block tridiagonal inversions. Both codes are operational on the CRAY-XMP and VPS-32 supercomputers.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Aerodynamics Numerical Methods Aircraft Design/Analysis Euler Equations	[redacted] Subject Category 02

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified		

[redacted]