INFORMATION TO USERS

This reproduction was made from a copy of a document sent to us for microfilming. While the most advanced technology has been used to photograph and reproduce this document, the quality of the reproduction is heavily dependent upon the quality of the material submitted.

The following explanation of techniques is provided to help clarify markings or notations which may appear on this reproduction.

1. The sign or "target" for pages apparently lacking from the document photographed is "Missing Page(s)". If it was possible to obtain the missing page(s) or section, they are spliced into the film along with adjacent pages. This may have necessitated cutting through an image and duplicating adjacent pages to assure complete continuity.

2. When an image on the film is obliterated with a round black mark, it is an indication of either blurred copy because of movement during exposure, duplicate copy, or copyrighted materials that should not have been filmed. For blurred pages, a good image of the page can be found in the adjacent frame. If copyrighted materials were deleted, a target note will appear listing the pages in the adjacent frame.

3. When a map, drawing or chart, etc., is part of the material being photographed, a definite method of "sectioning" the material has been followed. It is customary to begin filming at the upper left hand corner of a large sheet and to continue from left to right in equal sections with small overlaps. If necessary, sectioning is continued again—beginning below the first row and continuing on until complete.

4. For illustrations that cannot be satisfactorily reproduced by xerographic means, photographic prints can be purchased at additional cost and inserted into your xerographic copy. These prints are available upon request from the Dissertations Customer Services Department.

5. Some pages in any document may have indistinct print. In all cases the best available copy has been filmed.

University Microfilms International

300 N. Zeeb Road
Ann Arbor, MI 48106

1320923

NEBECK, HOWARD EDWARD

THREE-DIMENSIONAL GRIDS FOR AERODYNAMIC APPLICATIONS

THE UNIVERSITY OF ARIZONA M.S. 1983

University
Microfilms
International 300 N. Zeeb Road, Ann Arbor, MI 48106

THREE DIMENSIONAL GRIDS FOR AERODYNAMIC APPLICATIONS

by

Howard Edward Nebeck

A Thesis Submitted to the Faculty of the

DEPARTMENT OF AEROSPACE AND MECHANICAL ENGINEERING

In Partial Fulfillment of the Requirements
For the Degree of

MASTER OF SCIENCE

In the Graduate College

THE UNIVERSITY OF ARIZONA

1 9 8 3

STATEMENT BY AUTHOR

This thesis has been submitted in partial fulfullment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this thesis are allowable without special permission, provided that accurate acknowledgement of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his judgement the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Howard E Nebeck

APPROVAL BY THESIS DIRECTOR

This thesis has been approved on the date shown below:

K. Y. FUNG
Professor of Aerospace and
Mechanical Engineering

May 12, 1983
Date

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Professor K. Y. Fung for his guidance and counsel during the course of this work. Dr. Helmut Sobieczky is thanked for his assistance in implementing the computer software necessary, and Dr. S. S. Sritharan is especially thanked for his many helpful suggestions. Finally, Mrs. Laraine Seitz is thanked for her skillful typing of this thesis.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

ABSTRACT

A simple algebraic method is studied for parameterizing a wing fuselage surface and generating a three dimensional grid around this surface. Preliminary evaluation of the surface coordinate system is undertaken by using it to create a wing fuselage for input to a transonic finite volume analysis code. The attributes of the three dimensional grid are discussed and suggestions for improving it are made.

CHAPTER 1

INTRODUCTION

The aerodynamic properties of various objects can be computed numerically by solving an approximation to the Navier-Stokes equations using finite difference techniques. These finite difference algorithms require that a rational system of grid points surround the object in question. Typically these grid points represent intersection points of curvilinear coordinate curves constructed around the object. Thus the problem of mesh generation is really solved by establishing curvilinear coordinate system about the body, then choosing certain values of the coordinates to be grid points. In two dimensions the above process is highly refined, and several techniques exist to generate "tailor made" grids about a wide variety of two dimensional shapes. However, in three dimensions the problem is harder due in part to the geometric complexity of the bodies.

1.1 The Computational Problem

This report deals with three dimensional grid generation about transport aircraft configurations minus the empennage and engines. The aircraft is immersed in a steady, compressible, and irrotational airstream with regions of both subsonic and supersonic flow (the transonic regime). In such an irrotational flow, the Euler equations can be combined to form a single second order equation for velocity potential

$$(p\phi_x)_x + (p\phi_y)_y + (p\phi_z)_z = 0 \qquad (1.1)$$

where ϕ is velocity potential $\underline{U} = (u\ v\ w) = \nabla\phi$, and p is the density. Only one half of the flow field needs to be studied, due to symmetry. Thus the body of interest is a wing fuselage combination, as depicted in Figure 1. The remaining boundaries of the computational region are the symmetry plane, the outer boundary, and the inflow-outflow planes.

Two coordinate systems are needed in this region. Rectangular cartesian coordinates (r.c.c.) provide a comprehensible frame to describe the wing fuselage geometry and to display the flow variable results. However, the r.c.c. system is not appropriate for discretizing the region, principally because it does not conform to the highly curved boundaries. Instead a curvilinear system is employed to accomodate the differencing scheme. Each boundary of the region is a curvilinear coordinate surface. Coordinate curves do not intersect orthogonally and are not necessarily parameterized by arc length. Equation (1.1) is solved by differencing in the directions defined by these coordinate curves. Grid points can be viewed as a discrete description of the curvilinear system. They are located at equal increments of the coordinate and so their spacing represents the parameterization used along the curve. Each grid point has unique curvilinear (hereafter called computational) coordinates, typically labeled i,j,k and rectangular coordinates x,y,z. The functional relation between coordinates can be stated as

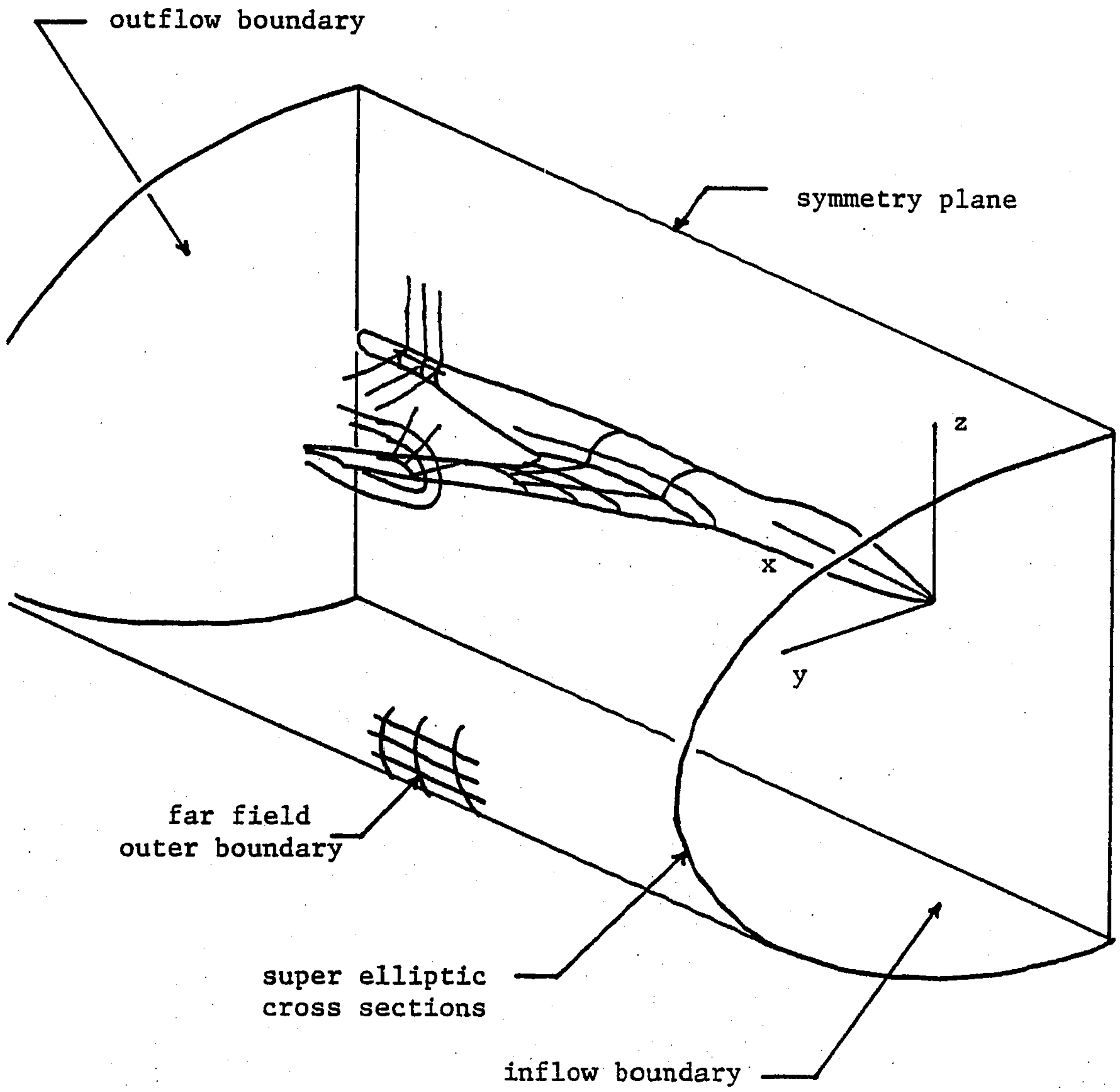


Figure 1. Computational Region

$$\begin{aligned} x &= x(i,j,k) & i &= i(x,y,z) \\ y &= y(i,j,k) & j &= j(x,y,z) \\ z &= z(i,j,k) & k &= k(x,y,z) \end{aligned} \tag{1.2}$$

Or more concisely as

$$x^i = x^i\,(e^j\,) \qquad e^i = e^i\,(x^j\,)$$

where $x = x^1$ $y = x^2$ $z = x^3$ $i = e^1$ $j = e^2$ $k = e^3$. These equations define a transformation (mapping) between the two systems. Establishing a desirable computational system and then determining the transformation are the major problems of grid generation.

Frequently computational coordinates are displayed as a rectangular cartesian system in a "warped" space, called computational space, logical space, or parameter space. Such a convention may facilitate development and coding of finite difference schemes, but gives little insight into the geometric nature of the grid generation problem. Indeed, an over reliance on the logical space diagram can mask the physical space characteristics of the grid. This display technique will not be used in the sequel.

1.2 Inclusion of the Fuselage

The elliptic nature of Equation (1.1) means that boundary conditions will be paramount in determining its solution. Of primary concern is the implementation of the Neumann boundary condition on the body, that is mass flux cannot penetrate the body surface. To enforce this boundary condition, the relation

$$\underline{V} \cdot \underline{n} = 0$$

where n is the unit normal

must hold on the surface. Grid lines, hence volume elements, must be dense enough on the body to allow an accurate computation of $\underline{V}$ and $\underline{n}$. Additionally, all important surfaces should be included as boundaries of the region.

Aircraft flowfields are primarily determined by the wing and airplanes have been modeled in the past by a wing alone, with a symmetry boundary plane replacing the fuselage. In this configuration, the wing root "sees" an undisturbed freestream flow. A wing fuselage combination more accurately describes real aircraft, primarily because the wing root flow has been deflected and accelerated by the fuselage. This effect can initiate a shock wave which propagates down the upper surface of the wing, and thus influences the flow far away from the wing fuselage junction. Yet the importance of including a fuselage is subject to debate. Figure 2 shows that the fuselage boundary condition and the symmetry plane boundary condition are really quite similar. A wing mounted well behind the fuselage nose resembles a "wing on a wall" for the purposes of computational aerodynamics[1]. However, the wide variation in the amount of sculpturing used to blend the wing and body surfaces seen in Figure 3 and Figure 4 suggests that the wing fuselage interface is indeed an important boundary and so the fuselage should be included in the model.

A grid around a wing fuselage combination is considerably more difficult to construct than one around a wing alone. The rapid (sometimes discontinuous) change in surface slope and curvature directions at the wing root discourages continuing the wing parameterization onto the fuselage surface. Yet creating a continuous grid requires that

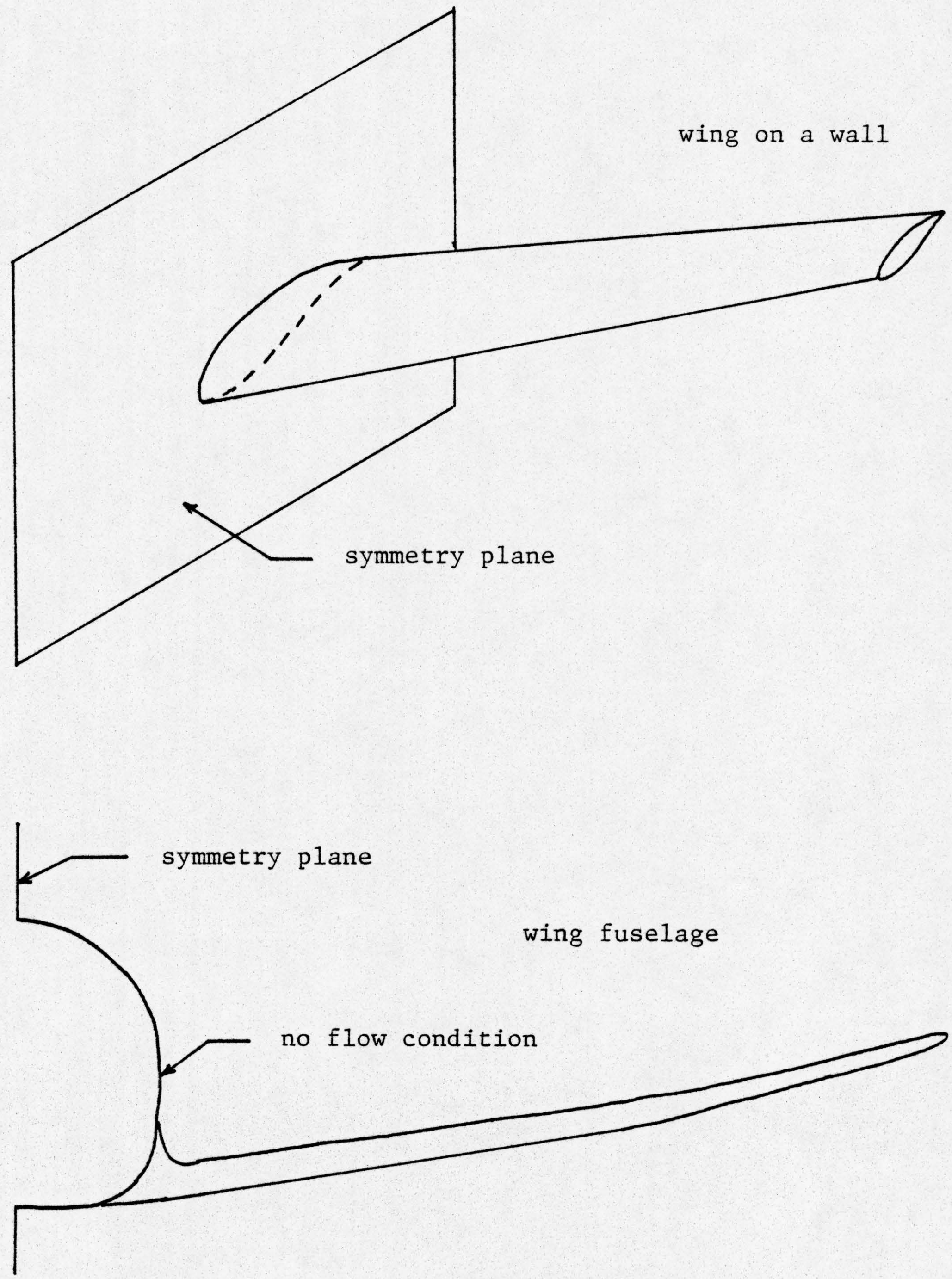


Figure 2. Boundary Condition at the Wing Root

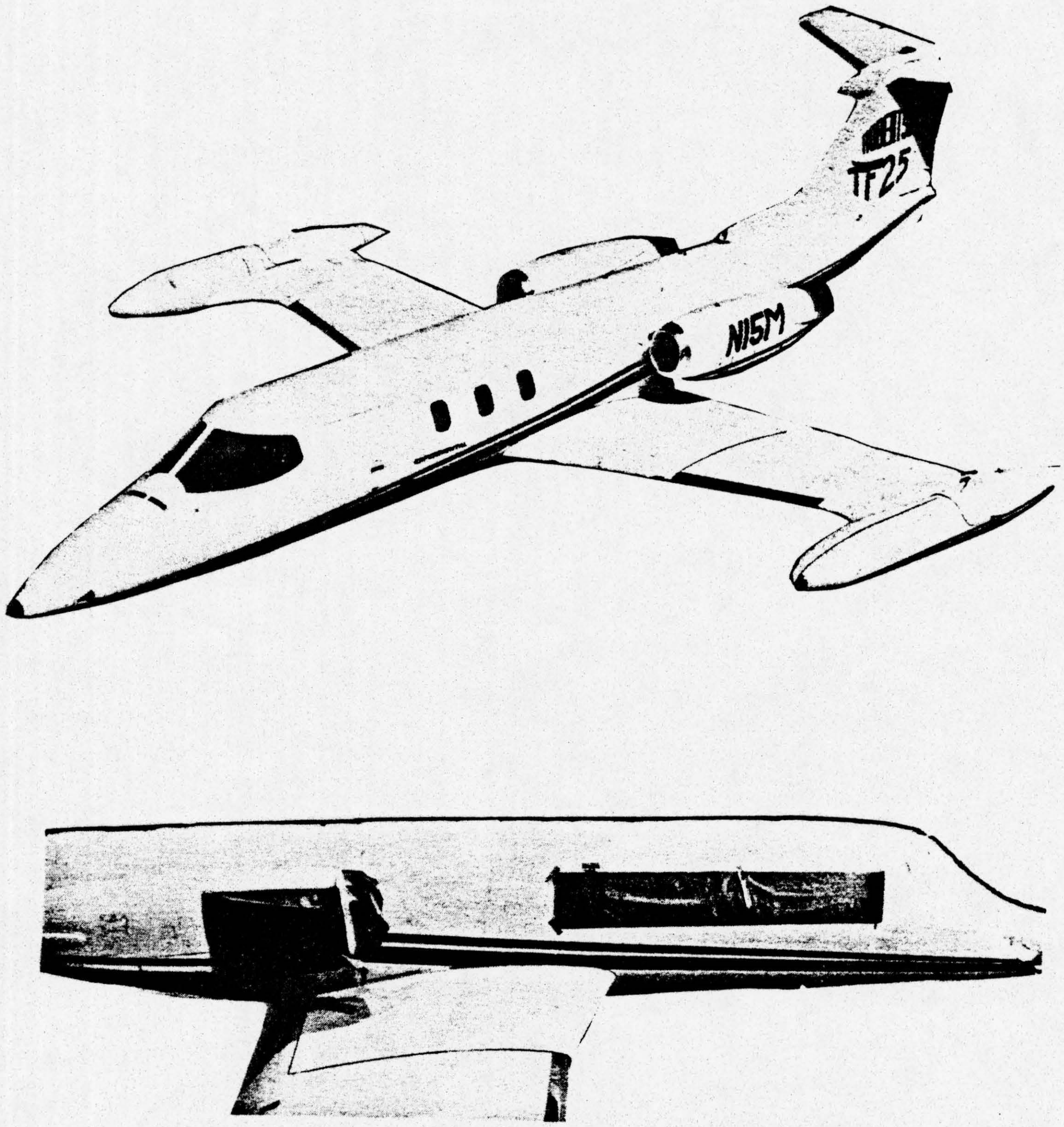


Figure 3. Wing Fuselage Junction Without Sculpturing

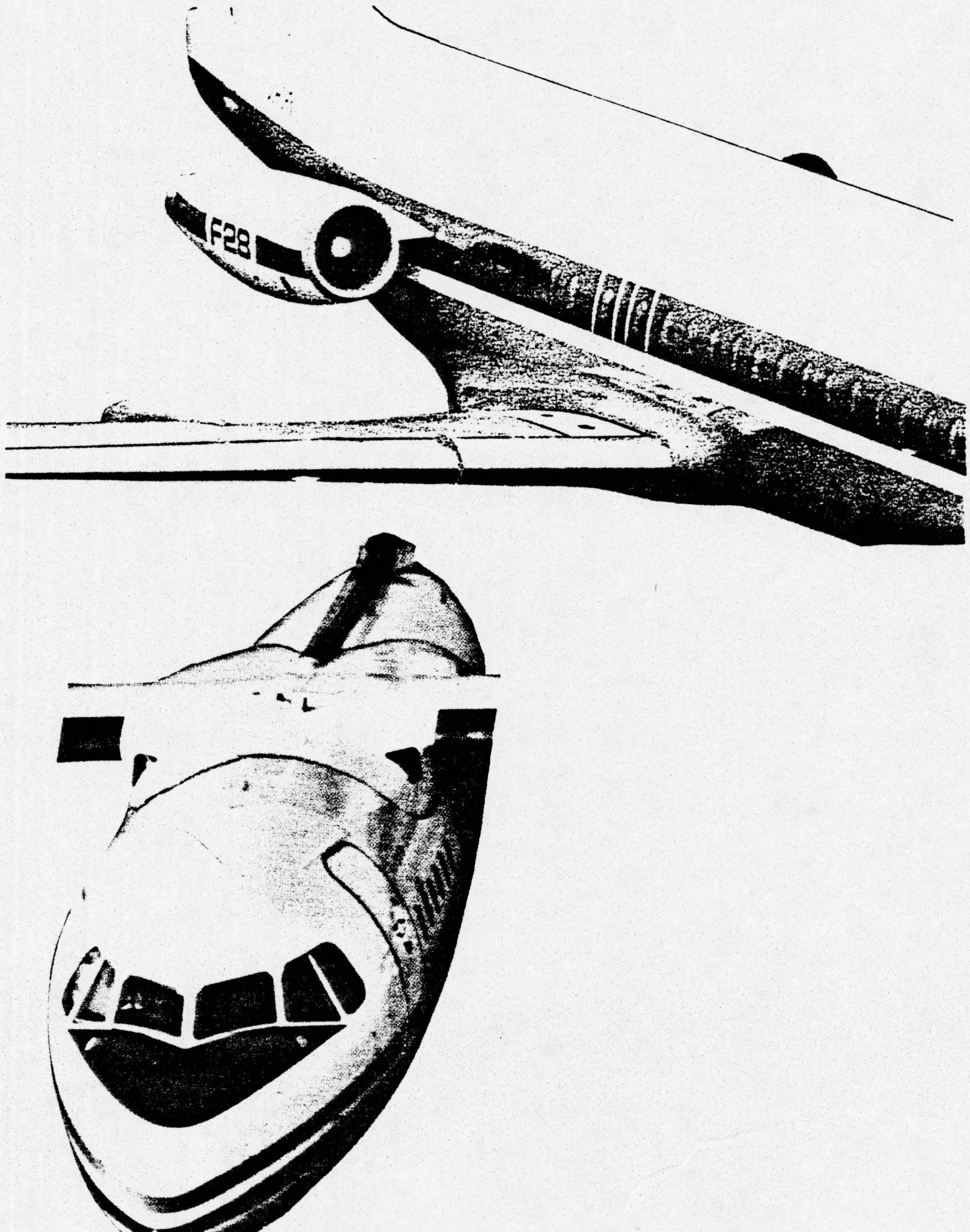


Figure 4. Wing Fuselage Junction With Sculpturing

this be done. The curvilinear coordinate "normal" to the surface may change, as shown in Figure 5. The fuselage surface has become a doubly connected region due to the wing root, and so a "cut" in the surface is necessary to ensure a single valued coordinate system. Other problem areas are the fuselage nose and tail, where the surface collapses to a line. These are the difficulties that have previously discouraged computations involving Equation (1.1) about a wing fuselage model.

1.3 Finite Difference Techniques

Conventional finite difference techniques approximate the derivatives du/dx , du/dy by the differences $\Delta u/\Delta x$, $\Delta u/\Delta y$ plus higher order terms, where $\Delta u, \Delta x, \Delta y$ are the differences in the respective variables between consecutive grid points. Including the higher order terms or modeling higher order derivatives requires using more than two consecutive grid points. Obviously, this construction implies that the x and y directions are defined throughout space and that differences will be computed along lines of constant x and y. Additionally, the intervals Δx , Δy should remain constant for coding efficiency and ease in error analysis. Figure 6 illustrates a two dimensional mesh designed to meet these needs.

In irregular regions curvilinear coordinates are employed to create a grid which conforms to the boundaries and clusters grid points in the desired areas (see Figure 6). Differences are taken along the curvilinear coordinate curves s and t. Employing the curvilinear system requires that the differential equation and the boundary conditions be rewritten in the new coordinates. Thus the coordinate system must

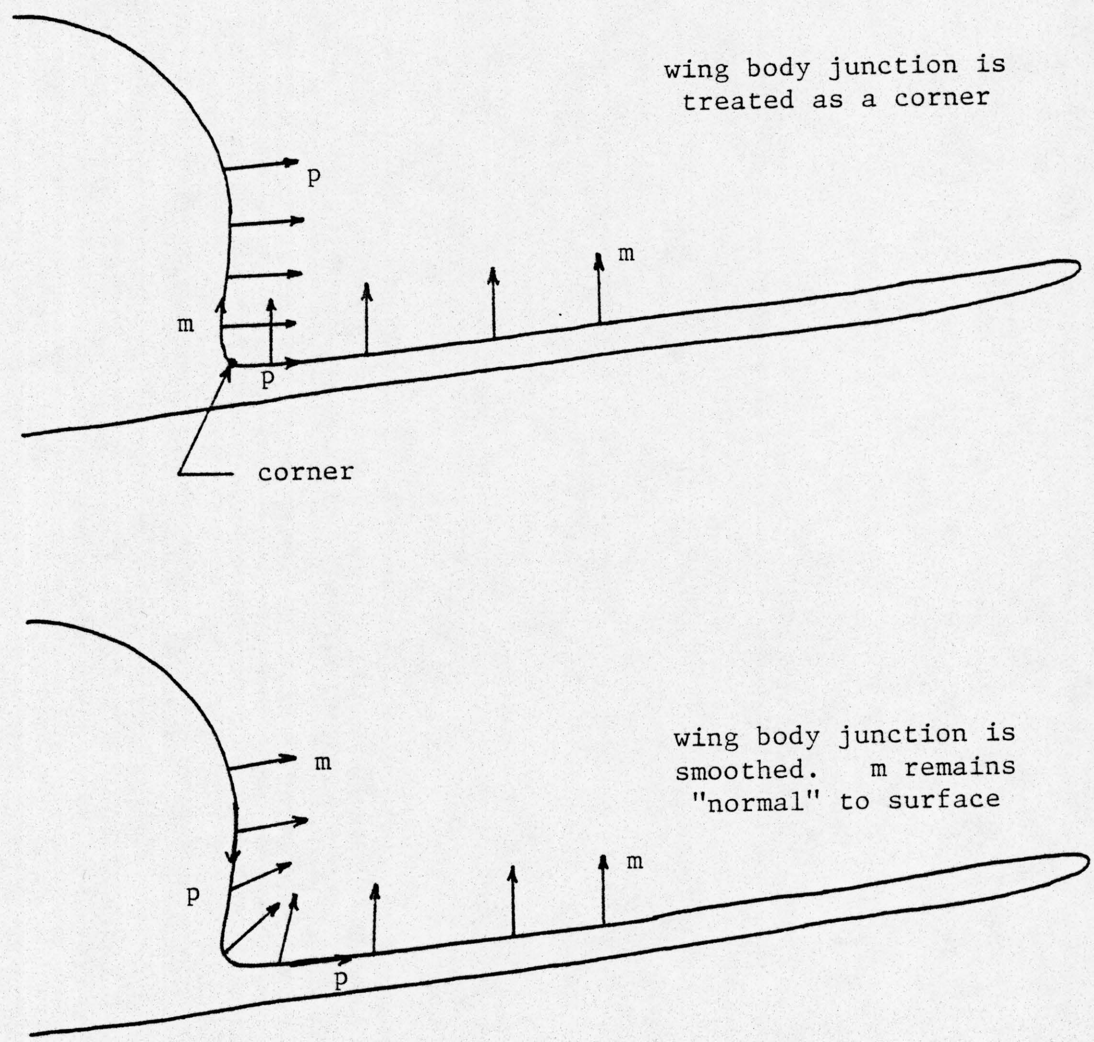
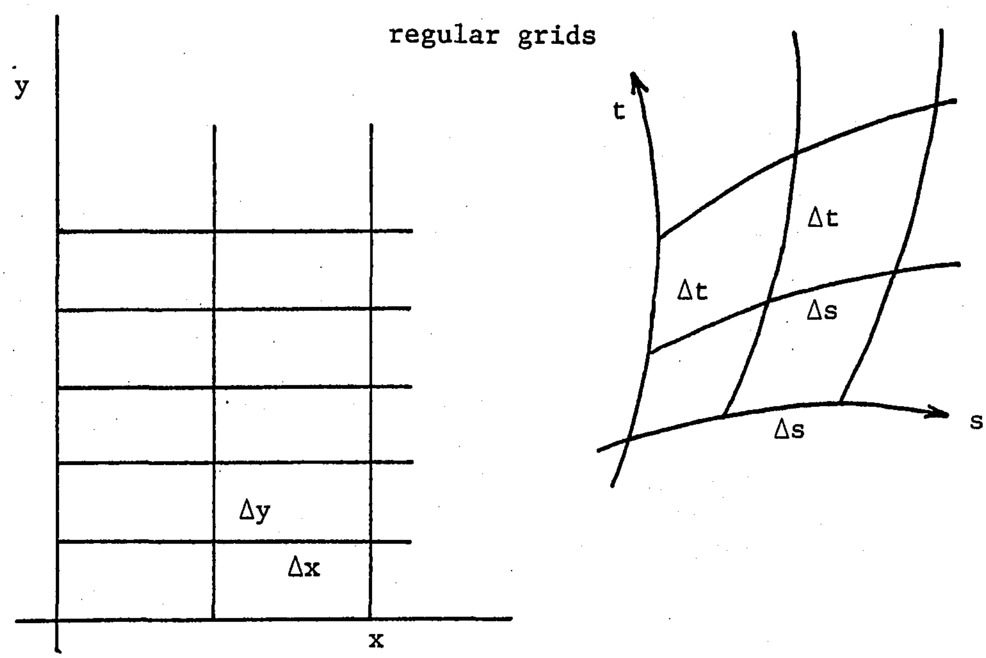


Figure 5. Choice of Surface Parameters



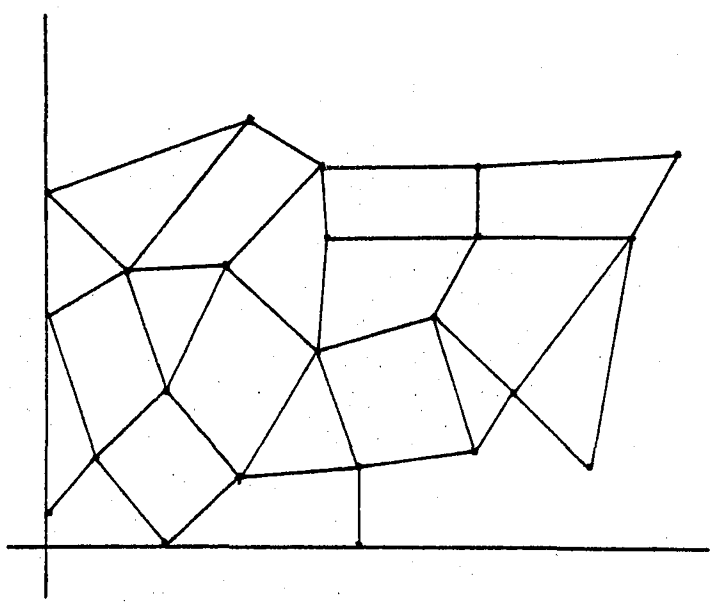


Figure 6. Regular and Irregular Grids

be generated by a global transformation which can be applied to the differential equation as well. Furthermore, changing the shape of the computational region or substantially altering the grid spacing requires that a new mapping be devised. Consequently the transformed differential equation changes and a new differencing algorithm must be written.

These two drawbacks of conventional finite difference procedures are covercome by using the finite volume[2] approach to solving Equation (1.1). The finite volume technique decouples the partial differential equation from the grid generation process. Only the grid point locations need to be passed to the solution algorithm, not the global coordinate transformation which created them. Thus any method can be used to establish a reasonable grid without regard for the curvilinear system represented by the grid.

The finite volume method uses the grid to divide physical space into distorted cubes (volume elements), each one defined by the eight grid points at its vertices (Figure 7). Every volume is endowed with its own curvilinear coordinate system, created via a transformation which takes the eight grid points into those of a unit cube. This transformation is algebraically identical for each volume. The governing equation is rewritten in this local coordinate system and the dependent variable is calculated at the grid points. Applying this algorithm successively to each cube accomplishes one iteration sweep through the mesh. It can be shown that this differencing scheme is equivalent to enforcing a mass flux balance on secondary cubes, whose vertices are the midpoints of the primary cubes. Since the governing equation only seems the local trnasformation of the volumes, a single

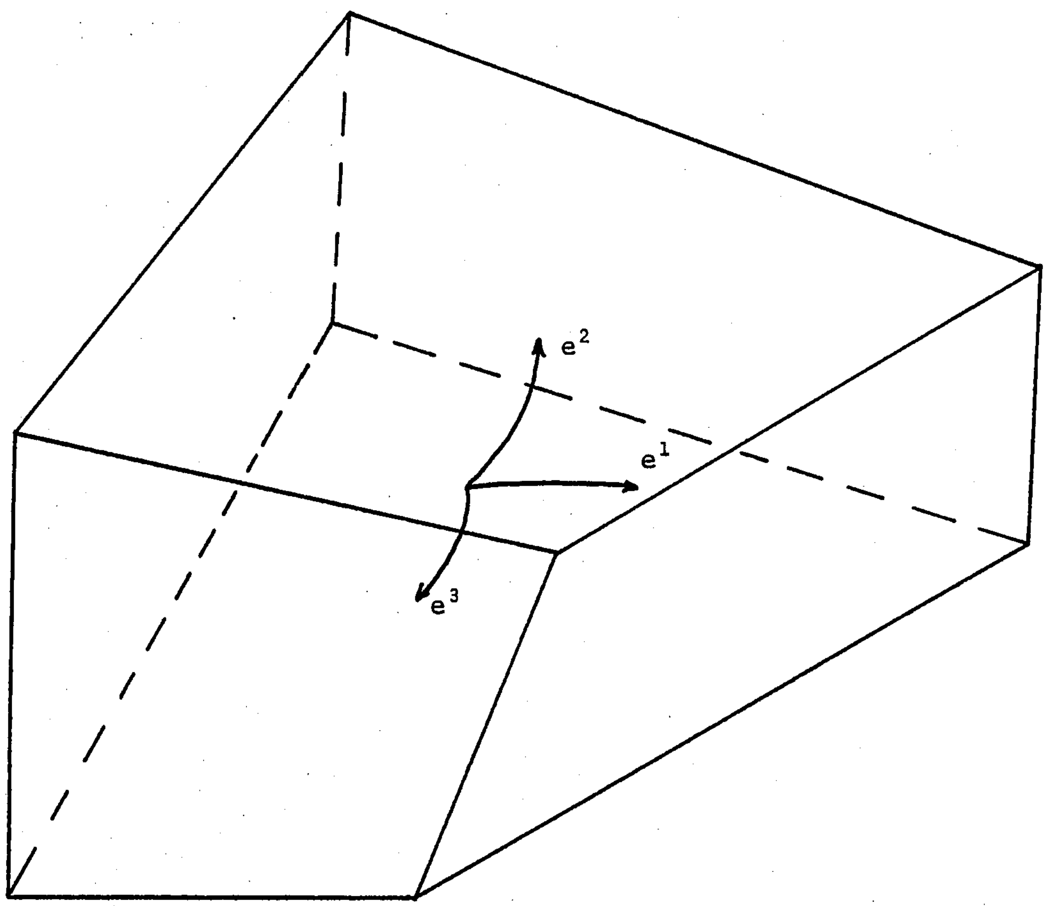


Figure 7. Cube in Curvilinear Coordinates

differencing algorithm (that is, code) can be used for a variety of physical regions, each with its own unique global coordinate system.

1.4 Requirements

The wing body mesh system discussed here is entended to be used with a finite volume computer code[3] that solves Equation (1.1). Therefore any method of distributing the grid points will be acceptable so long as the volume elements created are geometrically similar, that is, each is a cube rather than, say, a tetrahedron. The task of the grid generator is then to create an acceptable mesh around the wing body configuration and pass the grid point locations to the finite volume code.

Accuracy of the finite volume solution depends in part on the quality of the grid. A grid for computing the wing fuselage flowfields should have the following features.

1.4.1 Boundaries

The physical surface should also be a coordinate surface, so that boundary conditions can be applied accurately.

1.4.2 Spacing

Grid points, hence the volume elements, should be clustered near the surface, where the flow variables change most rapidly. Conversely, the grid far away from the body should be less dense, because flow variables change slowly in this region.

1.4.3 Size

To avoid introducing excessive truncation error, adjacent volume elements must be similar in size and shape.

1.4.4 Orthogonality

The curvilinear system should be as orthogonal as possible. Volume elements are then nearly cubical in shape and the discretization of the equation is more compatable with the local transformation. In a broader sense, orthogonality of the curvilinear system means that differences taken in each direction retain maximum independence from each other. This accelerates convergence of the finite difference calculations.

CHAPTER 2

COORDINATE TRANSFORMATIONS

All techniques of grid generation are involved with constructing the transformation of Equation (1.2). The final relation may or may not have a closed form expression. Frequently several intermediate transformations are required to incorporate all of the features desired in the curvilinear system.

The first step in establishing the transformation is to define the region of interest and identify all relevant boundaries. These boundaries are parameterized in a manner consistant with the desired grid spacing. For example, the curve defining an airfoil surface may have a parameter which changes most rapidly (with respect to arc length) in regions of high curvature. This would effectively cluster grid lines leaving the surface in such areas. Three dimensional (3-D) regions are bounded by two dimensional surfaces, which are each bounded by one dimensional space curves. So the boundary parameterization process is carried out twice for a 3-D region.

Once the boundaries are parameterized, several methods could be used to generate the curvilinear system in the computational space. Most of these can be classified as algebraic procedures or elliptic equation procedures.

2.1 Algebraic Procedures

Methods which seek to explicitly construct the coordinate transformation functions through a series of analytic mappings and interpolations are termed algebraic grid generation procedures.

Interpolation of boundary data is perhaps the most widely used algebraic technique. The versatility (and complexity) of these procedures depends upon the order of the interpolant used and the manner of incorporating all the boundary data into the interpolation formula. The transfinite interpolation scheme of Gordon and Hall[4] and coordinate generation of Cook[5] are examples of transformations constructed by interpolation alone.

An elaborate construction technique using interpolation between many auxiliary boundary surfaces is advocated by Eiseman[6]. The specification of these supplementary boundary curves within the computational region allows Eiseman to use higher order interpolants and thereby precisely specify various aspects of the grid. However, his procedure may be difficult to implement in three dimensions.

2.2 Elliptic Equation Methods

Curvilinear coordinates in a closed region can be viewed as level curves of an elliptic boundary value problem. The parameterizations used on the boundaries then specify Dirichlet conditions for the region. The transformation functions of Equation (1.2) are then solutions of equations such as

$$a^i e^i_{xx} + b^i e^i_{yy} + c^i e^i_{zz} = q^i \qquad i=1,2,3$$

Typically it is the inverse system

$$A^i x^i_{rr} + B^i x^i_{ss} + C^i x^i_{tt} = Q^i \qquad (2.1)$$

$$r = e^1 \quad s = e^2 \quad t = e^3 \qquad\qquad i = 1,2,3$$

which is actually solved numerically, since boundary data is easier to specify on the regular r, s, and t surfaces in the transformed space than on the irregular, curved boundaries in physical space. The forcing functions (q^i) are chosen to control the spacing of the coordinate curves. Any numerical technique can be chosen to solve equations (2.1).

Elliptic equation methods have been used extensively in two dimensions[7,8], where choice of the clustering functions is well understood[9]. At least one successful 3-D elliptic grid generator has also been reported for wing body configurations[10].

CHAPTER 3

AN ALGEBRAIC GRID GENERATION PROCEDURE

In this chapter, an algebraic procedure is described for generating a three dimensional grid about a wing fuselage combination, like the one shown in Figure 1. Note the reference cartesian coordinate system. This procedure was conceived by Dr. H. Sobieczky[11] and is implemented in a Fortran computer code, referred to herein as E88. It is similar to the coordinate transformation of Eriksson[12]. E88 is based on the idea that the end result of any three dimensional grid generator is a collection of coordinate lines with grid points distributed along them. The coordinate line shapes and distribution functions can be approximated, at least locally, by simple algebraic functions. Sobieczky exploits this by using simple analytic functions to directly create all coordinate lines and to distribute grid points on these lines. The program requires a rather high degree of user interaction to successfully generate a grid.

E88 establishes curvilinear coordinates in the three dimensional computational region by creating two dimensional parameterized boundary surfaces in this region, then connects corresponding points of appropriate boundary surfaces by straight lines. On these connecting lines, a third coordinate can be established. The two dimensional surfaces are parameterized in an analogous manner, that is, coordinate points are established on their boundary curves, and corresponding points are connected with straight lines. So the three dimensional coordinate system is determined entirely by the parameterization of the many bounding curves present in the region. Figure 8 shows some of these boundaries. Grid points are distributed on these curves by using a simple analytic stretching function. Clustering points on the boundaries allows the grid to be "tailored" for higher resolution in certain areas. However, the linear connection between boundaries does not provide much control of grid characteristics in the interior of the region. Thus, some very desirable properties of more sophisticated grid generators, such as "near orthogonality" of the curvilinear system, are lost in this approach. Perhaps more serious is the radical departure from orthogonality (of the grid) at the body surface. The effect of such defects has not yet been established.

3.1 Utility Functions

One utility function used for both coordinate stretching and geometric definition is built by adding a scaled power law to a linear term. The resulting function can be controlled by manipulating four parameters.

$$\text{FORM1}(x;a,b,e,f) = ax + \{1- 1- x^{e}\}^{f}([1-b] + bx-ax)$$

a and b are the tangents to the curve at the left and right endpoints, while e and f control the curvature (in a non-obvious way). FORM1 can have at most one inflection point. Figure 9 shows some examples of FORM1. Generally, the endpoint tangents a, b will not be reflected in a discrete sampling of FORM1 values. Rather, the four parameters should be used together to specify the overall nature of the curve.

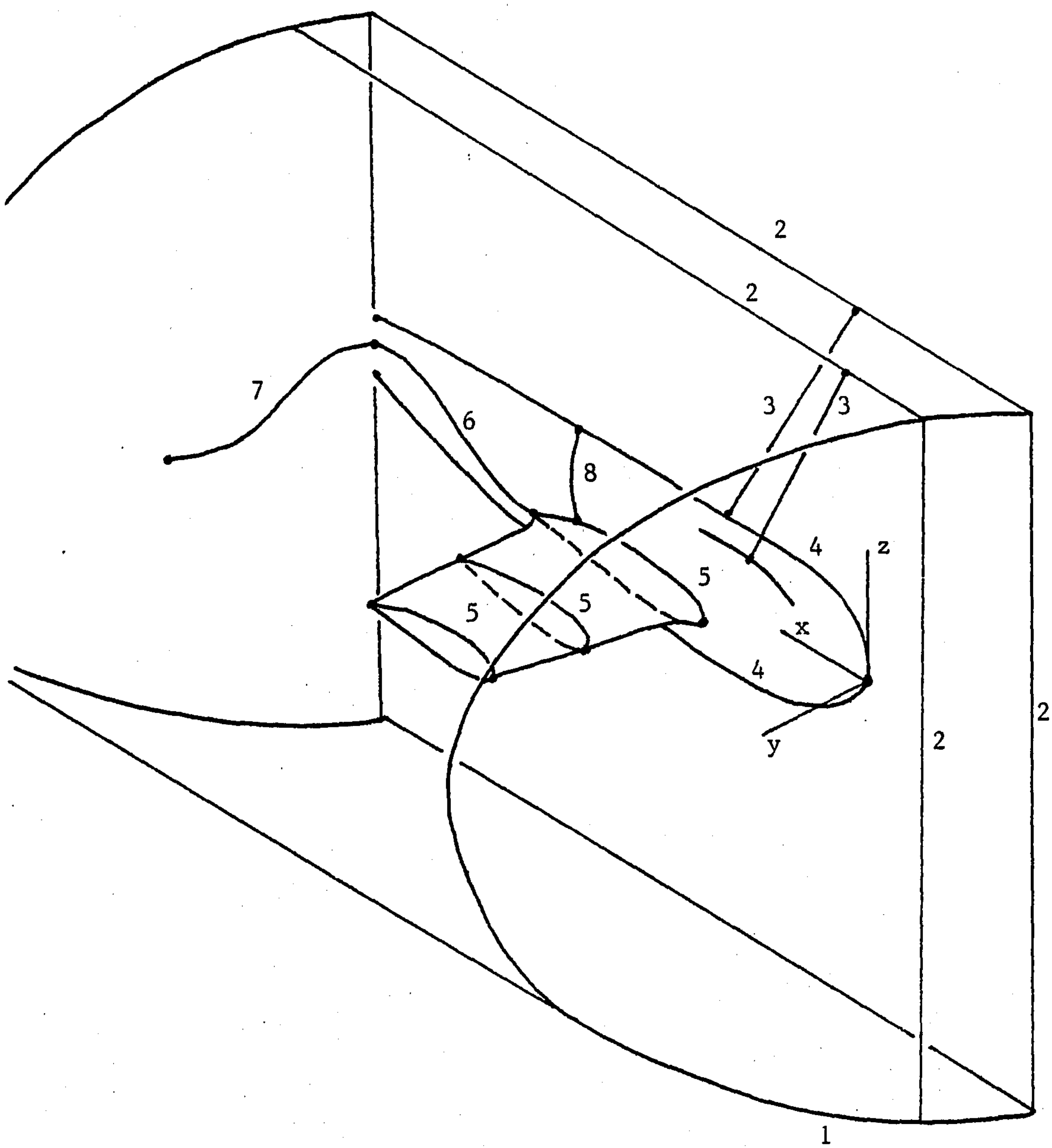


Figure 8. Boundary Curves on the Wing and Fuselage

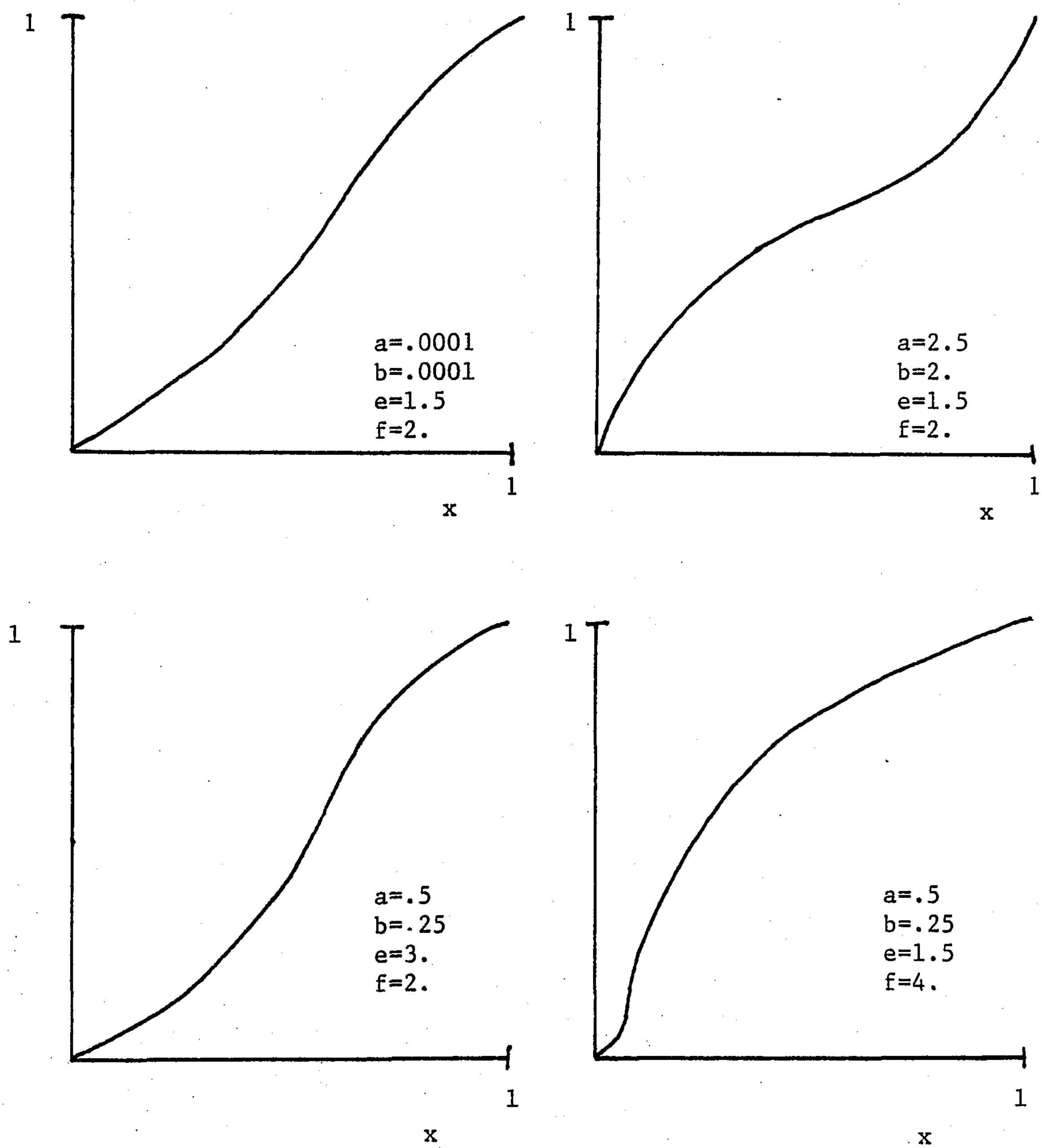


Figure 9. Examples of FORM1

A two parameter stretching function of known slope at each endpoint is also used frequently in E88. The slope is greater than 1 at one end, less than 1 at the other, and varies continuously in between. The curve has no inflection point and is again a general (non-integral) power law added to a linear term.

$$\text{FORM0}(x;a,b) = ax + (1-a)x^e \qquad e = \frac{a-b}{a-1}$$

$$0 \leqq \text{FORM0} \leqq 1 \qquad 0 \leqq x \leqq 1$$

Examples of FORM0 are given in Figure 10.

A third utility function is composed of FORM1 plus a linear relation.

$$\begin{aligned} \text{SHAPE}(x,x1,x2,x3;a,b,e,f) &= \text{FORM1}(x;a,b,e,f) && x1 \leqq x \leqq x2 \\ &= y2 + x((y3-y2)/(x3-x2)) && x2 \leqq x \leqq x3 \end{aligned}$$

E88 can also create a smooth function of x by a superposition of a bump and ramp function. The bump is a curve with zero slope at its endpoints and crest, and two inflection points. Crest height, crest location, and endpoint position can be specified. The ramp is created by the FORM1 function. If the addition of these two simple curves results in a non-zero ordinate at the right endpoint, then a step function can be used to extend this constant function for a specified additional distance. The bump or ramp can be used alone by simply specifying that the other has a height equal to zero.

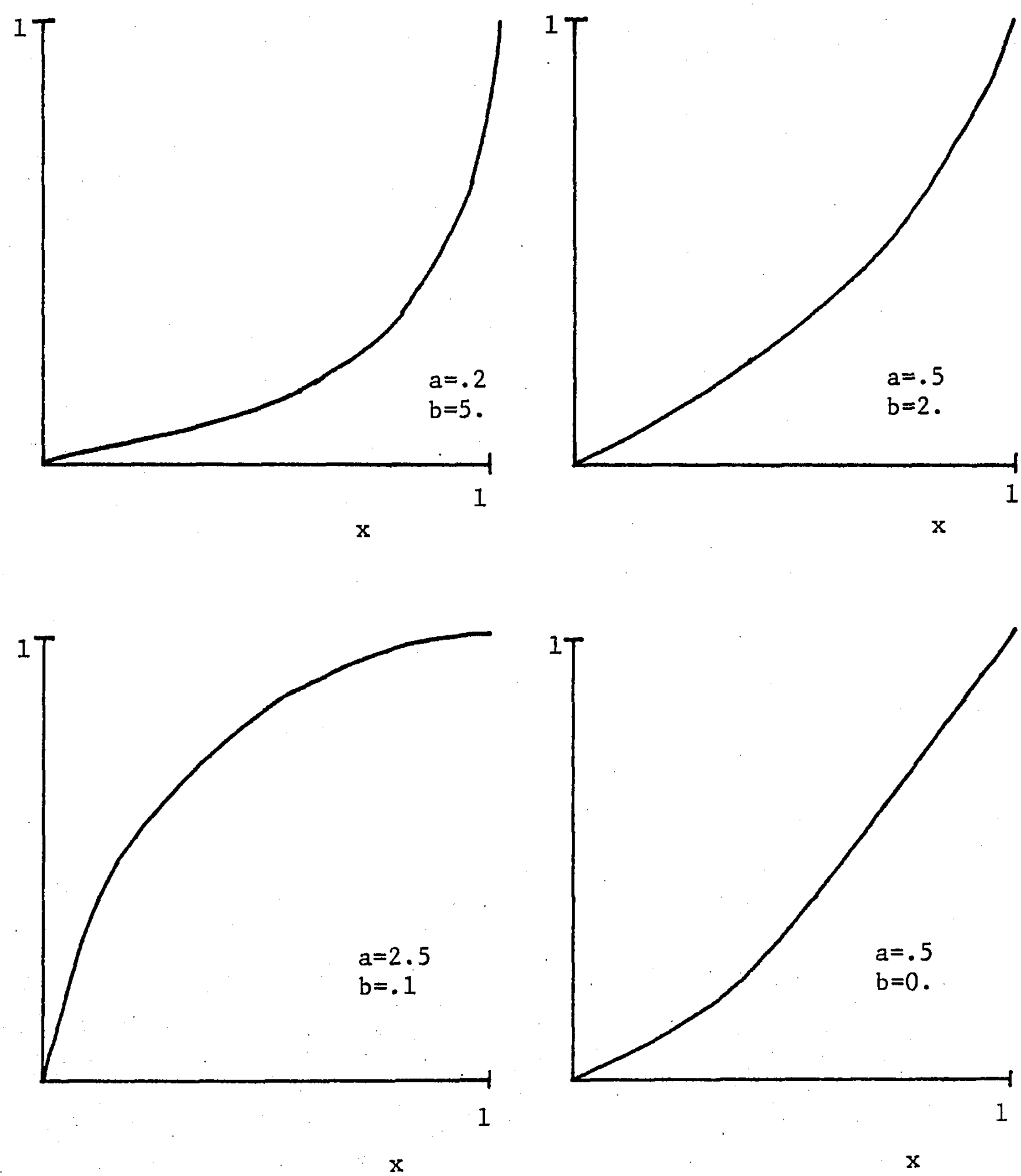


Figure 10. Examples of FORMO

3.2 Boundary Parameterization

Consider one boundary curve upon which grid points are to be distributed. The boundary curve has endpoints (x1,y1,z1) , (x2,y2,z2) and a functional form given by a discrete or continuous specification $y=y(x)$, $z=z(x)$ or $y_n=f_n(x_n)$, $z_n=g_n(x_n)$. One of the cartesian coordinates is always used as parameter. The curve is split, if necessary, such that it is monotonic in the independent (cartesian) variable. For example, the curve describing an airfoil section given by (x_n,y_n) , $n=1,m$ is split into an upper surface and lower surface, each of which has a monotonic variation in x ($x_n < x_{n+1}$ for all n). Parameters a,b,e,f are chosen to create the desired stretching function via FORM1. FORM1 is then sampled at equal intervals to produce a sequence of values in the range [0,1]. These values are regarded as the normalized x coordinates at which to place grid points on the curve. This method is straightforward, yet in a sense awkward because the arc length location of the grid points is not being directly determined by the stretching function. Since this discretized curve is a coordinate curve in the curvilinear system, it is the arc length location of the grid points that is important for clustering purposes.

Thus grid point clustering can not be controlled in a direct manner by the stretching functions. Astute clustering of the grid points can still be achieved, but it requires the program user to consider the shape of the curve he is parameterizing when choosing a stretching function.

3.3 Fuselage Specification

The fuselage is defined by a sequence of cross sections, each representing a constant x slice of this body. Such x stations are selected by a FORM1 stretching function applied to the overall length of the fuselage. Construction of the fuselage begins by specifying the upper crown line, lower crown line, and planform line as continuous functions of x, created by superpositions of bump and ramp functions. In each cross section these lines define three points. Two super elliptic arcs connect these three points, as shown in Figure 11. Exponents used to create these arcs are specified as functions of x, again formed by a superposition of ramps and bumps. The respective crown point and planform point define the axes of the upper and lower super ellipses. That is, each super elliptic arc is one fourth of the entire super ellipse. Both arcs then have a vertical tangent at the planform point, so they join together smoothly. Z is the independent variable used to generate points on these arcs. Particular values of z are chosen automatically to give adequate resolution of the high curvature regions of the arcs (see Figure 12).

This construction technique allows great latitude in the cross section shapes. For example, a smooth transition from circular to "boxed" lower cross sections can be achieved, as shown in Figure 13. Such a squared off section may be found in the wing root or landing gear area. Figure 14 presents some examples of super ellipses. It is worth mentioning that circles and normal ellipses are special cases of super ellipses.

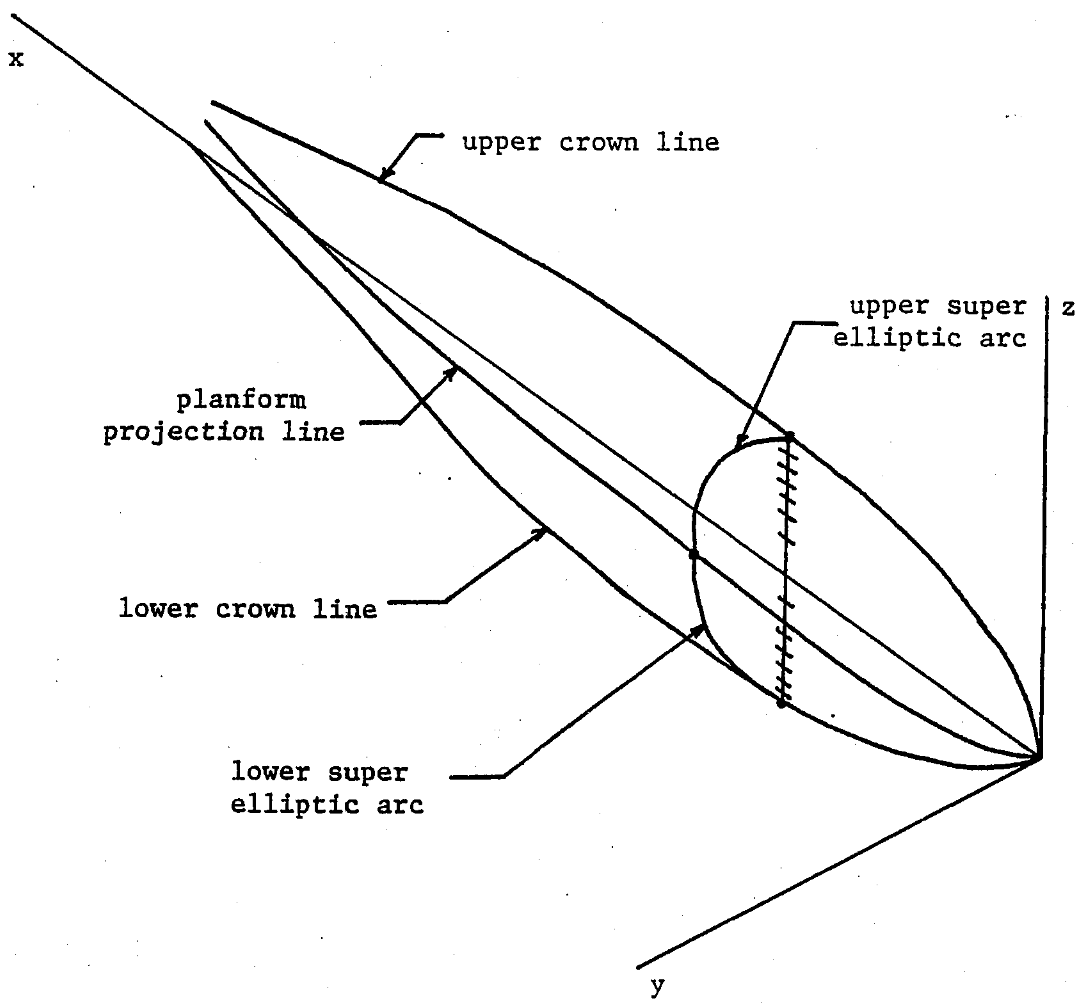


Figure 11. Initial Definition of Fuselage

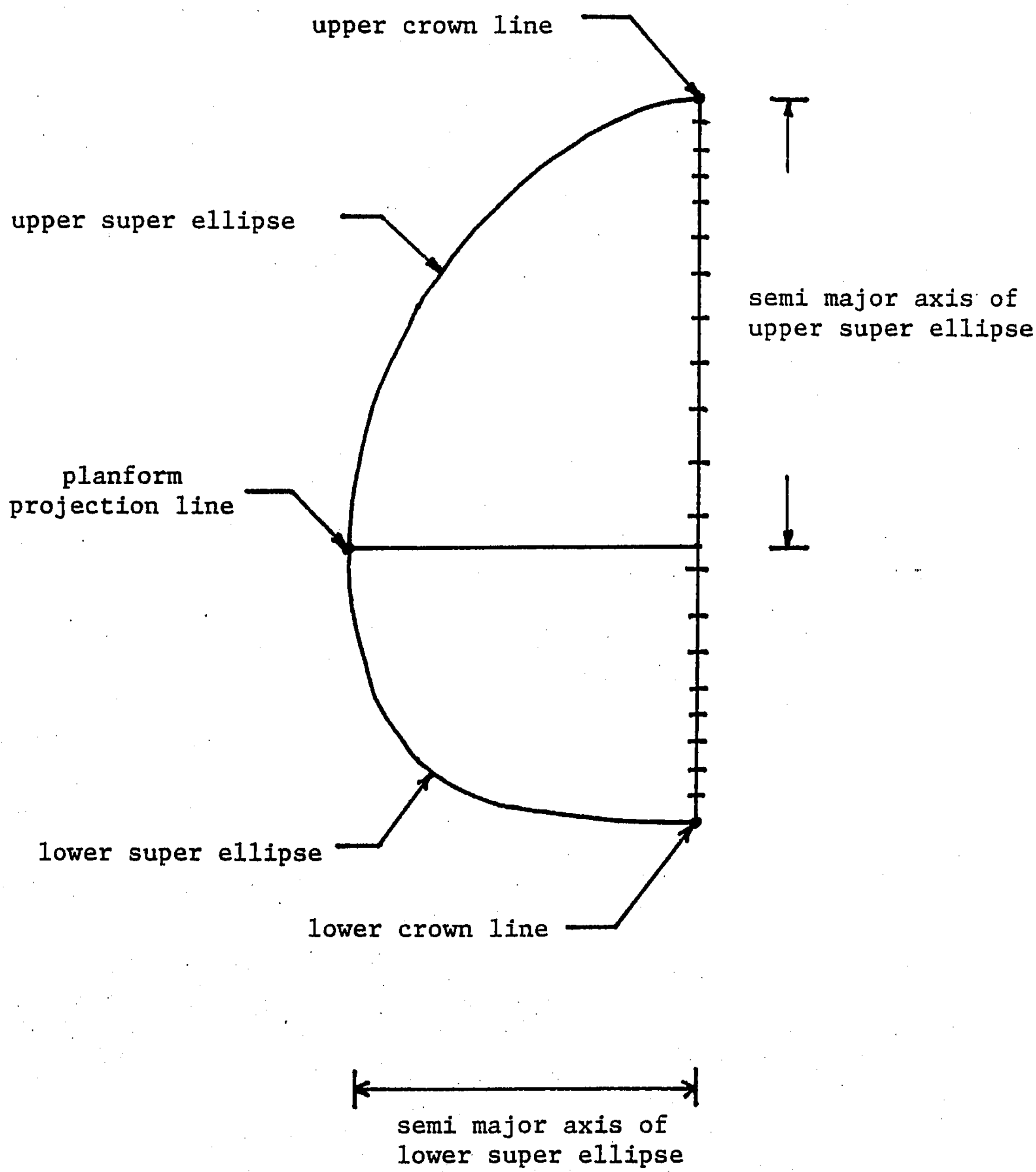


Figure 12. Fuselage Cross Section

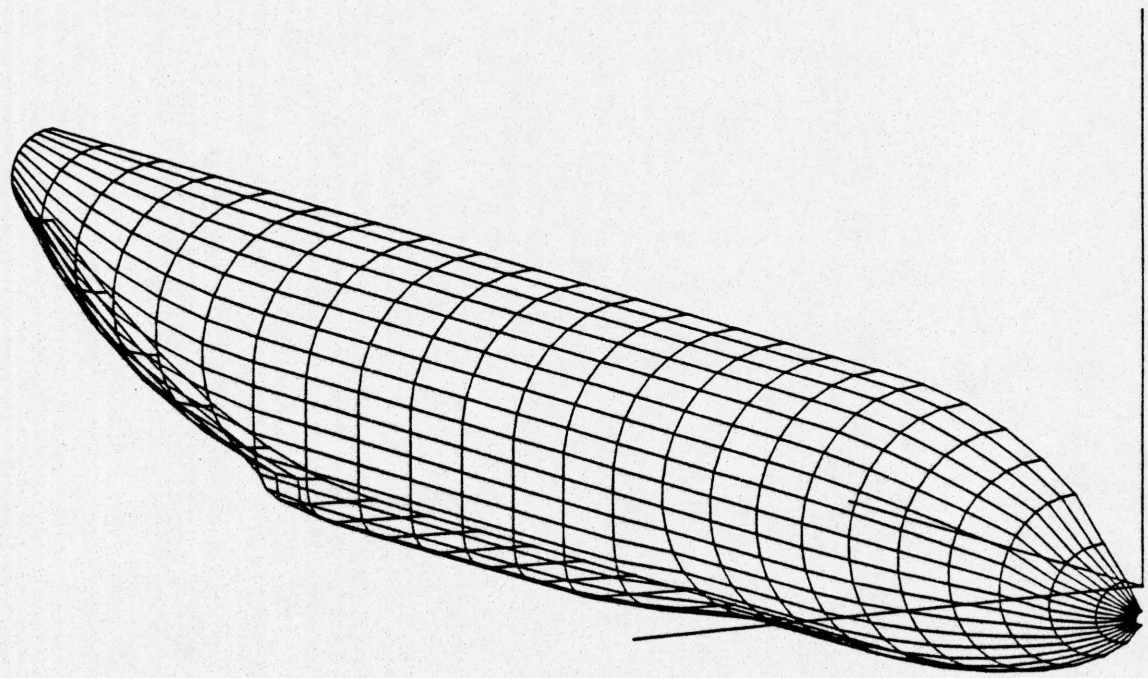
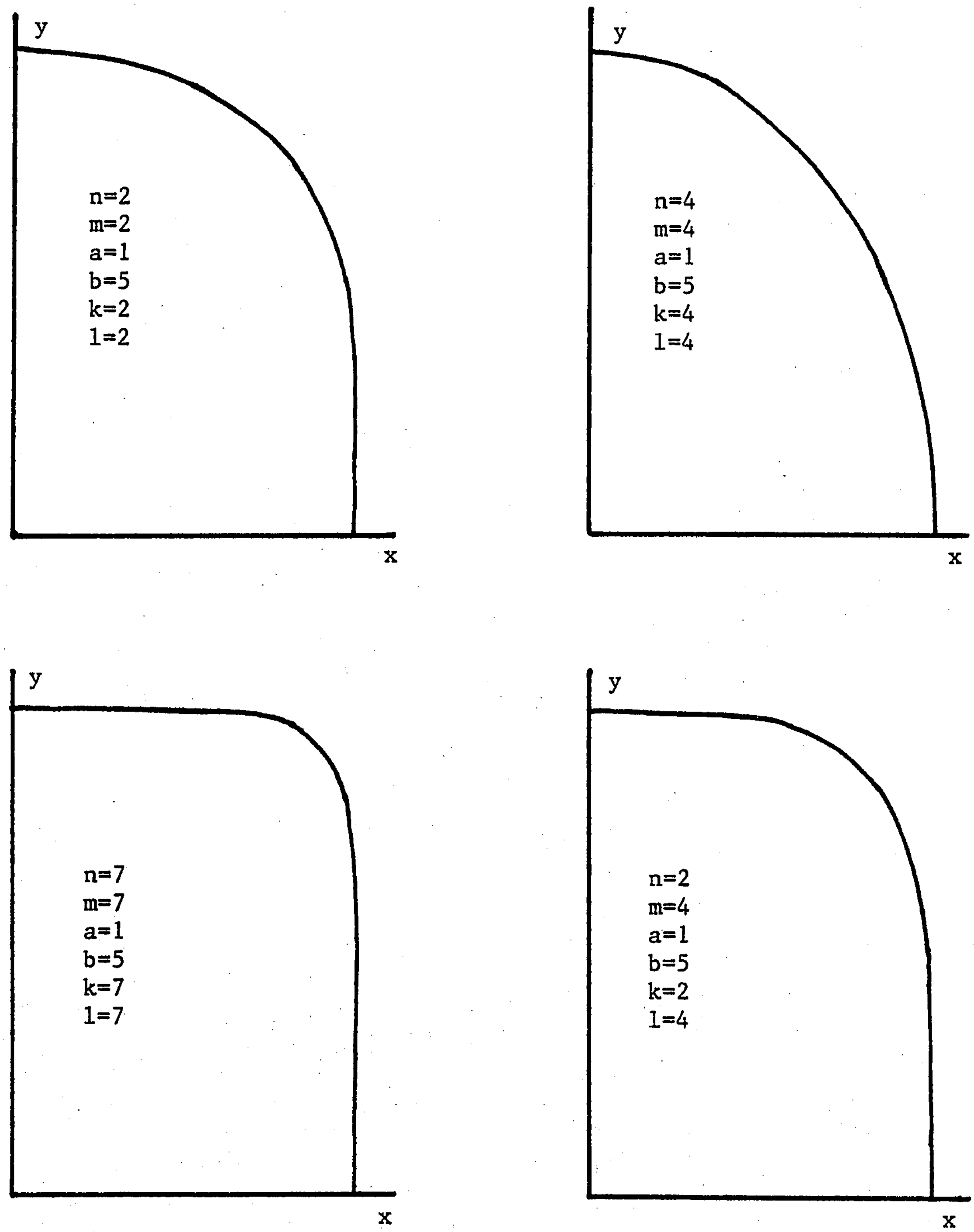


Figure 13. Transition from Circular to Boxed Lower Cross Sections



$$x^n/a^k + y^m/b^l = 1$$

Figure 14. Examples of Super Ellipses

The fuselage has thus been specified and a coordinate system has been established on its surface. One set of coordinate curves are the cross section arms. The second set are curves connecting corresponding points on successive cross sections.

3.4 Wing Construction

The wing is constructed from information about its planform, twist, dihedral, and section shapes. Planform leading and trailing edge curves, twist axis location, and twist function are all defined by a SHAPE function. The leading edge shape defines the sweep of the wing and the twist axis defines the dihedral. Figure 15 and Figure 16 illustrate these curves.

Three airfoil sections describe the wing shape at the root, mid, and tip span stations. These sections are blended together smoothly via a weighted interpolation to define section coordinates at any other span location. A number of such sections then represent the wing. These are suitably chosen so that rapidly changing wing geometry is accurately resolved by this discrete data.

Leading and trailing edge functions are specified in global coordinates. They define the x position of the wing on the body. The z position is given by the twist axis. That is, the leading edge point of the section coordinates is placed at the global point given by the leading edge x coordinate and twist axis z coordinate. The most inboard y coordinate of the twist axis defines the spanwise attachment point of the wing.

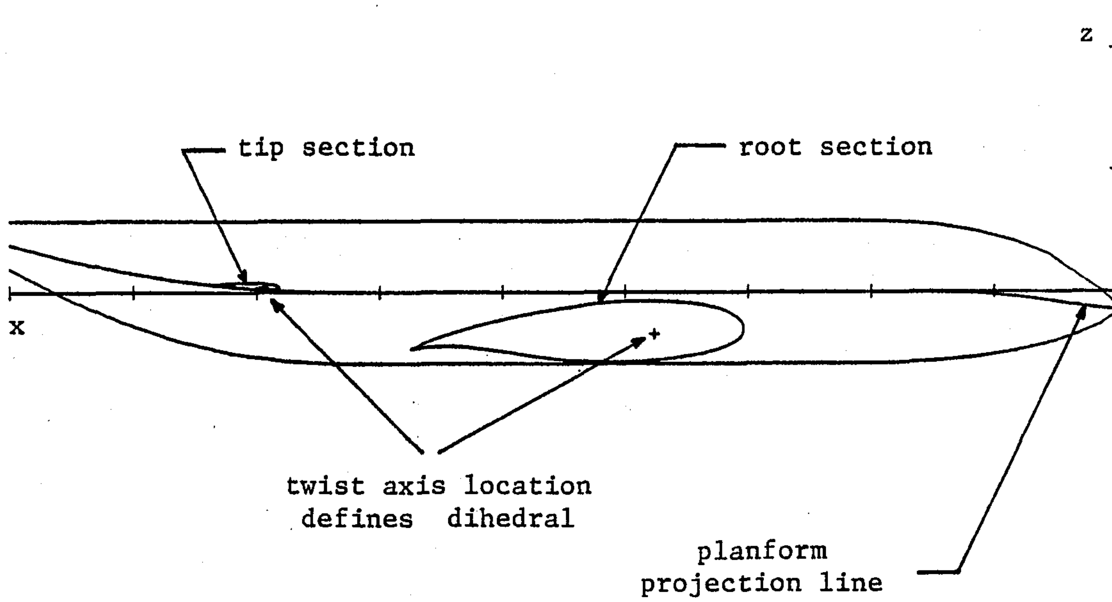


Figure 15. Wing Definition

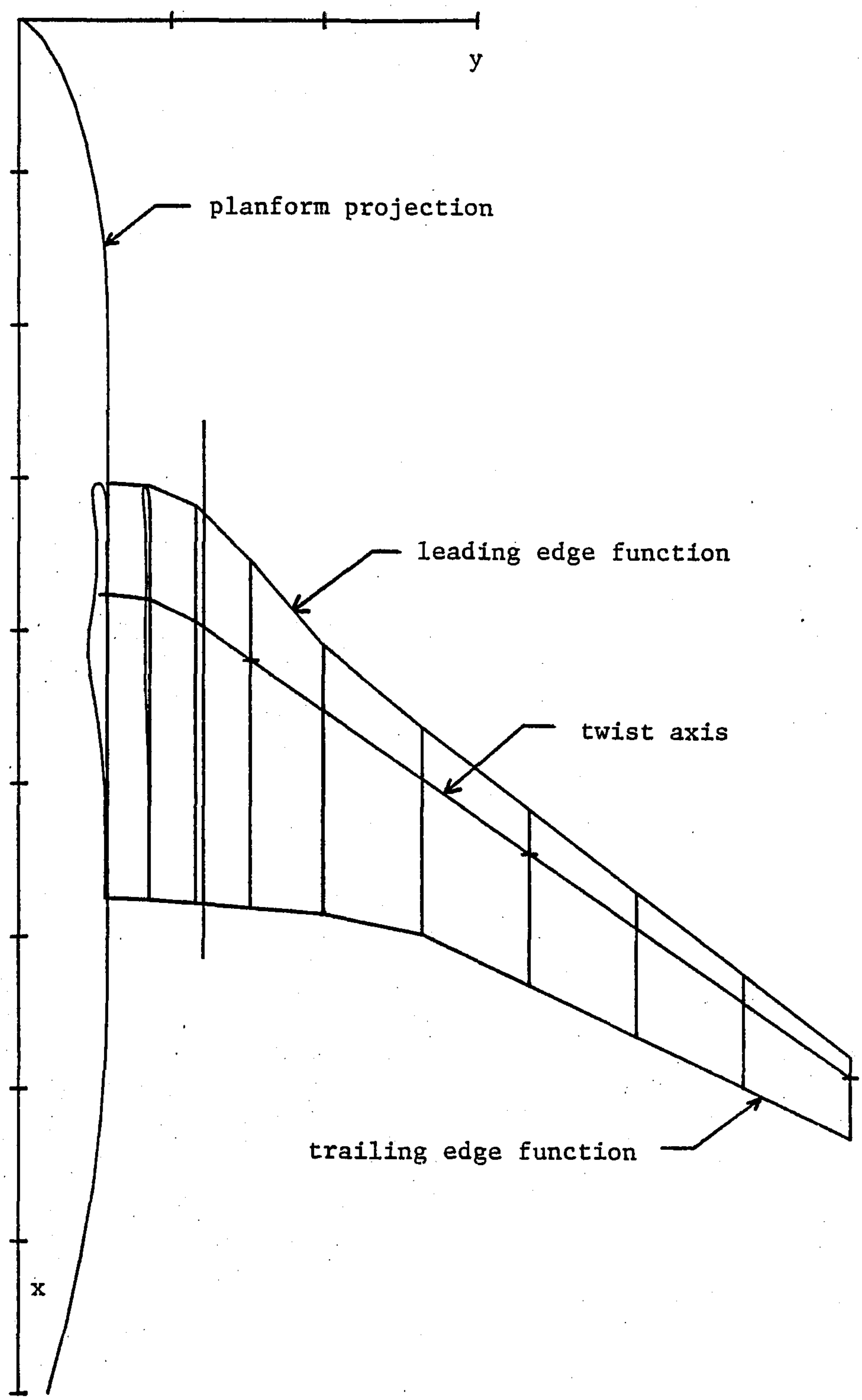


Figure 16. Wing Planform Definition

Airfoil section coordinates are normalized by the section chord length, so the actual thickness of a wing section is directly proportional to its chord length. In E88, section chord length is given by the difference between trailing and leading edge coordinates. Notice that "trumpet like" flair of the leading and trailing edges near the wing fuselage junction dramatically increase chord length, hence thickness, in this area. Thus the leading and trailing edge functions control both the growth in chord and thickness near the wing body junction.

Wing sections have been specified as curves in y = constant planes. In order to smoothly blend the surface geometry at the wing fuselage junction the wing root must conform to the curvature of the fuselage at the attachment point. To achieve this, the root section of the wing is projected onto the fuselage surface (see Figure 17). Other wing sections near the fuselage are projected toward its surface by weighted amounts. This projection process is cut off at a user defined span station. Within this region, the wing sections are three dimensional space curves, while outside this region the wing sections remain planar curves.

A surface coordinate system on the wing has been established by the above procedure. The wing sections can be thought of as coordinate curves with the stations (grid points) specifying the variation of the coordinate. Curves connecting corresponding points on successive sections also form coordinate curves, with the "section number" describing the variation of the coordinate along these curves. Figure 18 illustrates the coordinate systems on the wing and fuselage.

The wing and fuselage construction described above is suitable for generating wing fuselage surface coordinate points to serve as inputs to

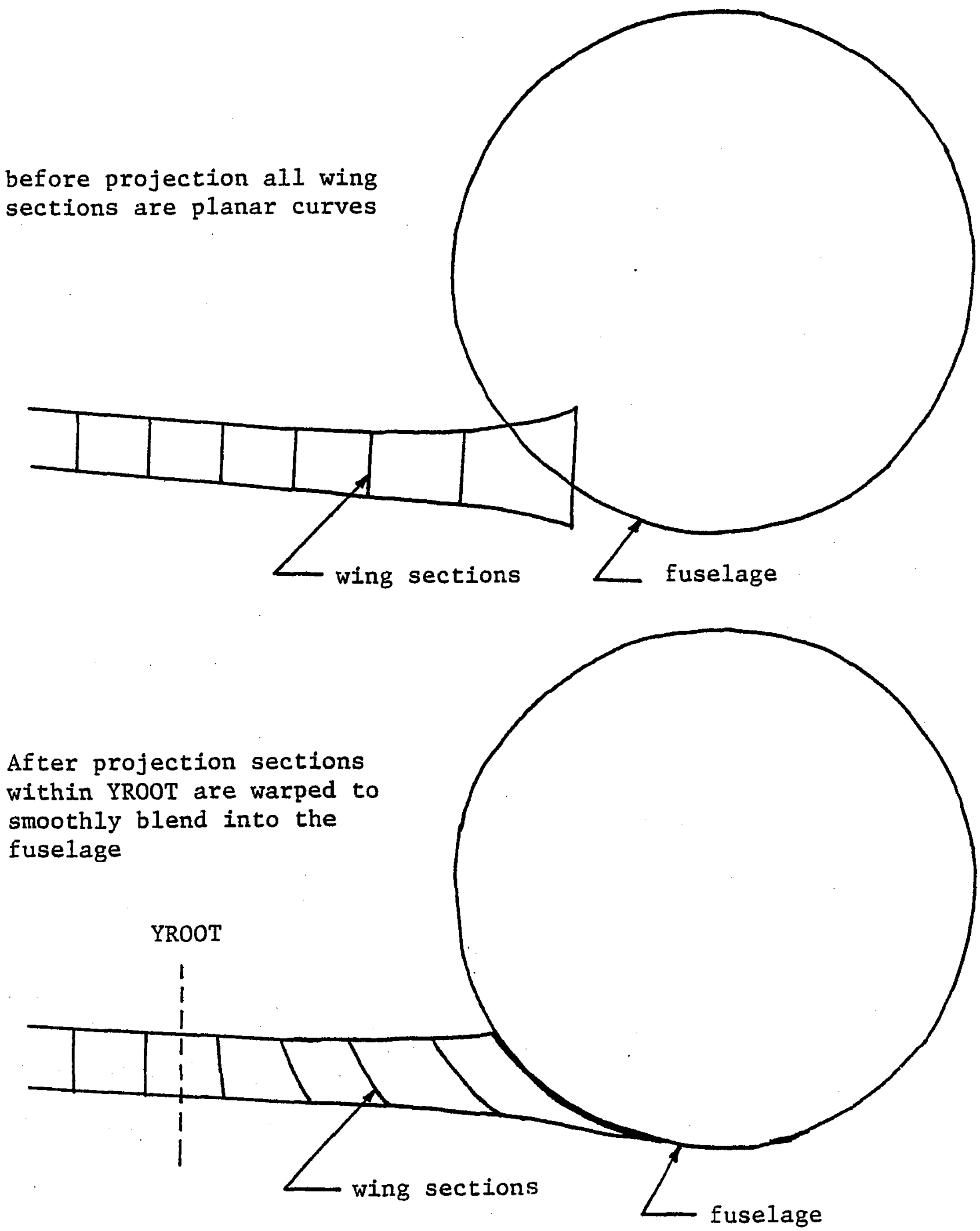


Figure 17. Projection of Wing onto Fuselage

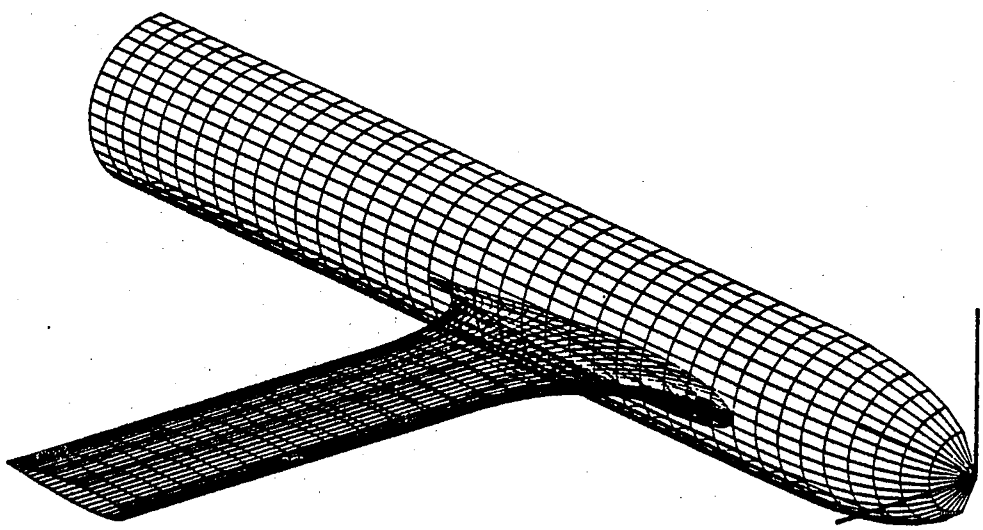


Figure 18. Growth of Wing Thickness Near Wing Fuselage Junction

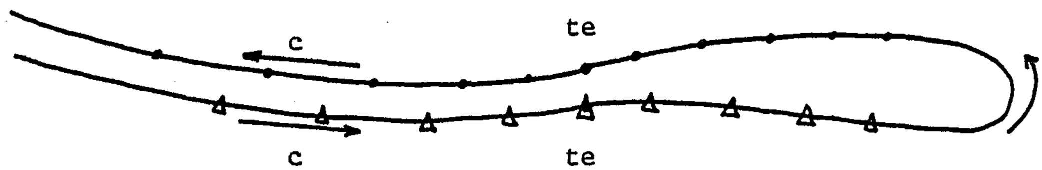
other codes which require such discrete data. Notice, however, that there has been no blending of the wing and fuselage coordinate systems into a single surface parameterization. Such a blending is carried out by E88, and is described in the following sections.

3.5 The Wake

A wing immersed in a real flow will introduce vorticity into the flow, via the action of viscosity and through lift production. This vorticity is shed from the wing trailing edge and is convected downstream, creating a contact discontinuity called the wake sheet. The wake sheet should be included in any model of the flow around a wing, and in models using the potential approximation, boundary conditions (for the potential) must be imposed on this surface. The true position of the wake is known only when the flowfield is known. However, because of this boundary condition requirement, codes which solve a potential equation usually guess a wake sheet position initially and then use this surface to divide the computational region. This cut may or may not be necessary to make the region simply connected. The guessed wake sheet remains fixed throughout the computation.

E88 creates a wake sheet by extending the airfoil upper and lower surface coordinates downstream of the trailing edge, forming a "tail" from each wing section. These upper and lower surface wake points coincide in physical space but are distinct insofar as the curvilinear coordinates are concerned, as is shown in Figure 19.

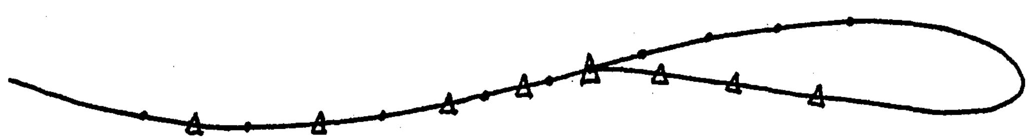
Wake curves emanating from sections close to the fuselage are projected toward the fuselage in a manner similar to that done for wing sec-



c coordinate line, made from
the wake and airfoil surface

• upper surface

Δ lower surface



the upper and lower wake curves
coincide in physical space

Figure 19. Upper and Lower Surface Wakes

tions. Curves for the rest of the sections remain at the constant span location of the section trailing edge. Wake curves leave the section at the section trailing edge angle and intercepts the "wake sheet trailing edge" at an angle computed for overall smoothness of the curves. The shape of the wake, including the wake trailing edge, is determined by the FORM1 function. Grid points are placed on these curves by using FORM1 as a stretching function.

3.6 New Parameterization of the Fuselage

The cross sections (of constant x) defining the fuselage are discarded. A new parameterization is introduced by considering the crown lines and wing root section (plus wake) contour to be boundaries of the fuselage surface (see Figure 8). The wing root section retains its previous parameterization. Grid points are distributed on the crown lines such that each point on the wing root section has a corresponding point on one of the crown lines. A number of intermediate coordinate lines are placed on the fuselage between the wing root and the crown lines via a weighted interpolation between these two curves.

Thus the wing and body are described by a sequence of sections (c curves) each of constant p coordinate and each having the same number of stations. In this method of surface parameterization, the fuselage is essentially treated as a series of large, distorted wing sections. Figure 20 shows that the entire wing fuselage is one coordinate surface (constant m). The section and station numbers of each point represent the computational coordinates p and c located on the surface.

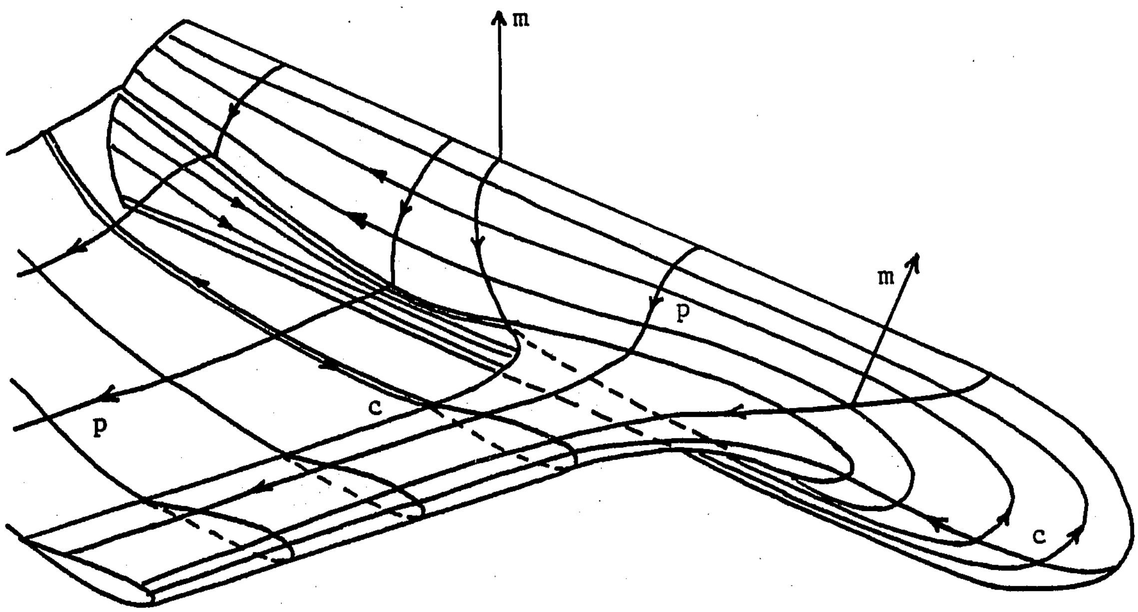


Figure 20. Coordinate System on Wing Fuselage Surface

3.7 Extension of Coordinates to Three Dimensions

The inner boundary of the computational region has now been parameterized and the same must be done for the outer boundary. Then the three dimensional grid can be constructed using these two surfaces.

3.7.1 Far Field Boundary

The outer boundary of the computational region is a half cylinder of super elliptic cross section, cut by the symmetry surface and bounded on each end of planar cross sections. The inflow boundary can be placed a specified distance in front of the fuselage nose, while the outflow plan is located at the rear of the fuselage. Displacing the outflow boundary downstream could be accomplished by adding a "sting" onto the fuselage.

3.7.2 Fill in the Computational Region

The three dimensional grid is composed of a number of two dimensional sheets. Each sheet is bounded by a c line in the wing fuselage surface and a rectangular curve located in the far field boundary surface (see Figure 21). Grid points are distributed on the far field curve by using a FORM1 stretching function. Sheets are constructed by connecting corresponding points on these two curves by straight lines. Each sheet is, therefore, a ruled surface. These straight lines are m coordinate curves. Grid points are distributed along these lines by using FORM1 function. Corresponding grid points on successive sheets define the p coordinate curves.

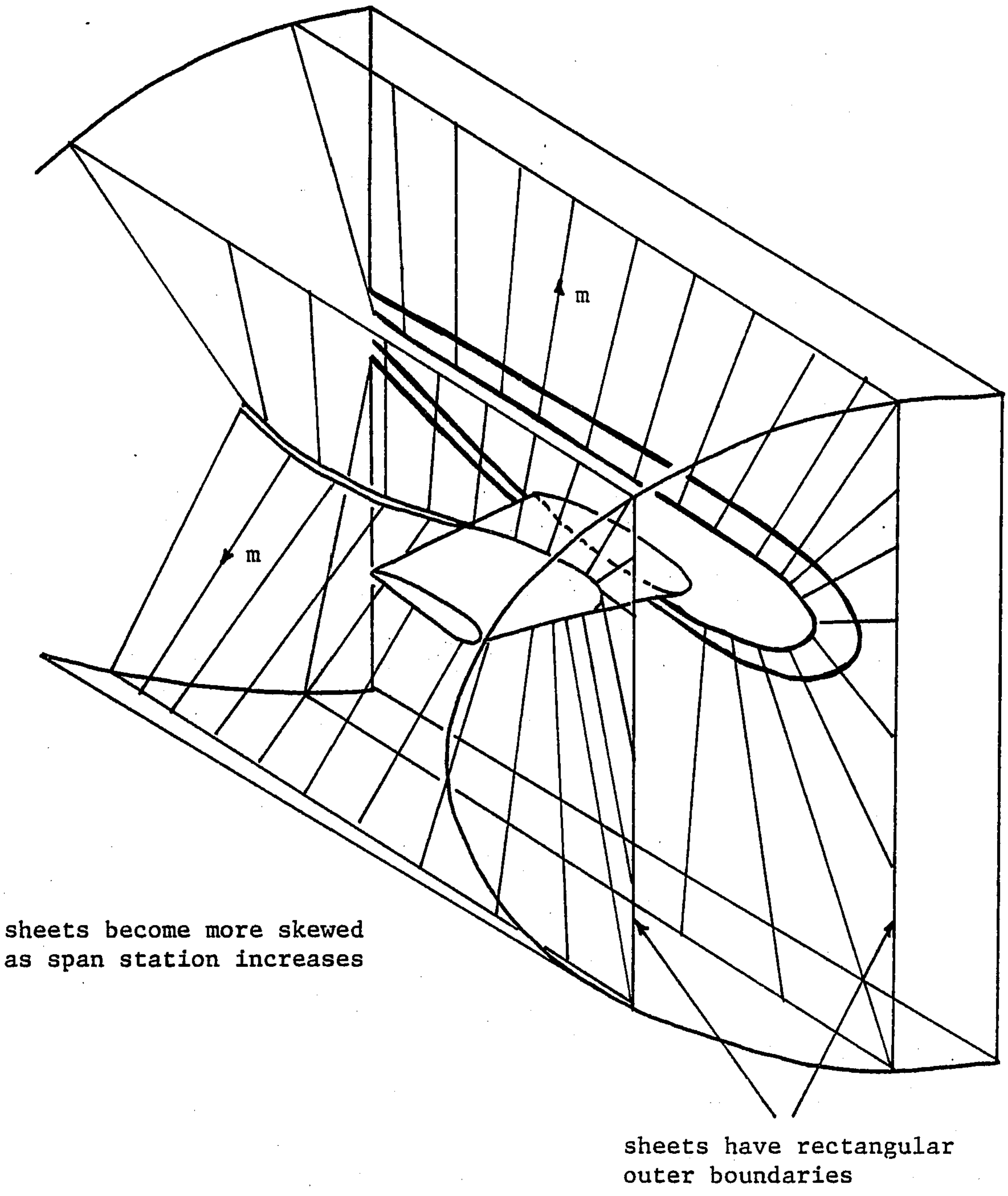


Figure 21. Three Dimensional Region Filled by Two Dimensional Sheets

3.7.3 Wingtip Extension

The grid sheets extending from the wing c curves to the outer boundary become progressively more "folded over" upon themselves as the span station increases. To complete the grid smoothly in the region outboard of the wingtip, an additional "false" wing section (of zero thickness) is created a specified distance beyond the wingtip (see Figure 22). The grid sheet emanating from this section is degenerate. Its "upper" and "lower" (z less than zero) parts coincide and it terminates on a line (collapsed rectangle) in the far field boundary, as shown in Figure 22.

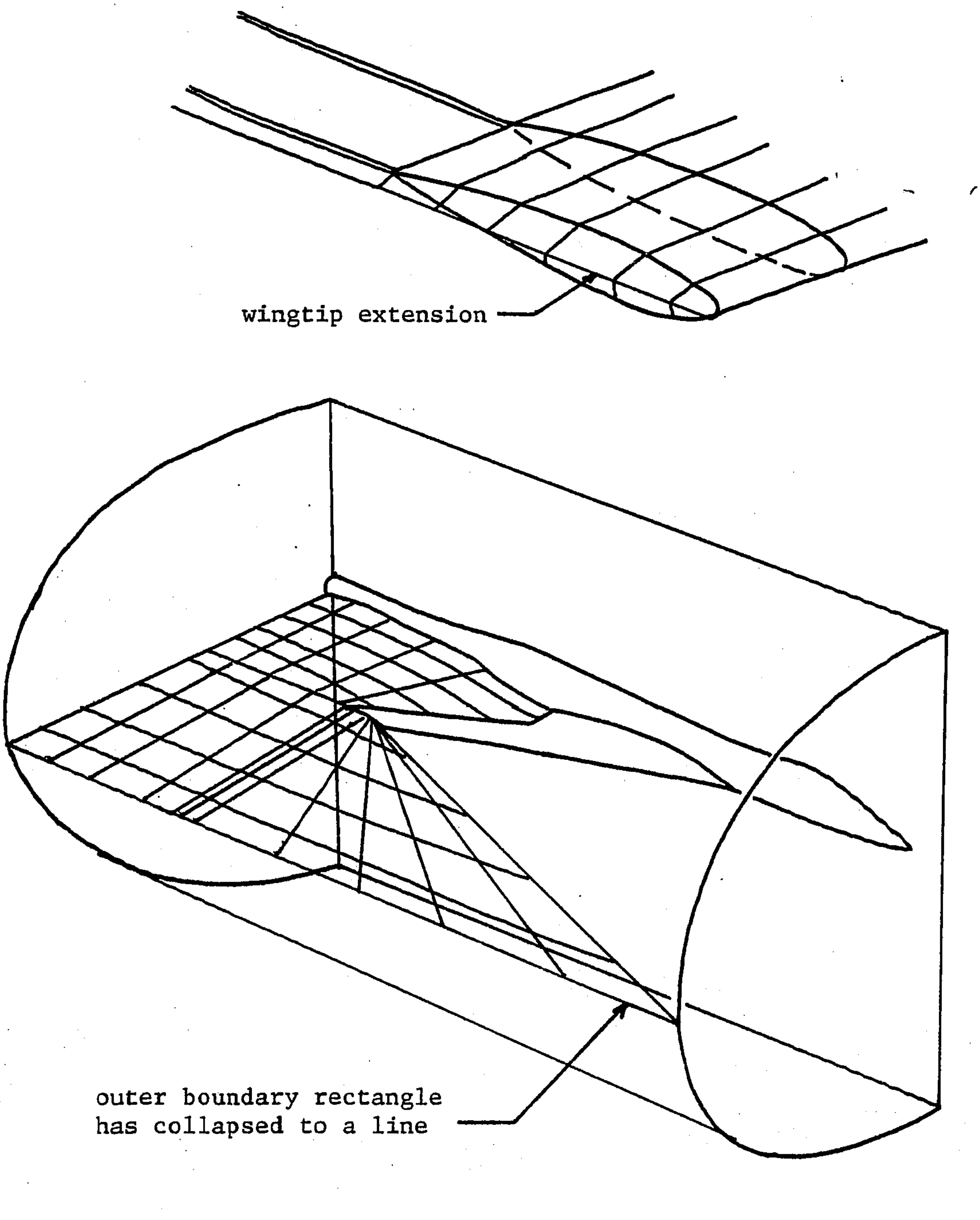


Figure 22. Wingtip Extension and Degenerate Grid Sheet

CHAPTER 4

RESULTS AND DISCUSSION

One capability of E88, generating wing body surface data, has been evaluated by creating two aircract configurations, a business jet and a large commercial transport. The surface data generated for these two designs was successfully used as input to the finite volume analysis code FLO30[3]. Recreating the known boundary curves of these aircraft was rather difficult to accomplish using the utility functions in E88, since no straightforward procedure exists for fitting these parametric curves to known points. This should not be viewed as a drawback of the code, since E88 was intended to create discrete points, rather than fit given ones. However, creating a desired curve shape via parametric manipulation in E88 is a difficult trial and error process. Choice of distribution functions for placing wing and fuselage sections and weighting the influence of specified wing sections were other problems encountered in generating these aircraft configurations.

The combined wing fuselage surface parameterization for each of the two designs is shown in Figure 23 and Figure 24. Notice the high concentration of grid lines on the wing and how they radiate outward from the wing root onto the fuselage. Spacing between these curves is controlled by the distribution of grid points on the wing root sections and fuselage crown lines. The variation in wake shape displayed between the two figures is primarily a result of specifying different wake trailing edge curves. Wing trailing edge angle and the wake curve emanating from the wing root section also influence the wake shape.

The symmetry plane sheet of the three dimensional grid is also shown in Figures 23 and 24. Clearly using straight lines as the m coordinate curves has introduced considerable distortion into some of the volume elements created by these grids. Such distortion becomes more severe in other sheets of the grid.

Coupling the wing thickness to its chord length in E88 should perhaps be modified to allow more control over the wing root fuselage junction. Presently all sculpturing in this area must be done by varying fuselage shape, with control of the wing root limited to specifying a single airfoil section. Allowing several input airfoil sections in the wing root area would provide a more versatile method of design.

Lack of a comprehensive three dimensional graphics display is a critical shortcoming of E88. The effects of minor variations in choice of parameters and auxiliary functions can not be easily observed in the existing display, hence the usefulness of these design tools is lost.

E88 parameterizes the wing fuselage combination as a single surface, rather than two intersecting ones. To investigate the suitability of this choice for three dimensional grid construction, the following experiment was performed.

A wing fuselage was created using E88. The "fuselage" consisted of large wing sections (see Figure 25). Surface coordinates of this body were then regarded as those of a wing alone, with an exaggerated

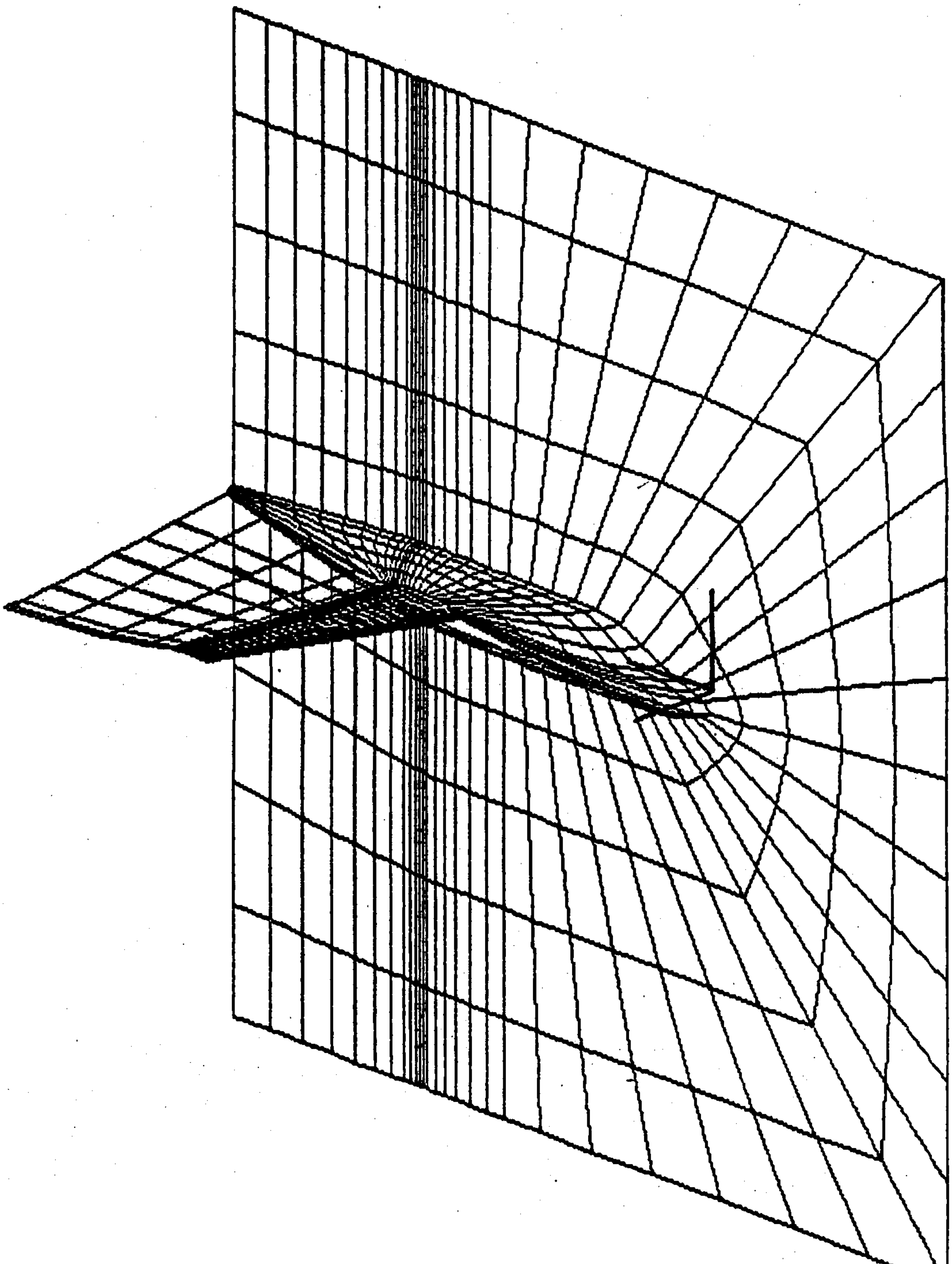


Figure 23. Surface Coordinates and Symmetry Plane Sheet for a Business Jet

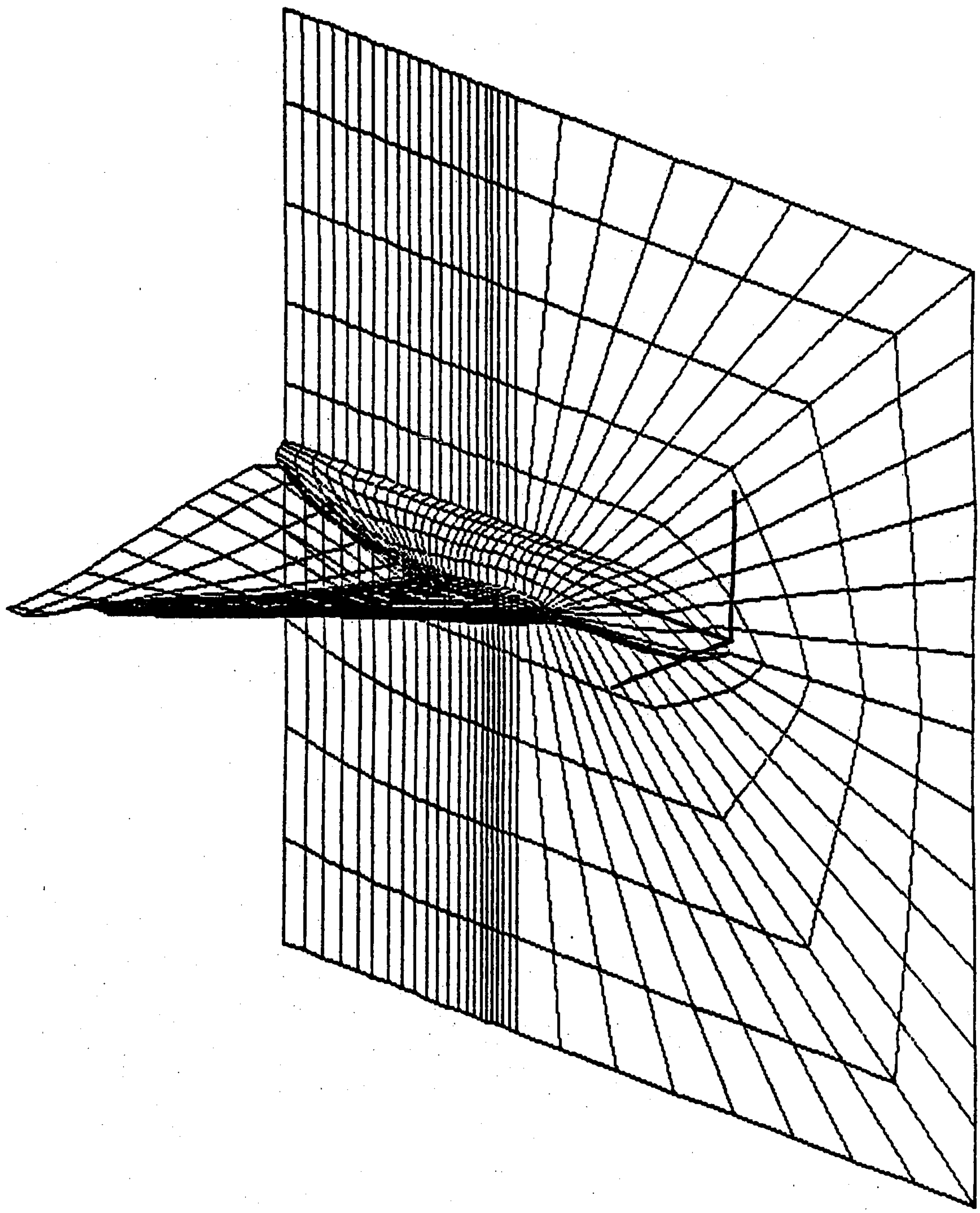


Figure 24. Surface Coordinates and Symmetry Plane Sheet for a Transport Aircraft

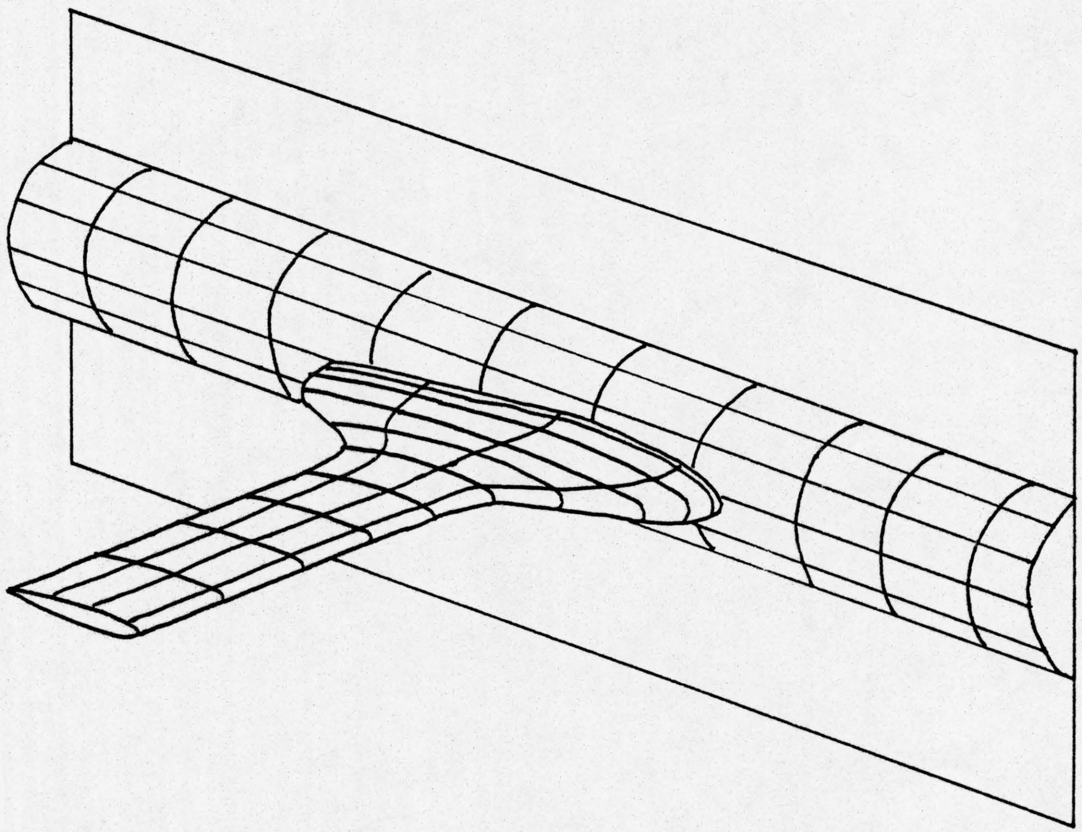


Figure 25. Wing Mounted on a Circular Cylinder

root section. This "wing" was mounted on a cylindrical fuselage and the resulting configuration was analyzed by FLO30. The grid generator in FLO30 constructed a grid about this distorted wing (parameterized as a single surface) and used the grid for its numerical calculations.

It was assumed that the cylindrical fuselage would not significantly influence the flow. Circular and super elliptic cross sections were tried for this fuselage, but super elliptic cross sections proved to be incompatable with the FLO30 grid generator, so circular cylinders were used exclusively. Cylinders of infinite length and tapered, finite length cylinders gave equivalent results.

The E88 body was characterized by the ratio, R, of its "fuselage" chord length to wing chord length. A successful grid was generated for $R \leqq 1.5$, but not for $R \geqq 2.75$. Realistic wing fuselage configurations, such as the one in Figure 15, have $R \geqq 3.7$. A successful FLO30 grid for values of $R \geqq 3.7$ would encourage further development of the E88 grid generation algorithm. However, failure to achieve this does not conclusively refute the usefulness of the single surface parameterization used by E88.

CHAPTER 5

CONCLUSIONS

E88 is a viable tool for creating wing fuselage surface data. Parametric specification of boundary curves facilitates exploring both wide and subtle variations in fuselage and wing planform shape. This surface data can be used with existing grid generators (of all types) and analysis codes for design optimization purposes.

Further study of the grid failures experienced with the distorted wing should be undertaken before the E88 grid in interfaced to a finite volume algorithm. The appearance of the grids generated in this experiment is not available and so the exact cause of failure us unknown.

Certainly E88's three dimensional grid could be made more flexible by replacing the straight m coordinate lines with curves. Additional discrete data would be required to generate the curves, probably in the form of one or more control points for each curve. Another possibility is smoothing the E88 mesh with an equipotential smoothing algorithm, such as those used with some finite element mesh generators[13].

The construction approach to surface specification and grid generation is much more suited to an interactive, rather than batch, environment. The user is then able to see the results of each input as the body is formed. E88 should be modified to run interactively, with "menu type" input commands and a three dimensional graphics display of hidden line suppression.

REFERENCES

1. Miranda, L. R., "A Perspective of Computational Aerodynamics from the Viewpoint of Airplane Design Applications," presented at the AIAA 20th Aerospace Sciences Meeting, January 11-14, 1982.

2. Jameson, A. and Caughey, D. A., "A Finite Volume Method for Transonic Potential Flow Calculations," Proceedings of AIAA 3rd Computational Fluid Dynamics Conference, Albuquerque, N. M., June 27-29, 1977, pp. 35-54.

3. Mercer, J. E., Jou, W. H., Jameson, A. and Caughey, D. A., Development of Finite Volume Methods for Three Dimensional Transonic Flows, Flow Research Report No. 166, 1980.

4. Gordon, W. J. and Hall, C. A., "Construction of Curvilinear Co-Ordinate Systems and Applications to Mesh Generation," International Journal for Numerical Methods in Engineering, Volume 7, 1973, pp. 461-477.

5. Cook, W. A., "Body Oriented (Natural) Co-Ordinates For Generating Three-Dimensional Meshes," International Journal for Numerical Methods in Engineering, Volume 8, 1974, pp. 27-43

6. Eiseman, P. R., Coordinate Generation With Precise Controls Over Mesh Properties, Institute for Computer Applications in Science and Engineering (ICASE), Report No. 80-30, November, 1980.

7. Thompson, J. F., Thames, F. C., and Mastin, C. W., "Automatic Generation of Body-Fitted Curvilinear Coordinate Systems for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," Journal of Computational Physics, Volume 15, 1974, pp. 299-319.

8. Haussling, H. J., "Boundary Fitted Coordinates for Accurate Numerical Solution of Multibody Flow Problems," Journal of Computational Physics, Volume 30, 1979, pp. 107-124.

9. Sorenson, R. L., A Computer Program to Generate Two Dimensional Grids About Airfoils and Other Shapes by the Use of Poisson's Equation, NASA Technical Memorandum 81198.

10. Yu, N. J., "Grid Generation and Transonic Flow Calculations for Three Dimensional Configurations," Presented at AIAA 13th Fluid and Plasma Dynamics Conference, July 14-16, 1980.

11. Sobieczky, H., Surface Generation for Aerodynamic Applications, Engineering Experiment Station, University of Arizona,Report TFD 81-03, June, 1981.

12. Eriksson, L., "Three Dimensional Spline Generated Coordinate Transformations for Grids Around Wing Body Configurations," NASA Conference Publication 2166, October 1980.

13. Thacker, W. C., "A Brief Review of Techniques for Generating Irregular Computational Grids," International Journal for Numerical Methods in Engineering, Volume 15, 1980, pp. 1335-1341.