UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188
Exp. Date: Jun 30, 1986

1a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

1b. RESTRICTIVE MARKINGS

2a. SECURITY CLASSIFICATION AUTHORITY

3. DISTRIBUTION/AVAILABILITY OF REPORT
Approved for public release; distribution unlimited.

2b. DECLASSIFICATION/DOWNGRADING SCHEDULE

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
HDL-TR-2140

5. MONITORING ORGANIZATION REPORT NUMBER(S)

6a. NAME OF PERFORMING ORGANIZATION
Harry Diamond Laboratories

6b. OFFICE SYMBOL (If applicable)
SLCHD-ST-AD

7a. NAME OF MONITORING ORGANIZATION

6c. ADDRESS (City, State, and ZIP Code)
2800 Powder Mill Road
Adelphi, MD 20783-1197

7b. ADDRESS (City, State, and ZIP Code)

8a. NAME OF FUNDING/SPONSORING ORGANIZATION
U.S. Army Laboratory Command

8b. OFFICE SYMBOL (If applicable)
AMSLC

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c. ADDRESS (City, State, and ZIP Code)
2800 Powder Mill Road
Adelphi, MD 20783-1145

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
P611102.H4400	AH44		

11. TITLE (Include Security Classification)
An Analytic Model for Aerodynamic Shapes

12. PERSONAL AUTHOR(S)
Michael J. Vrabel

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Final	FROM Oct 86 TO Oct 87	March 1988	62

16. SUPPLEMENTARY NOTATION
HDL project: AE1716; AMS code: 611102.H4400

17. COSATI CODES		
FIELD	GROUP	SUB-GROUP
01	01	
12	01	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
Computer graphics, lofted surface, superquadric, analytic surface model, surface modeling, topological transformation

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

Exact analytical models represent a viable alternative for applications requiring geometric constructs with a level of surface perfection unachievable with traditional polynomial techniques. To avoid the limitations in shape complexity normally associated with traditional analytic models, a model based upon a topological transformation of a lofted surface was developed. The consequence is, if not an equation of all shapes, then, at least, an equation describing a great many arbitrarily complicated shapes. A full theoretical development of the model is presented along with a model-implementing FORTRAN source module and some example geometries.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION
UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL
Michael J. Vrabel

22b. TELEPHONE (Include Area Code)
(202) 394-3140

22c. OFFICE SYMBOL
SLCHD-ST-AD

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

HDL-TR-2140
March 1988

An Analytic Model for Aerodynamic Shapes

by Michael J. Vrabel

U.S. Army Laboratory Command
Harry Diamond Laboratories
Adelphi, MD 20783-1197

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturers' or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Preface

There are 14 figures within this report, many with a level of complexity demanding a full-page presentation. To avoid the disruption to the text that the interspersing of multiple full-page illustrations causes, all figures are grouped together just before the appendices. It is hoped that the reader does not find this to be an inconvenience.

Again, with regard to the figures—all surface representations were produced by using the analytic surface model code to generate an array of surface-defining points. These points were joined by the plotting routines, for constant values of the parametric variables, with straight line segments. Any perceived surface blemishes are attributable to the plotting routines and the simple fashion in which they were implemented by the author.

It is regretted that the press of circumstances has not permitted the development of any truly complicated surfaces. The relative simplicity of the modeled shapes should not be taken to imply any undue limitations on the capacity of the model.

Michael J. Vrabel

There are more ways than one to skin a cat.

R. A. Habas, Morals for Moderns

Contents

List of Figures

1 Analytic Surface Model – Theory

1.1 Introduction

The ability to generate high-fidelity surface models is fundamental to the solution of many classes of problems in three dimensions. The traditional approach to the modeling of a complex surface is based on a polynomial fit to an array of surface-defining points. This approach can suffer from significant shortcomings. In particular, depending on the characteristics of the polynomial fitting algorithm, the exact shape of the resultant surface can be difficult to predict and control. More specifically, a polynomial-generated surface can result in a level of surface irregularity that, while not always visually apparent, is nevertheless readily detectable when a surface characteristic such as curvature is calculated.

For shapes possessing exact analytic descriptions, these shortcomings do not exist. Exact analytic models of structures are generally viewed as restricted to objects of very simple shape. To be meaningful, such an approach must extend to arbitrarily complex shapes. This report develops a technique for the piece-wise analytic modeling of arbitrary surfaces. Succinctly, the approach is to design a surface based on a lofting procedure and then perform a topological transformation on the lofted model. Lofting refers to a procedure for creating a structure by "stretching" a surface over two or more parallel structural cross sections. Two seminal papers[1,2] provided the stimulus for the development of this model—the first by developing the basis for an analytic lofted surface, and the second by demonstrating the flexibility of the superquadrics for shape modeling. It should be made clear from the onset that, while the title of this report indicates a model restricted to aerodynamic shapes, the model is general and can be used for a wide range of shapes. It is for aerodynamic shapes with their smooth (and smooth is used here in its colloquial sense) transitioning surfaces that the model is most easily adapted. Included within the report is the full theoretical framework,

[1] R. L. Barger and M. S. Adams, "Semianalytic Modeling of Aerodynamic Shapes," National Aeronautics and Space Administration, NASA TP-2413, 1985.

[2] A. H. Barr, "Superquadrics and Angle-Preserving Transformations," IEEE CG&A, Vol. 1, No. 1, Jan. 1981.

a users guide for a FORTRAN implementation of the model, a source code listing, and numerous examples.

1.2 The Barger Model

The Barger and Adams paper describes a lofting technique for the analytic modeling of classes of smoothly varying surfaces. A surface is defined as a sequence of subsurfaces formed by the transitioning between bounding cross-sectional shapes. The equation for a subsurface in Cartesian coordinates is

$$y(x,t) = S(x)\{[1 - T(x)]y_1(t) + T(x)y_2(t)\} \tag{1}$$

$$z(x,t) = S(x)\{[1 - T(x)]z_1(t) + T(x)z_2(t)\} \tag{2}$$

where $y_1(t)$, $z_1(t)$, $y_2(t)$, and $z_2(t)$ are analytic functions in parametric form of the bounding cross sections, $T(x)$ is the transition function governing the shape of the subsurface and varying from 1 to 0 as x varies from 0 to 1, $S(x)$ is the size function providing an independent function scaling, t is a parametric variable, and x is the normalized (over a subsurface) longitudinal coordinate. The condition imposed on the transition function at the interface between two subsurfaces (A and B) where continuity is required is

$$T_A^N(1) = T_B^N(0) = 0 \tag{3}$$

where N defines the order of the derivative for which continuity is to be enforced. The further conditions on $T(x)$ are that it be single-valued and monotonic. Since a surface describable by an arbitrary transition function can be divided into elements for which monotonicity can be enforced, this latter condition poses no undue surface modeling limitation.

Although equations (1) and (2) indicate that the size function $S(x)$ operates over a single subsurface, where multiple continuous elements are defined, $S(x)$ will operate over each continuous subset. In this way, interelement continuity is not affected. The conditions on $S(x)$ are that it be smooth, single-valued, and non-negative.

1.3 Analytic Term Definition

The previous section provides a framework for defining a class of analytic surfaces. It is appropriate at this point to both generalize terms within the lofted surface equations and change notation, substituting for t and x the parametric variables u and v. This latter change permits the three components of the surface-defining vector to be expressed as dependent variables:

$$y_l(u,v) = S_y(v)\{[1 - T_y(v)]y_1(u) + T_y(v)y_2(u)\} \tag{4}$$

$$z_l(u,v) = S_z(v)\{[1 - T_z(v)]z_1(u) + T_z(v)z_2(u)\} \tag{5}$$

$$x_l(v) = v \tag{6}$$

The reasons for these changes will become apparent.

To be a viable model, for the generalized transition, size, and cross-section functions, expressions must be substituted that are a function of and can be readily correlated with geometric shape. Rather than requiring the user to define unique expressions for each geometric construct, this section intends to demonstrate a set of equations general enough to be used without modification for defining a broad range of shapes.

1.3.1 Transition Function

A superellipse is defined as

$$y = A_1 \sin^{\epsilon} u \tag{7}$$

$$z = A_2 \cos^{\epsilon} u \tag{8}$$

where $0 \leq u \leq 2\pi$ and $\epsilon > 0$. To generate a four quadrant plot (see fig. 1), these equations must be modified:

$$y = A_1 \sin u \mid \sin u \mid^{\epsilon-1} \tag{9}$$

$$z = A_2 \cos u \mid \cos u \mid^{\epsilon-1} \tag{10}$$

To preserve the functional simplicity of the original definition and allow its use in all subsequent operations, the following rule is adopted: the sign of the sin and cos terms is to be taken out from under the exponent. This rule makes equations (7) and (9) and equations (8) and (10) equivalent and can be carried through to define a proper set of function derivatives.

Equations (7) and (8) can be combined:

$$y = A_1 \left[1 - (z/A_2)^{2/\epsilon}\right]^{\epsilon/2} \tag{11}$$

If for ϵ one substitutes

$$\epsilon = a_1 + a_2 z^{a_3} + a_4 z^{a_5} \tag{12}$$

where $a_{1,\ldots,5}$ are constants, then equation 11 can be transformed into an equation that can be used as a transition function:

$$T(v) = \left(1 - v^{Q}\right)^{1/Q} \tag{13}$$

where

$$Q = 2/\left(a_1 + a_2 v^{a_3} + a_4 v^{a_5}\right) \tag{14}$$

An alternate form for Q is

$$Q = 2/\left[a_1 + a_2 v^{a_3} + a_4 \sin^{a_5}(\pi v)\right] \tag{15}$$

This expression generates a large family of monotonic curves meeting the requirements that $T(0) = 1$ and $T(1) = 0$, and the conditions imposed by equation (3) to, at least, the second derivative level at both $v = 0$ and $v = 1$. This is demonstrated, for a select sample, in figures 2 through 4. The transition functions demonstrated in these figures are intended to meet the indicated boundary value conditions. Where less rigorous conditions are to be met, quite different curve shapes are possible. To meet first-derivative continuity conditions the limits on ϵ are $\epsilon \leq 1.4$ about $v = 0$ and $\epsilon \geq 2.9$ about $v = 1$. To meet second-derivative conditions, $\epsilon \leq 0.8$ about $v = 0$ and $\epsilon \geq 4.9$ about $v = 1$. There is little loss in the generality of the lofted surface model if equation (13) is made the transition function. The Barger model includes the same transition function for the y and z components of the vector defining the lofted surface. A useful generalization is to allow the transition function to be different for y and z. This is indicated in equations (4) and (5) and is included in the codes listed in appendix A and the final equation summary.

1.3.2 Cross Section

Figure 1 demonstrates the diversity of shapes possible with the simple function defining the superellipse. This poses the possibility that some variation on the superellipse may prove useful in defining a generalized equation for modeling cross sections. Central to this task is the recognition that the relationship between the parametric variable and the y-z location of the function is well defined for four values of u independent of ϵ:

$$\begin{aligned} u &= 0 && : +\text{zaxis} \\ u &= \pi/2 && : +\text{yaxis} \\ u &= \pi && : -\text{zaxis} \\ u &= 3\pi/2 && : -\text{yaxis} \end{aligned}$$

Consequently, a more general (and more useful) expression can be developed:

$$y = A_1 \sin^{\epsilon}\left(mu + u_y\right) \tag{16}$$

$$z = A_2 \cos^{\epsilon}\left(nu + u_z\right) \tag{17}$$

where m, n, u_y, and u_z are constants that can be redefined whenever the following relationships are satisfied:

$$mu + u_y = \pi k_m/2$$

$$nu + u_z = \pi k_n/2$$

for $k_m, k_n = 0, 1, 2, \ldots$. The resultant function will be, at least, analytic in each interval. Further flexibility results from making ϵ a function of the parametric variable:

$$\epsilon = b_1 + b_2 q^{b_3} \tag{18}$$

$$q = \mid \sin(ju + u_q) \mid$$

where the constants associated with the argument of q are governed as in equations (16) and (17).

These equations define curves with fixed locations with respect to the coordinate system origin and axes. To allow for rotation and translation in a plane, an affine transformation is performed. The resultant cross-section equations are

$$y = A_3 \left\{ \left[A_1 \sin^{\epsilon}(mu + u_y) + y_{t_1} \right] \cos\psi + \left[A_2 \cos^{\epsilon}(nu + u_z) + z_{t_1} \right] \sin\psi \right\} + y_{t_2} \tag{19}$$

$$z = A_4 \left\{ - \left[A_1 \sin^{\epsilon}(mu + u_y) + y_{t_1} \right] \sin\psi + \left[A_2 \cos^{\epsilon}(nu + u_z) + z_{t_1} \right] \cos\psi \right\} + z_{t_2} \tag{20}$$

where y_{t_1}, y_{t_2}, z_{t_1}, and z_{t_2} define the components of two independent displacement vectors and ψ defines the angle of rotation about the origin. The translation and scaling redundancies included in these equations, by allowing the corresponding operations both before and after a rotation, facilitate model use.

The rotation and translation terms, combined with the parameter redefinition capability, permit a cross-section model fragmented into an unlimited number of open and closed curves. Such a capability, combined with the lofted surface model, allows the definition of multibranched bodies.

1.3.3 Size Function

The size function is intended to provide an independent scaling of the lofted surface. Although the size function is, in a sense, redundant, its presence provides a measure of model flexibility difficult to duplicate with just the cross-section and transition functions. Two useful forms for the size function are

$$S(v) = \sum_{n=1}^{m} c_{1n} v^{c_{2n}} \tag{21}$$

and an alternative,

$$S(v) = \left[\sum_{n=1}^{m} (c_{1n}v + c_{2n})^{c_{3n}} \right]^{c_4} \tag{22}$$

where m is generally less than or equal to 4, $0 \leq v \leq 1$, and $c_{1n,\ldots,3n}, c_4$ can take on any value consistent with the restrictions imposed upon $S(v)$.

As with the transition function, allowing for independent size functions for the y and z surface-defining vector components will prove useful.

1.4 Topological Transformation

The strength of a lofted model, namely its ability to define a shape through the specification of a limited number of cross sections, is also its weakness. At the model's ends, that is at $v = 0$ and $v = 1$, shape definition is constrained by the requirement that the end-defining cross sections be planar and normal to the model axis (x axis). This limitation can be circumvented by performing a topological transform on the lofted model. This transformation is to operate upon the x component of the surface-defining vector, substituting for the simple expression of equation (6) one that is a function of both u and v. Properly implemented, such a transformation allows complete freedom in specifying an out-of-plane shape for any model cross section.

Because of the success of the lofted model, the topological transformation will employ similar equations. For equation (6), one substitutes

$$x(u,v) = x_l(v) + x_t(u,v) \tag{23}$$

where $x_t(u,v)$ is yet to be defined. Equations (4) and (5) are left unchanged:

$$y(u,v) = y_l(u,v) \tag{24}$$

$$z(u,v) = z_l(u,v) \tag{25}$$

Analogous to the basic lofted-model equation, $x_t(u,v)$ is defined as

$$x_t(u,v) = \left(1 - x_{ln}^h(v)\right) G_f(u,v) + x_{ln}^h(v) G_b(u,v) \tag{26}$$

where $x_{ln}(v)$ is $x_l(v)$ normalized over an appropriate interval (either the interval between adjacent cross sections or the length of the modeled structure) and h is a constant that defines the rate at which $x_t(u,v)$ transitions between the two terms of this equation. The terms $G_f(u,v)$ and $G_b(u,v)$ are defined as

$$G_f(u,v) = h_{f1}\left[1 - \mid h_{f2} - y_{fl} \mid^{2/\epsilon_f}\right]^{\epsilon_f/2} + h_{f3}\left[1 - \mid h_{f4} - z_{fl} \mid^{2/\epsilon_f}\right]^{\epsilon_f/2} + h_{f10} \tag{27}$$

$$G_b(u,v) = h_{b1}\left[1- \mid h_{b2} - y_{bl} \mid^{2/\epsilon_b}\right]^{\epsilon_b/2} + h_{b3}\left[1- \mid h_{b4} - z_{bl} \mid^{2/\epsilon_b}\right]^{\epsilon_b/2} + h_{b10} \tag{28}$$

$$0 \leq y_{f,bl} \leq 1 \tag{29}$$

$$0 \leq z_{f,bl} \leq 1 \tag{30}$$

with

$$\epsilon_f = h_{f5} + h_{f6} y_{fl}^{h_{f7}} + h_{f8} z_{fl}^{h_{f9}} \tag{31}$$

$$\epsilon_b = h_{b5} + h_{b6} y_{bl}^{h_{b7}} + h_{b8} z_{bl}^{h_{b9}} \tag{32}$$

where $y_{f,bl}$ and $z_{f,bl}$ are the normalized y and z coordinates of cross-sections f and b—front and back—and $h_{f,b1...,10}$ are constants. The similarity between equations (26) through (32) and equations (4), (5), (12), and (13) should be noted.

To increase the flexibility of equation (26), parameter switching can be included. Since section 1.3.2 permits the definition of each cross section in terms of a composite of discrete curve segments, equation (26) can employ parameter switching for each discrete segment or for clusters of these segments.

1.5 Three-Dimensional Model

Equations (23), (24), and (25) are the components of a surface-defining vector. This vector contains 69 user-defined constants. With parameter switching, the total number of constants is effectively limited only by the structure of the model-implementing computer code. This number presently exceeds 15,000. The obvious implication is that the complexity of the surface capable of being modeled is limited by the user's capacity to define the appropriate parameters.

A compact final summary of the model equations follows.

$$y_l(u,v) = S_y(v)\left\{\left[1 - T_y(v)\right] y_1(u) + T_y(v) y_2(u)\right\} \tag{33}$$

$$z_l(u,v) = S_z(v)\left\{\left[1 - T_z(v)\right] z_1(u) + T_z(v) z_2(u)\right\} \tag{34}$$

$$x_l(v) = v \tag{35}$$

$$T_{y,z}(v) = \left(1 - v^Q\right)^{1/Q} \tag{36}$$

$$Q = 2/\left(a_{1y,z} + a_{2y,z} v^{a_{3y,z}} + a_{4y,z} v^{a_{5y,z}}\right) \tag{37}$$

$$S_{y,z}(v) = c_{1y,z} + c_{2y,z} v^{c_{3y,z}} + c_{4y,z} v^{c_{5y,z}} \tag{38}$$

$$y_{1,2}(u) = A_3\left\{\left[A_1 \sin^{\epsilon}\left(mu + u_y\right) + y_{t_1}\right]\cos\psi + \left[A_2 \cos^{\epsilon}\left(nu + u_z\right) + z_{t_1}\right]\sin\psi\right\} + y_{t_2} \tag{39}$$

$$z_{1,2}(u) = A_4\left\{-\left[A_1 \sin^{\epsilon}\left(mu + u_y\right) + y_{t_1}\right]\sin\psi + \left[A_2 \cos^{\epsilon}\left(nu + u_z\right) + z_{t_1}\right]\cos\psi\right\} + z_{t_2} \tag{40}$$

$$\epsilon = b_1 + b_2 q^{b_3} \tag{41}$$

$$q = \left|\sin\left(ju + u_q\right)\right| \tag{42}$$

$$y(u,v) = y_l(u,v) \tag{43}$$

$$z(u,v) = z_l(u,v) \tag{44}$$

$$x(u,v) = x_l(v) + \left(1 - x_{ln}^h\right) G_f(u,v) + x_{ln}^h G_b(u,v) \tag{45}$$

$$G_f(u,v) = h_{f1}\left[1 - \left|h_{f2} - y_{fl}\right|^{2/\epsilon_f}\right]^{\epsilon_f/2} + h_{f3}\left[1 - \left|h_{f4} - z_{fl}\right|^{2/\epsilon_f}\right]^{\epsilon_f/2} + h_{f10} \tag{46}$$

$$G_b(u,v) = h_{b1}\left[1 - \left|h_{b2} - y_{bl}\right|^{2/\epsilon_b}\right]^{\epsilon_b/2} + h_{b3}\left[1 - \left|h_{b4} - z_{bl}\right|^{2/\epsilon_b}\right]^{\epsilon_b/2} + h_{b10} \tag{47}$$

$$0 \le x_{ln}, y_{f,bl}, z_{f,bl} \le 1 \tag{48}$$

$$\epsilon_{f,b} = h_{f,b5} + h_{f,b6} y_{f,bl}^{h_{f,b7}} + h_{f,b8} z_{f,bl}^{h_{f,b9}} \tag{49}$$

1.6 Conclusion

A piece-wise analytic function capable of defining a wide range of geometric surfaces can be formed by combining a modified superellipse with a topologically transformed lofted model. The final form of the model's equations represents a reasonable compromise between the desire to keep the model simple (and hence easy to use) and to ensure the capability of modeling complicated surfaces. The interface with this function is an array of user-defined constants. Because these parameters are easily correlated with the shape of the modeled structure, shape definition and alteration is readily implemented. For applications restricted to the well-behaved surfaces of exact analytic models, this function opens a host of new geometries.

2 Analytic Surface Model – Code

2.1 Code User's Manual

The user's manual is a guide to the use of the model-implementing computer code (FORTRAN source module listed in app A). The code is designed to make optimum use of the capabilities implicit in the theoretical development of the previous sections. This means that more flexibility is built into the code than is explicitly indicated by the theoretical treatment. This will become apparent within the body of the user's manual.

All input to the code is via a namelist statement. Each term within the namelist statement is defined below.

2.1.1 xl

xl relates to the longitudinal (x-axis) length of the lofted model. The code has the capability of treating a maximum of 10 independent structures in a single pass. These structures can be joined at common cross sections to form a composite body. There are several advantages to using multiple independent structures to define a single body:

- Each independent structure can be developed and tested separately, reducing the modeling burden on the user.
- Structures that are not monotonic in x can be modeled.
- The parameters within the size and transition functions can be changed over the length of the composite structure.

xl is the x-axis location of all model cross sections (station cuts) listed in order. xl must be monotonic in x within an independent structure but need not be monotonic between structures. Where independent structures are to be joined at a common cross section, the x-axis location of that station cut must appear twice in the xl sequence. When using a composite body, one must exercise caution to ensure that continuity conditions are met at all interfaces.

2.1.2 ncount

ncount is the number of independent structures.

2.1.3 ns

ns is the number of model segments defined by station cuts within each independent structure (**ns** = number of station cuts less 1).

2.1.4 anx

anx relates to the density of calculated surface points and is the surface grid count in the x direction for each independent structure. For the x direction, surface grid density is constant within individual independent structures.

2.1.5 mid

mid relates to the relative placement of the calculated surface point for the x coordinate of the position vector. For **mid** = 0, the first point falls on the first grid line, and subsequent points on subsequent grid lines. The first grid line is coincident with the corresponding first station cut. For **mid** $\neq$ 0, each point falls midway between grid lines.

2.1.6 npoint

npoint is the surface grid count transverse to the x direction for each independent block.

2.1.7 ang

ang determines the relative point distribution about the perimeter of each station cut. Each station cut can be composed of a maximum of 15 curve segments. The summation of **ang** for all 15 segments (or utilized fraction thereof) must equal 360. The grid spacing is, generally, not uniform within each curve segment but tends to cluster more densely in regions where curvature is changing most rapidly.

2.1.8 nseg

nseg is the number of curve segments contained within each station cut for a maximum of 25 station cuts.

2.1.9 uy

uy is the starting value in degrees of the parametric variable, u, for the y coordinate of each individual segment composing a station cut. It corresponds to u_y in equation (39).

2.1.10 uz

uz is the corresponding value for the z coordinate, and for the present code structure *must be set identically equal to* **uy**. It corresponds to u_z in equation (39).

2.1.11 alim

alim is the corresponding stopping value in degrees of the parametric variable, u, for both y and z coordinates independent of the values of **am** and **an**. **alim** can be made larger or smaller than **uy** and **uz**. For values of **alim** or **uy** and **uz** spanning 360 degrees, use an appropriate negative value for the stop or start point so that the absolute value of **alim** minus **uy** remains the angular extent of the spanned interval. To calculate the appropriate value of **alim**, temporarily treat parameters **am** and **an** as equal to 1. In this way the full angular extent spanned in the definition of a cross section or cross-section segment can be divorced from the 0 to 360 degree span of the parametric variable, u.

2.1.12 uq

uq corresponds to u_q in equation (42) and is analogous to **uy** and **uz**. The range spanned by the argument of the sin function in equation (42) is governed by **alim**.

2.1.13 am

am corresponds to m in equation (39).

2.1.14 an

an corresponds to n in equation (39).

2.1.15 aj

aj corresponds to j in equation (42).

2.1.16 b1 b2 b3

b1, **b2**, and **b3** correspond to b_1, b_2, and b_3 in equation (41).

2.1.17 a b

a and **b** correspond to A_1 and A_2, respectively, in equation (39).

2.1.18 aa bb

aa and **bb** correspond to A_3 and A_4, respectively, in equations (39) and (40).

2.1.19 psi

psi corresponds to ψ in equation (39).

2.1.20 zt yt

zt and **yt** correspond to z_{t_2} and y_{t_2}, respectively, in equations (39) and (40).

2.1.21 ztp ytp

ztp and **ytp** correspond to z_{t_1} and y_{t_1}, respectively, in equation (39).

2.1.22 a1 a2 a3 a4 a5

a1,..., **a5** correspond to a_1,...,a_5, respectively, in equation (37), where the transition function is identical for both y and z.

2.1.23 ay1 ay2 ay3 ay4 ay5

ay1,..., **ay5** correspond to $a_{1y}, \ldots, a_{5y}$ in equation (37), where the transition function is different for y and z.

2.1.24 az1 az2 az3 az4 az5

az1,..., **az5** correspond to $a_{1z}, \ldots, a_{5z}$ in equation(37), where the transition function is different for y and z.

2.1.25 c1 c2 c3 c4 c5

c1,..., **c5** correspond to $c_1, \ldots, c_5$ in equation (38), where the size function is identical for both y and z.

2.1.26 cy1 cy2 cy3 cy4 cy5

cy1,..., **cy5** correspond to $c_{1y}, \ldots, c_{5y}$ in equation (38), where the transition function is different for y and z.

2.1.27 cz1 cz2 cz3 cz4 cz5

cz1,..., **cz5** correspond to $c_{1z}, \ldots, c_{5z}$ in equation (38), where the transition function is different for y and z.

2.1.28 h hf1 hf2 hf3 hf4 hf5 hf6 hf7 hf8 hf9 hf10

h, **hf1**,..., **hf10** correspond to $h, h_{f1}, \ldots, h_{f10}$ in equations (46) and (49).

2.1.29 hb1 hb2 hb3 hb4 hb5 hb6 hb7 hb8 hb9 hb10

hb1,..., **hb10** correspond to $h_{b1}, \ldots, h_{b10}$ in equations (47) and (49).

2.1.30 ntop

ntop(n) = 1 is used to show that a topological transform is to be performed upon independent structure, n. Otherwise, **ntop(n)** is set to 0.

2.1.31 nwi

nwi determines the x interval over which the topological transform operates within an independent structure. If **nwi** is set to 1, then the topological transform operates over the entire independent structure. For **nwi** set to 0, it operates over individual segments (segment count determined by **ns**) within the independent structures.

2.1.32 ngroup

ngroup defines clustering of cross-section curve segments for use by topological transform. **ngroup** is composed of groups of five numbers. The first and third are the cross-section numbers (same value). The second is the starting cross-section curve number. The fourth is the stopping cross-section curve number (the second and fourth are permitted to be the same). The fifth is a requisite 0. Within each five-unit cluster defined by **ngroup**, all previously defined **h** parameters are held constant. List all **ngroup** parameters in ascending order.

2.1.33 list3 nblock

list3 determines, in part, the format of the code output. For **list3** equal to 0, the x component of the surface-defining vector is stored on FORTRAN logical unit jx, y is stored on jx+1, and z is stored on jx+2. For **list3** not equal to 0, x, y, and z are stored on logical unit jx (3e15.6).

nblock allows the output to be stored on separate groups of FORTRAN logical units for different groups of independent structures. With the independent structures numbered sequentially and in ascending order, **nblock** is the independent structure number of the last contiguous member of each independent block. Each independent block is incremented by 3 with jx initially set to 60 (see **list3**). This arrangement permits any combination of **nblock** and **list3**.

2.2 Model Demonstration – Examples

This section provides some guidelines for the use of the model. These guidelines primarily demonstrate the characteristics of the model equations.

2.2.1 Transition Function

In the theory section, the discussion of the requirements imposed on the transition function for interelement continuity (eq. (3)) was based on the most general interpretation of the Barger model. Unfortunately, the requirements of this equation can lead to undesirable structural shapes or severe limitations on the shape of the modeled structure. For this reason, examining the basis for equation (3) is worthwhile. Consider a structure formed by two elements: a (defined by cross sections 1 and 3) and b (defined by cross sections 3 and 2). (Refer to the original NASA paper [ref 1] for details of the derivation of the following equation.) To ensure first-derivative continuity at common cross section 3, the following relationship must be satisfied:

$$\frac{y_2 - y_3}{x_2 - x_3}\frac{dT_b}{dx_b} = \frac{y_3 - y_1}{x_3 - x_1}\frac{dT_a}{dx_a} \tag{50}$$

where an identical expression exists for the z component. For certain values of the x and y parameters, this relationship can be satisfied for conditions other than $dT_b/dx_b = dT_a/dx_a = 0$.

- If $y_1 = y_2 = y_3$, this relationship is satisfied for any form of T_b and T_a. This seemingly trivial example is, in fact, important because many situations can be encountered where continuity is required over only a part of a surface defined by three (or more) identical partial cross sections.

- If $(y_2 - y_3)/(x_2 - x_3) = (y_3 - y_1)/(x_3 - x_1)$, then the relationship of equation (50) is satisfied for $dT_b/dx_b = dT_a/dx_a$. This situation is encountered, for example, for a frustum of a cone where the common cross section is located anywhere between the parallel end surfaces. In general, this condition is valid for surfaces of constant slope—a much encountered situation.

The same relationships can be shown to hold true for higher order derivatives. This broadening of interelement continuity requirements significantly increases the utility of the analytic model.

The form of the exponent in equation (37) was selected because of its ability to transition, in both a complicated and readily predictable fashion, among the first quadrant curves of the superellipse as v goes from 0 to 1. In lieu of generating reams of transition function curves for various combinations of exponent parameters, the most useful tool for designing an appropriate transition function becomes merely a family of curves detailing the behavior of $v^{\aleph}$ for a broad range of values of $\aleph$. This is done in figure 5.

2.2.2 Cross-Section Function

The best way to demonstrate the performance of the cross-section function is to start with the simplest form of the function, demonstrate its performance, and repeat the process as additional, complicating terms are included.

Step 1

$$y = A_1 \sin^{\epsilon}(mu + u_y) \tag{51}$$

$$z = A_2 \cos^{\epsilon}(nu + u_z) \tag{52}$$

For step 1, $A_1 = A_2 = 1$, and ϵ is held constant. The constants u_y and u_z define phase terms for y and z. The consequences of setting $u_y \neq u_z$ are, for the more complex cross-section models, difficult to predict and, for a model based on intuitive parameters, therefore difficult to exploit. For this reason, this inequality is not permitted by the code. The consequences of setting $u_y \neq u_z$ in equations (51) and (52) are demonstrated in figure 6.

The constants m and n define the range of the arguments of the sin and cos functions and, unlike with u_y and u_z, having $m \neq n$ can prove useful. The consequences of setting $m \neq n$ are also demonstrated in figure 6, along with the results of simultaneously varying both sets of parameters. At this point a few observations are appropriate.

- The generated shapes are complicated.
- The generated shapes are not particularly interesting.
- Although the curve shapes should be predictable, they are not easily predictable.

Step 2

$$y = A_1 \sin^{\epsilon}(mu + u_y) \tag{53}$$

$$z = A_2 \cos^{\epsilon}(nu + u_z) \tag{54}$$

$$\epsilon = b_1 + b_2 q^{b_3} \tag{55}$$

$$q = \mid \sin(ju + u_q) \mid \tag{56}$$

In this step, ϵ is made a function of the parametric variable. For the proper operation of equation (55), it is highly desirable to constrain q to the range 0 to 1; hence the form of equation (56). The similarity in form between equation (56) and equations (53) and (54) is not coincidental, as the same considerations apply. Additionally, the similarity between equations (55) and (37) is intentional since, again, the same considerations are involved. The variable ϵ functions by transitioning among the family of curves defined by the superellipse as the parametric variable is exercised over its range. Examples of step 2 equations are given in figure 7. To constrain the number of curve shapes, all results represent variations on a circle. What this means should be obvious from the curves.

Step 3

This step operates off the same set of equations as the previous step, except all parameters are allowed to change whenever the following conditions are satisfied:

$$mu + u_y = \pi k_m/2 \tag{57}$$

$$nu + u_z = \pi k_n/2 \tag{58}$$

$$ju + u_q = \pi k_j/2 \tag{59}$$

where $k_m, k_n, k_j = 0, 1, 2, \ldots$. Since examples of the model at this step can be constructed by splicing together segments of the curves in the previous step, no specific examples are given.

Step 4

The final step allows arbitrary rotation and translation of model segments:

$$y = A_3 \left\{ \left[A_1 \sin^{\epsilon} (mu + u_y) + y_{t_1} \right] \cos\psi + \left[A_2 \cos^{\epsilon} (nu + u_z) + z_{t_1} \right] \sin\psi \right\} + y_{t_2} \tag{60}$$

$$z = A_4 \left\{ - \left[A_1 \sin^{\epsilon} (mu + u_y) + y_{t_1} \right] \sin\psi + \left[A_2 \cos^{\epsilon} (nu + u_z) + z_{t_1} \right] \cos\psi \right\} + z_{t_2} \tag{61}$$

Examples of the capability of the final cross-section model are given in figure 8.

2.2.3 Size Function

The size function provides an independent scaling of the lofted surface. By allowing different size functions for y and z, a more flexible model results. To demonstrate this, a right circular cylinder is to be operated upon by a range of y and z independent size functions. The results are demonstrated in figure 9.

2.2.4 Topological Transformation

The topological transformation derives its name from the manner in which it operates upon the lofted model. The transformation performs an out-of-plane distortion of selected model cross sections with the consequent alteration in surface shape. The similarity in form between the equations of the lofting procedure and the topological transformation precludes the need for any extended discussion of model behavior except to note that parameters $h_{f,b2}$ and $h_{f,b4}$ are generally restricted to values of 1 or 0. An example of a topologically transformed structure is given in figure 10. The starting lofted model was two identical, joined, right-angle rectangular tubes.

2.2.5 Three-Dimensional Model

To demonstrate the performance of the analytic surface model, three structures are modeled. The first is an abstract geometry having the general shape of a complex jet inlet duct. The second is the front end of a partial jet turbine. And the final is a model of a missile. Three views of the first model are given in figure 11. The second model is given in figure 12. It should be noted that, in this figure, the turbine blades are not simple plates but, rather, possess a finite thickness away from each edge. A full disk of 36 blades could have been incorporated but would have required a redimensioning of the model-implementing computer code—more on this in appendix A. The final model is given in figures 13 and 14. To demonstrate

the role of the topological transformation in this model, the first figure is the missile with the transformation suppressed. It should be noted that all tail surfaces taper from an edge to a finite central thickness and that the wings have a complex airfoil shape characteristic of subsonic aircraft. From the first two models, it should be apparent that an internal inlet duct, turbine front end, exhaust duct, and engine outlet could have been incorporated into this model. It must be stressed that the blades of figure 12 and the wing and tail surfaces of figures 13 and 14 are not modeled separately and then attached to the central body but, rather, are designed as an integral part of the central body. This is the only reasonable way of avoiding the inaccuracy inherent in the attempted mating of surfaces with complex shapes.

2.2.6 Concluding Remarks

The equations of the analytic model represent the best perceived compromise between simplicity—with all its benefits—and the need to model complicated surfaces in the most direct way possible. As a consequence, the most productive upgrades are not to the model but rather to the software implementation of the model. For instance, one useful improvement to the code is to permit a rotation and translation in three dimensions of individual model independent structures. Another is a further generalization of conditions for defining the interval over which the topological transformation operates. It is expected that future versions of the code will contain these upgrades.

The key to the proper use of the model is the ability to define an appropriate set of cross sections. As has been noted, cross sections are formed by joining limited numbers of curve segments. Ideally, one would hope that the cross-section function could define the sought-for shapes with a single curve and, in lieu of this, with as few curves as possible. To construct a cross section from curve segments, it is very helpful to define, based on the cross-section function, a set of curve primitives. One obvious choice is the superellipse itself, or segments thereof. Figures 6 and 7 define other potential candidates. Experience indicates that a potentially fruitful area to explore for defining curve primitives would be cross-section functions where parameters m and n are not set equal or set to 1. Such curves do not have the simplicity or visual appeal of figure 7, as an example, but do appear to generate potentially useful curve shapes.

Figure 1. Superellipse. From rectangle, $\epsilon = 0.01$, 0.5, 1, 2, 4, 8.

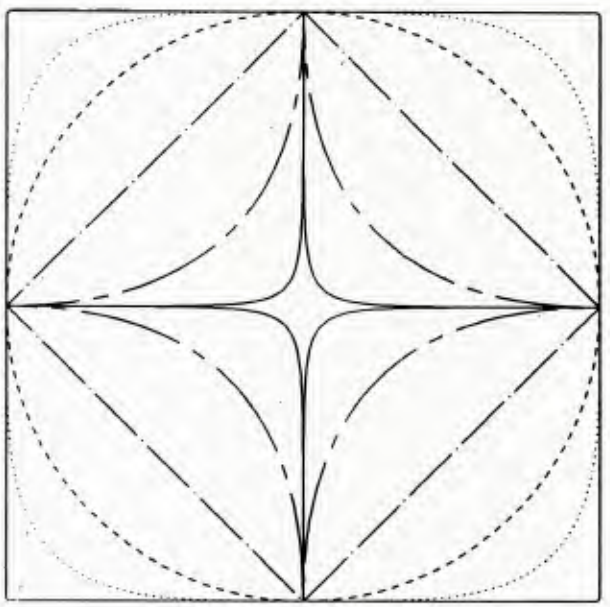


Figure 2. Transition function based on modified superellipse. Curve parameters are listed in appendix B.

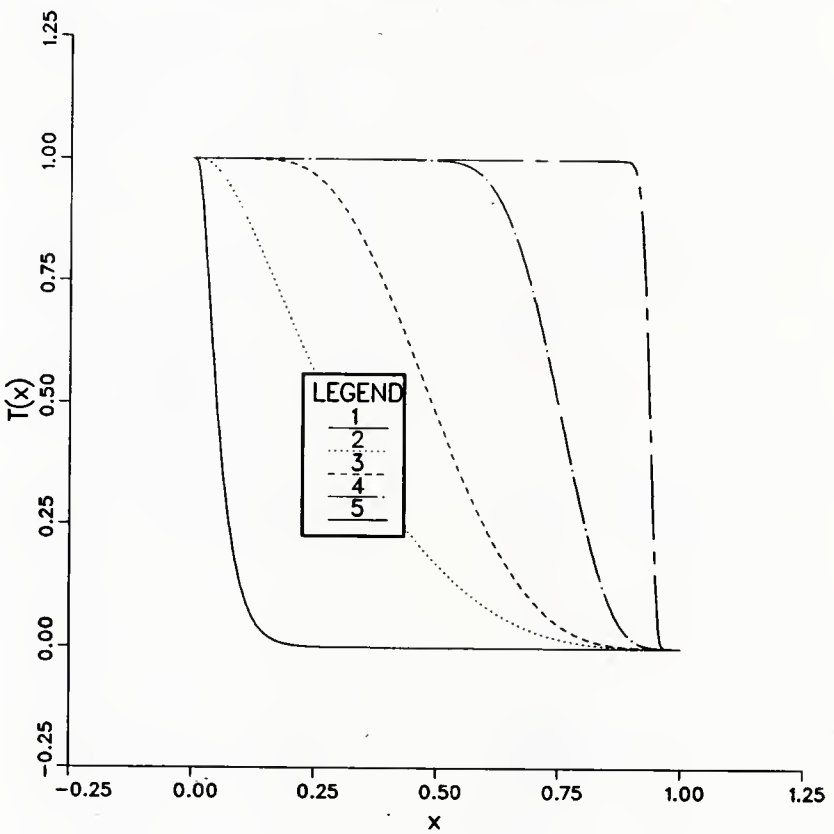


Figure 3. Absolute value of first derivative of transition function.

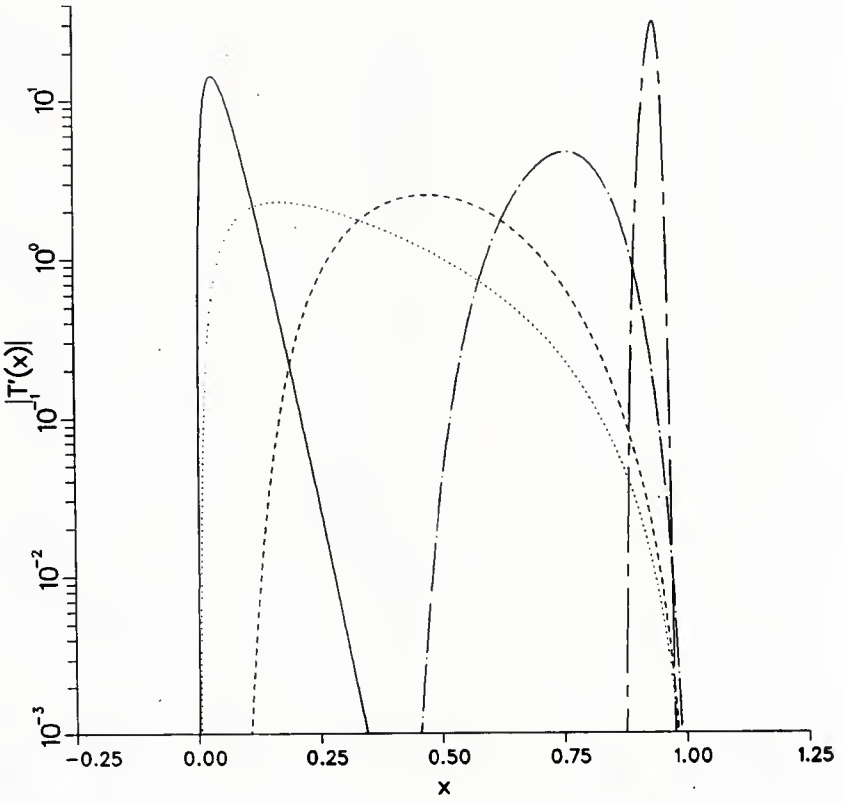
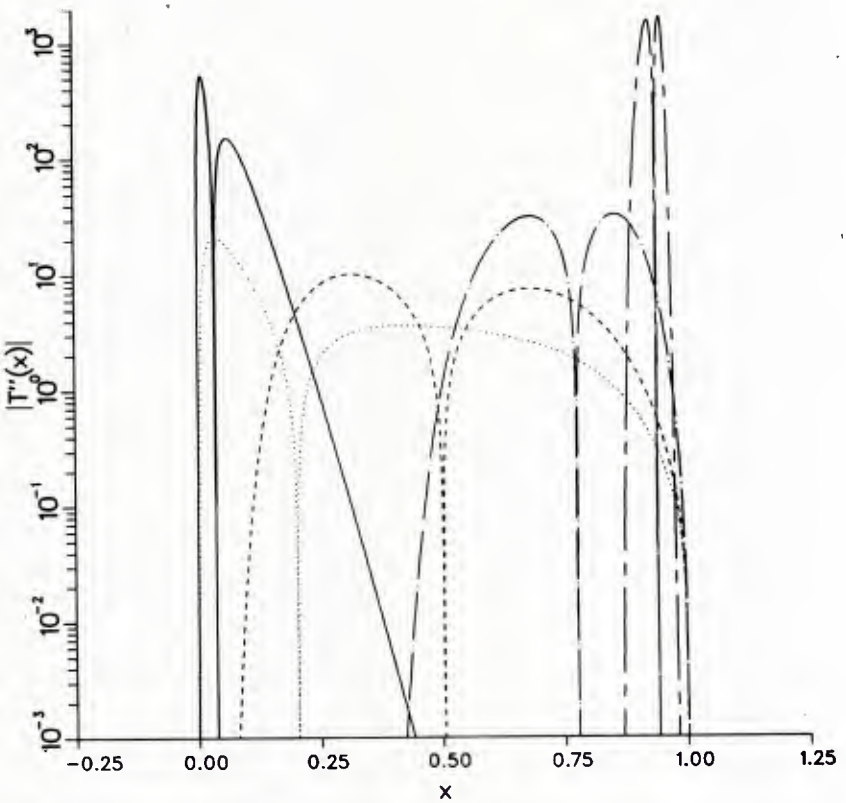


Figure 4. Absolute value of second derivative of transition function.



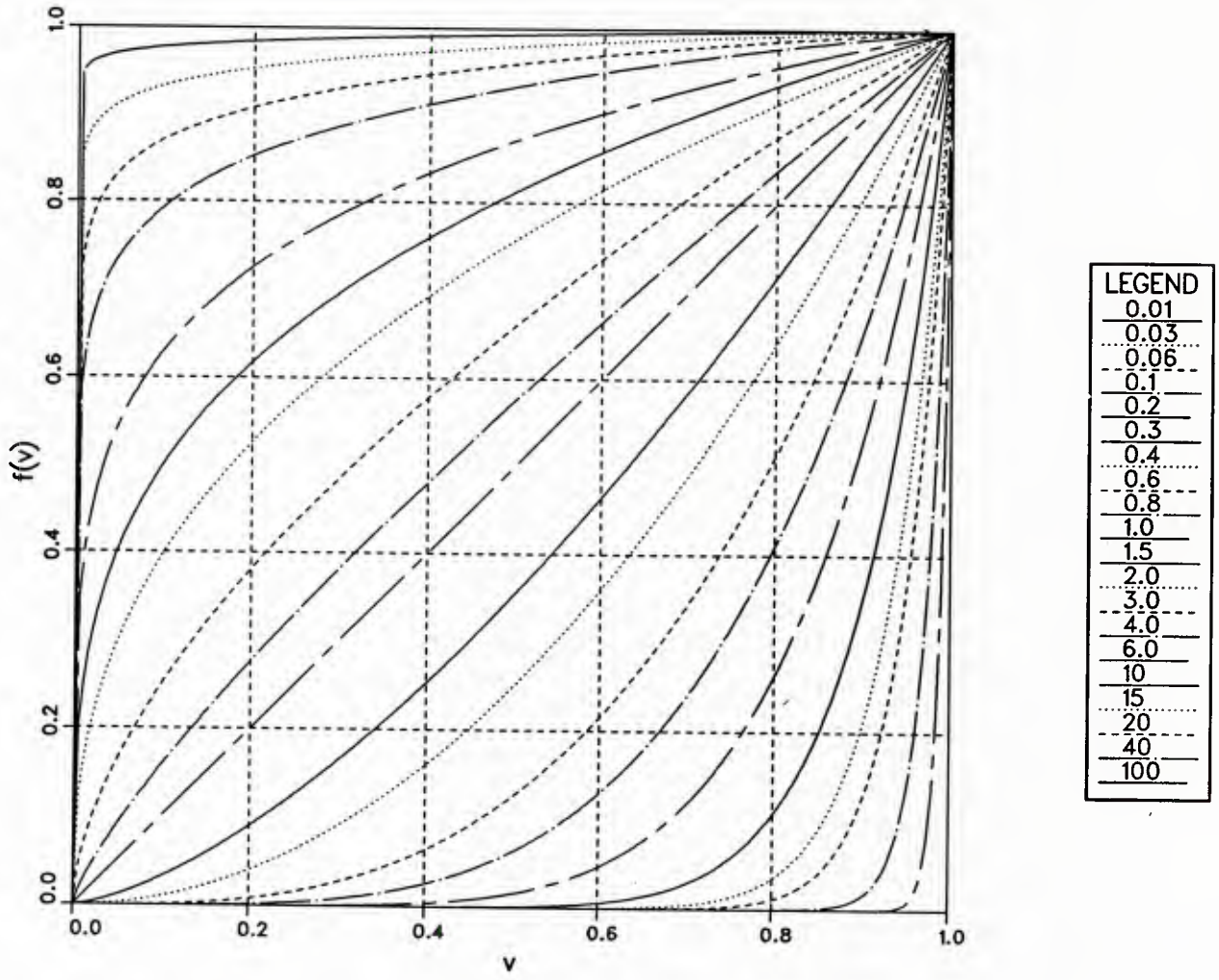
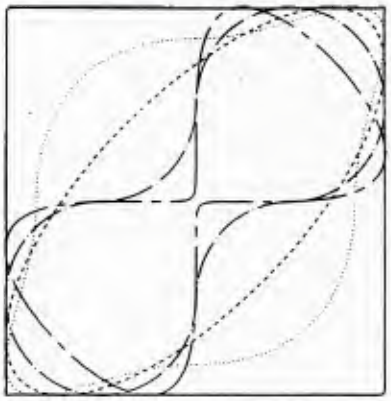


Figure 5. Curves of $f(v) = v^{\aleph}, \aleph$ given in legend.

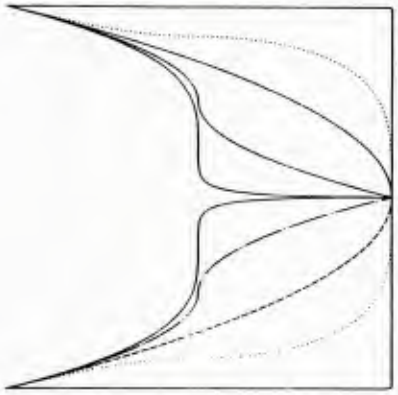
$m = n = 1 \;\; u_y = 0 \;\; u_z = 45$

$\epsilon = 0.01, 0.5, 1, 2, 4$



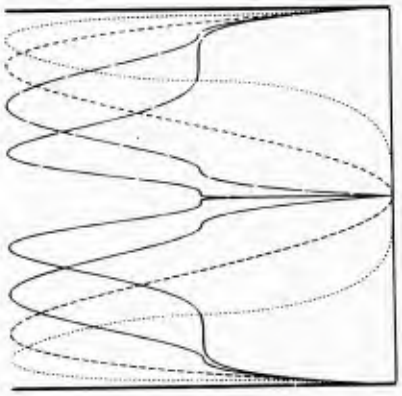
$m = 1 \;\; n = 2 \;\; u_y = u_z = 0$

$\epsilon = 0.01, 0.5, 1, 2, 4$



$m = 1 \;\; n = 4 \;\; u_y = u_z = 0$

$\epsilon = 0.01, 0.5, 1, 2, 4$



$m = 1 \;\; n = 2 \;\; u_y = 0 \;\; u_z = 45$

$\epsilon = 0.01, 0.5, 1, 2, 4$

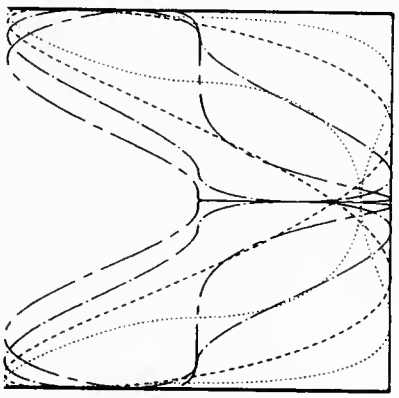
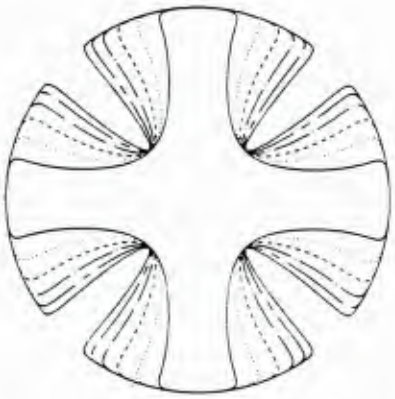


Figure 6. Examples of cross-section function–step 1 equations.

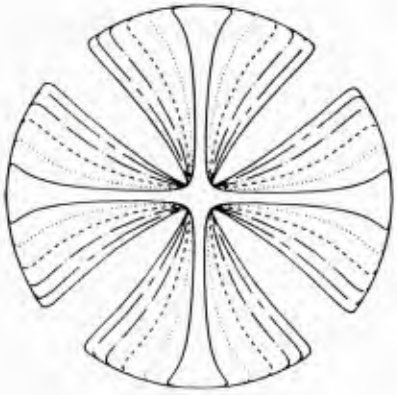
$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 2$

$b_1 = 1 \;\; b_2 = 3 \;\; b_3 = 4, 8, 16, 32, 64, 128$



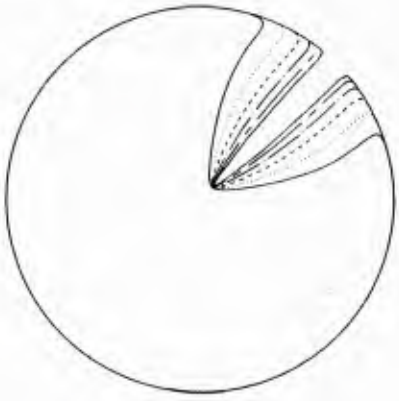
$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 2$

$b_1 = 1 \;\; b_2 = 7 \;\; b_3 = 4, 8, 16, 32, 64, 128$



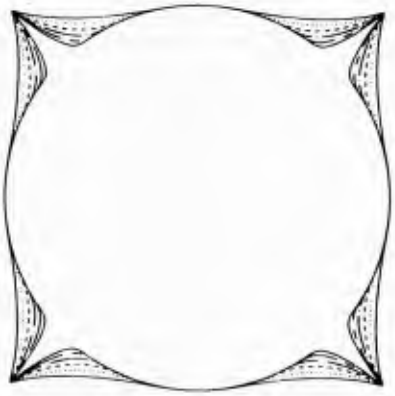
$m = n = 1 \;\; u_y = u_z = 0 \;\; u_q = 67.5 \;\; j = 0.5$

$b_1 = 1 \;\; b_2 = 7 \;\; b_3 = 200, 400, 800, 1600, 3200, 6400$



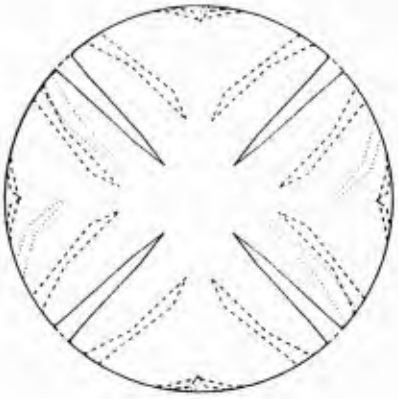
$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 2$

$b_1 = 1 \;\; b_2 = -0.9 \;\; b_3 = 4, 8, 16, 32, 64, 128$



$m = n = 1 \;\; u_y = u_z = 0 \;\; u_q = 0, 22.5, 0, \;\; j = 2, 3, 8$

$b_1 = 1 \;\; b_2 = 4 \;\; b_3 = 900, 500, 100$



$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 2$

$b_1 = 1 \;\; b_2 = 1 \;\; b_3 = 0.1, 1, 2, 4, 8$

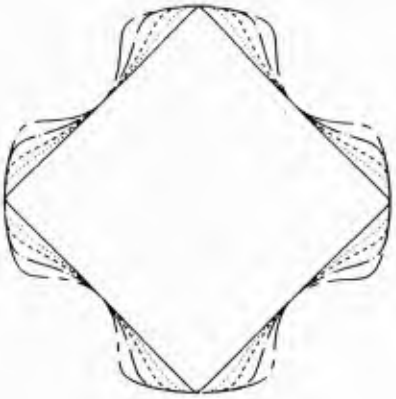
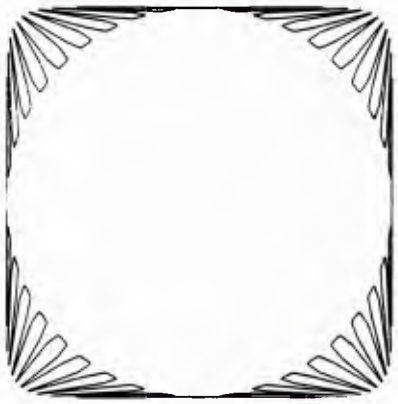


Figure 7a. Examples of cross-section function–step 2 equations.

$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 20$

$b_1 = 1 \;\; b_2 = -0.95 \;\; b_3 = 10$

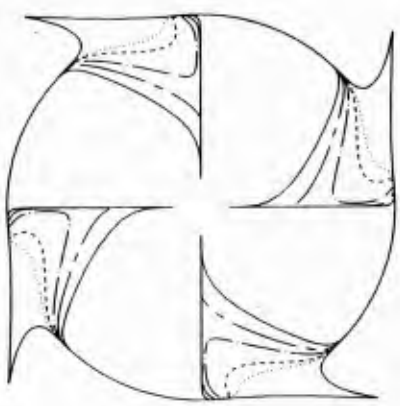


$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 20$

$b_1 = 2 \;\; b_2 = -1 \;\; b_3 = 0.05$

$m = n = 1 \;\; u_y = u_z = 0 \;\; u_q = 45 \;\; j = 2$

$b_1 = 1 \;\; b_2 = -0.9, 0.5, 1, 3, 7, 20 \;\; b_3 = 20$



$m = n = 1 \;\; u_y = u_z = u_q = 0 \;\; j = 20$

$b_1 = 10 \;\; b_2 = -9 \;\; b_3 = 0.05$

Figure 7b. Examples of cross-section function–step 2 equations.

Figure 8. Examples of cross-section function–step 4 equations. Curve parameters are listed in appendix B.

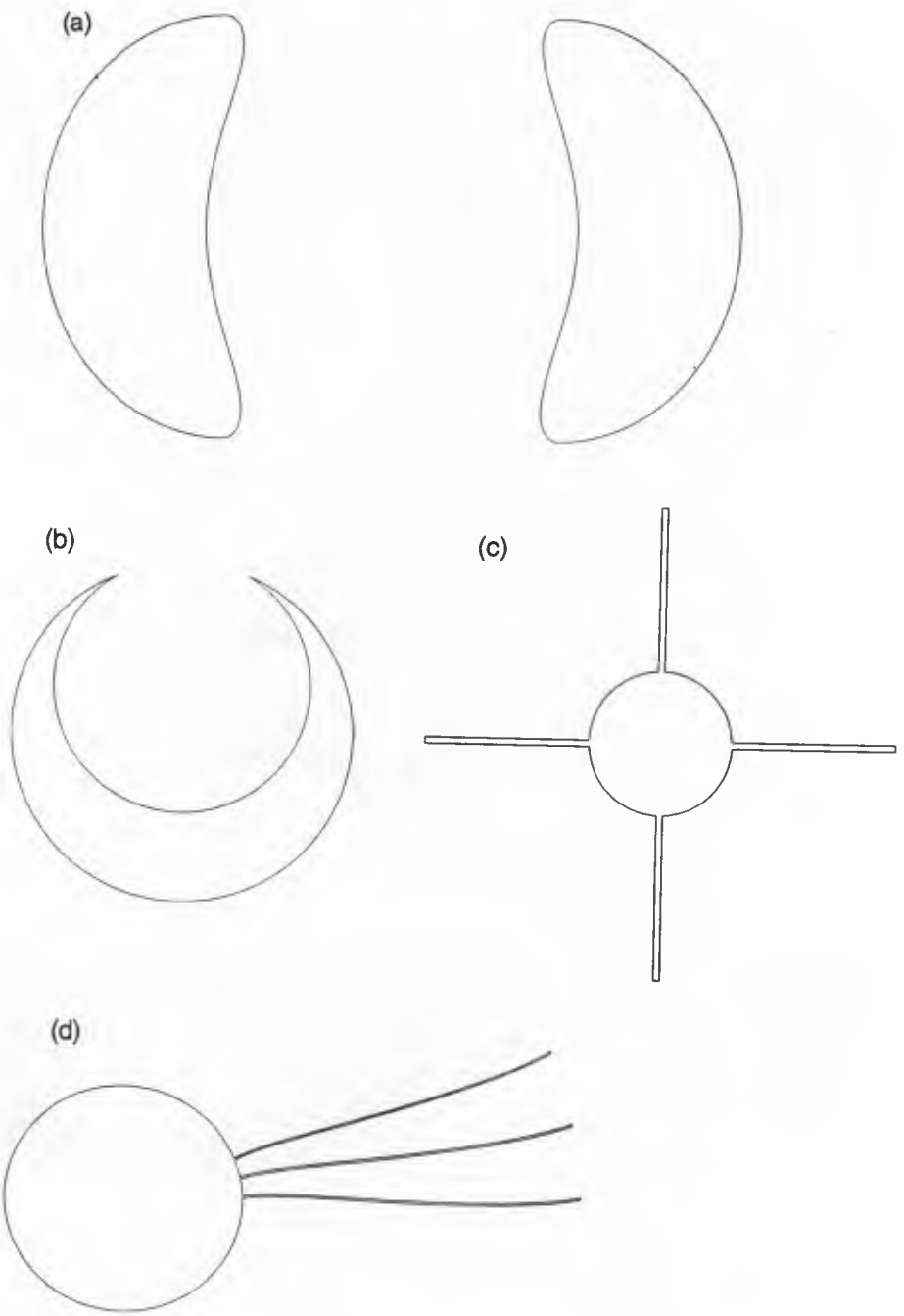


Figure 9. Examples of effect of size function equations on right circular cylinder geometry. Curve parameters are listed in appendix B.

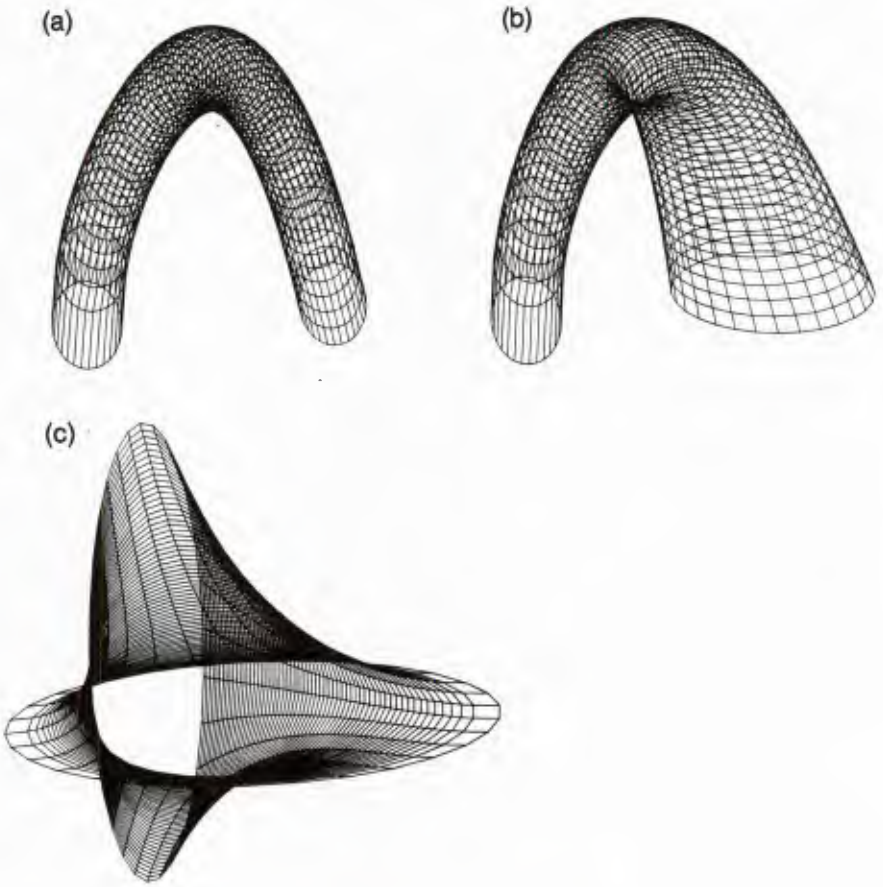


Figure 10. Topologically transformed rectangular cross-section tube. Surface parameters are listed in appendix B.

Figure 11. Duct model.

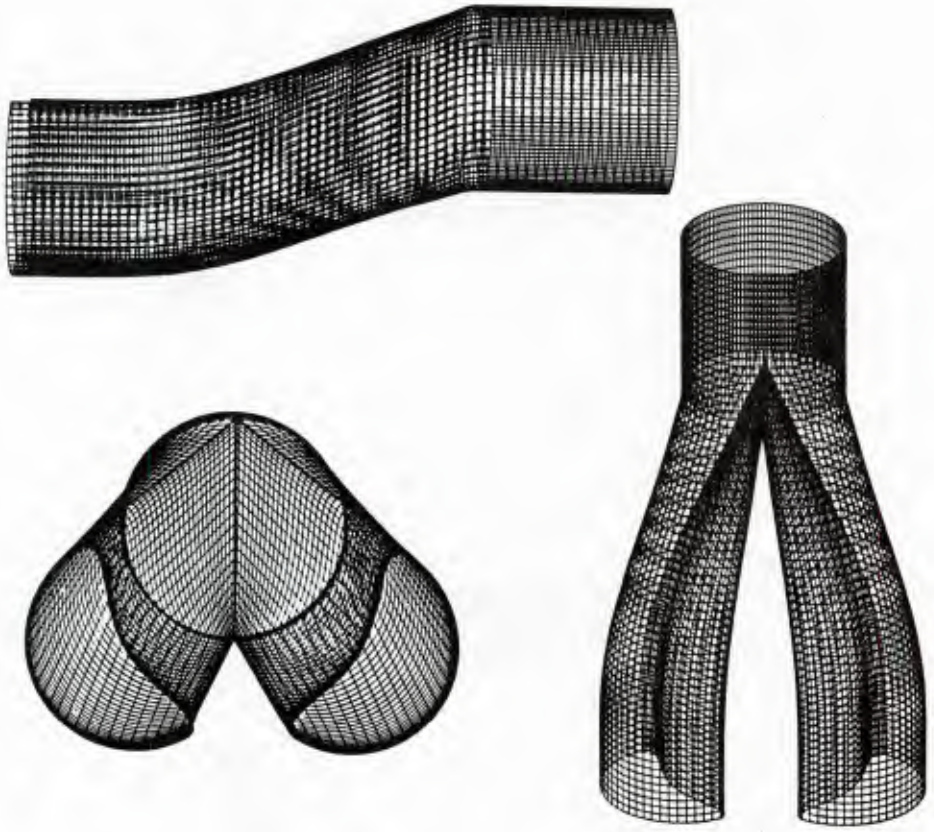


Figure 12. Turbine front end with three blades. Surface parameters are listed in appendix B.

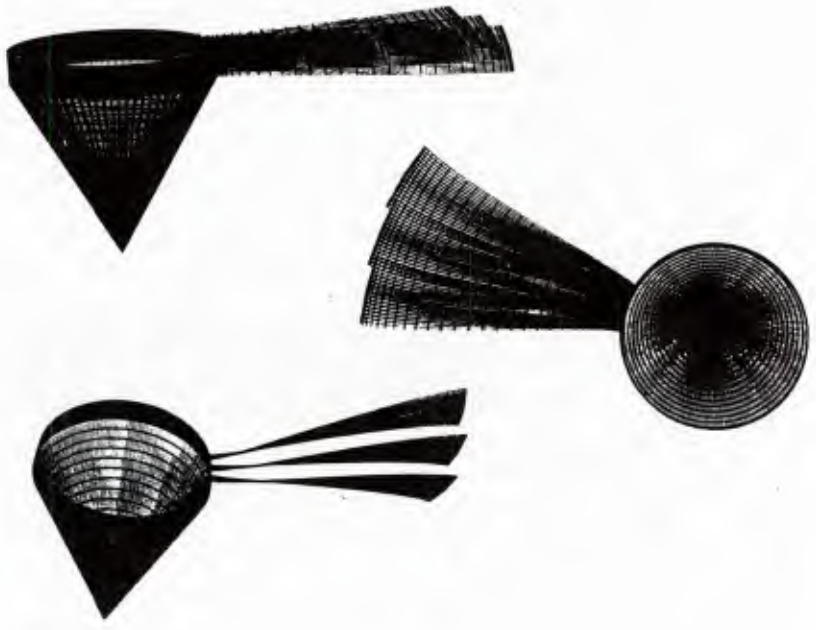


Figure 13. Exploded view of missile before topological transformation (model composed of five independent structures).

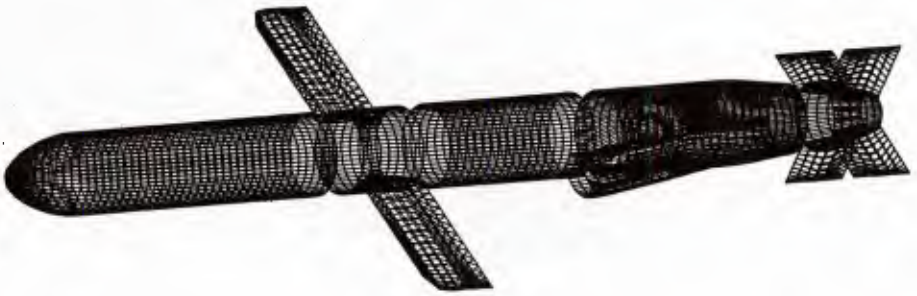
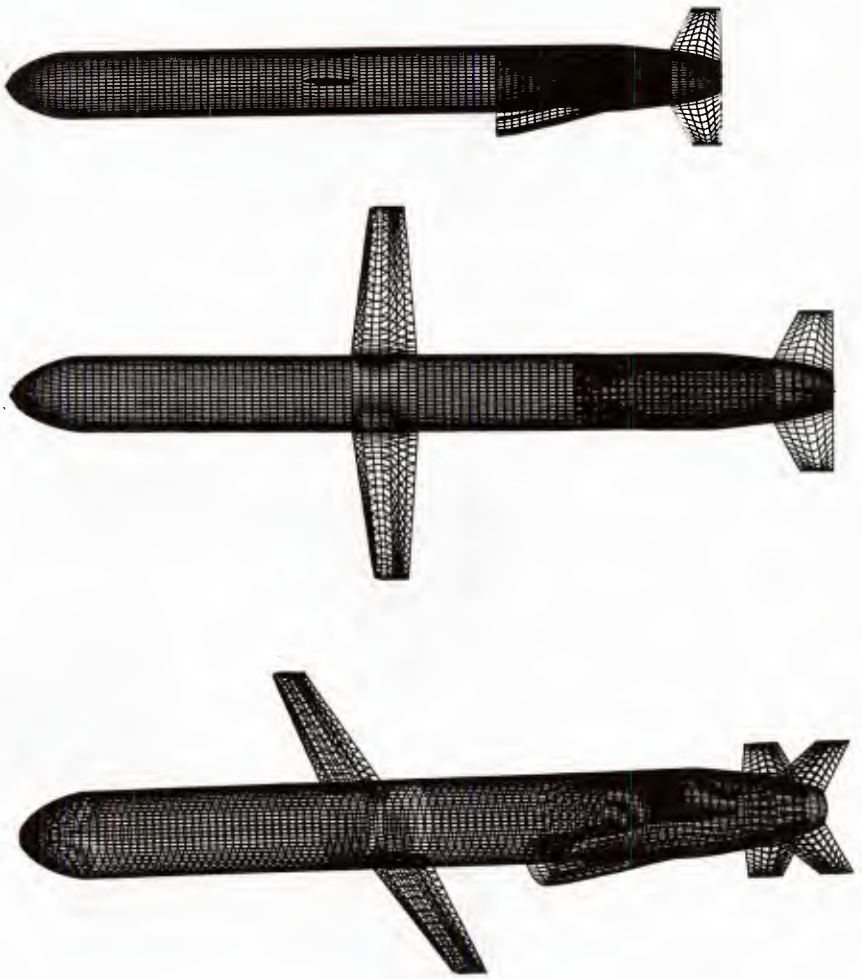


Figure 14. Missile after topological transformation. Parameters are listed in appendix B.



Appendix A

Source Code Listing

Included within this appendix is a FORTRAN source code listing of the analytic surface model and a listing of a code for implementing only the cross-section function.

If the user wishes to increase the limits on the various parameters of the analytic surface model code, this is especially easy to do. The code limits on parameters such as number of curves per cross section or number of cross sections per model (as defined in the code user's manual section) were all selected to have different values. This allows the user to readily determine the dimension statements that require changes. The code is presently dimensioned to calculate a maximum 120 by 240 grid of surface points (120 in the x (longitudinal) direction and 240 transverse). Again, these limits are readily changed within the block of dimension statements. The only other potential dimension statement upgrade is to ngroup (defined in the user's manual). In addition, the following lines of the surface code may need to be changed after any redimensioning:

- line 178 (do 212 jbj=1,15)
- line 187 (do 202 jvv=id,2000,5)
- line 264 (do 110 ic=1,15)
- line 288 (do 112 ic=1,15)

The upper limit on the statements of items 1, 3, and 4 reflects the maximum permissible number of curve segments per cross section. For item 2, the limit of 2000 should reflect the size of the dimension statement for array ngroup.

```
c  ANALYTIC SURFACE MODEL CODE
c
      dimension  x(120), xl(25), yc(240,25), zc(240,25)
      dimension a1(20), a2(20), a3(20), a4(20), a5(20), nblock(10)
      dimension y(240,120), z(240,120), ns(10), anx(10)
      dimension c1(10), c2(10), c3(10), c4(10), c5(10), xt(240,120)
      dimension an(15,25),am(15,25),alim(15,25),uy(15,25),uz(15,25)
      dimension aj(15,25),b1(15,25),b2(15,25),b3(15,25),a(15,25)
      dimension b(15,25),psi(15,25),yt(15,25),zt(15,25)
      dimension aa(15,25),bb(15,25),ang(15,25), npoint(10)
      dimension nseg(25),ytp(15,25),ztp(15,25),uq(15,25)
      dimension cz1(10), cz2(10), cz3(10), cz4(10), cz5(10)
      dimension cy1(10), cy2(10), cy3(10), cy4(10), cy5(10)
      dimension az1(20), az2(20), az3(20), az4(20), az5(20)
      dimension ay1(20), ay2(20), ay3(20), ay4(20), ay5(20)
      dimension ymin(15,25), ymax(15,25), zmin(15,25), zmax(15,25)
      dimension npn(15,25), npnn(15,25), yminn(15,2), ymaxx(15,2)
      dimension zminn(15,2), zmaxx(15,2), ngroup(2000)
      dimension h(15,25), hf1(15,25), hf2(15,25), hf3(15,25)
      dimension hf4(15,25), hf5(15,25), hf6(15,25), hf7(15,25)
      dimension hf8(15,25), hf9(15,25), hb1(15,25), hb2(15,25)
      dimension hb3(15,25), hb4(15,25), hb5(15,25), hb6(15,25)
      dimension hb7(15,25), hb8(15,25), hb9(15,25)
      dimension hf10(15,25), hb10(15,25), ntop(10), nwi(10)
c
      namelist/lista/an,am,alim,uy,uz,aj,b1,b2,b3,a,b,psi,yt,zt,
     &ang,nseg,npoint,aa,bb,ytp,ztp,uq,nblock,list3,
     &a1,a2,a3,a4,a5,ns,anx,xl,c1,c2,c3,c4,c5,ncount,mid,
     &cz1,cz2,cz3,cz4,cz5,cy1,cy2,cy3,cy4,cy5,
     &az1,az2,az3,az4,az5,ay1,ay2,ay3,ay4,ay5,
     &h,hf1,hf2,hf3,hf4,hf5,hf6,hf7,hf8,hf9,hf10,
     &hb1,hb2,hb3,hb4,hb5,hb6,hb7,hb8,hb9,hb10,ngroup,nwi,ntop
c
c  treat each block of sections defined by a discontinuous station cut
c  as independent...ncount= number of independent blocks- max.=5
c  specify all station cuts for each independent block
c  ns= number of model 3-D segments defined by station cuts
c  for each independent block...max. value= 20
c  anx= surface grid count in x direction
c
c
c
```

```
c
      read(3,lista)
      jv=1
      jw=0
      kq=0
      kqq=0
      nsa=1
88    i=0
      ia=1
      do 60 kz=1,ncount
      kzq=anx(kz)+1.1
      nsb=ns(kz)+nsa
      w=0.
      do 50 j=nsa,nsb-1
      w=xl(j+1)-xl(j)+w
 50   continue
      w=w/(anx(kz)-1.)
      if(xl(nsa+1).lt.xl(nsa)) then
        signa=-1.
      else
        signa=+1.
      end if
      jj=0
      jq=kq+1
      if(mid.ne.0) then
        wa=xl(nsa)+w/2.
        go to 2
      end if
      wa=xl(nsa)
2     do 62 j=1,kzq
      if(signa.gt.0.) then
        if(wa.gt.xl(nsb)) go to 63
        kq=1+kq
        x(kq)=wa
        wa=x(kq)+w*(1.-1.e-05)
        go to 62
      else
        if(wa.lt.xl(nsb)) go to 63
        kq=1+kq
        x(kq)=wa
        wa=x(kq)+w*(1.-1.e-05)
      end if
```

```
  62  continue
  63  continue
c  cross-section definitions:
c  store y and z coordinates of surface points in arrays y(n,m), z(n,m)
c  where m defines the x interval
c  nseg= number of curve segments- max.=15
c  npoint= number of plot points
c  ang= angular extent of each segment to be used to define point
c  distribution between segments    summation(ang)=360
c  alim= angular stop points for each segment for the sin and cos
c  arguments independent of parameters am and an
c  allow for axial scaling before and after rotation
c  each segment to be rotated and translated independently
      do 1 j=nsa,nsb
      k=0
      npp=0
      do 3 n=1,nseg(j)
      np=npoint(kz)*ang(n,j)/360.+0.001
      npn(n,j)=np
c  np and npn are number of plot points in each segment
c  always make uy= uz
c  insure that summation of np*nseg= constant for all model sections
      if(np.le.1) go to 5
      uu=(alim(n,j)-uy(n,j))/(np-1)
c  uu= angular increment size
      u=0.
      ymin(n,j)=1.e06
      ymax(n,j)=-1.e06
      zmin(n,j)=1.e06
      zmax(n,j)=-1.e06
      do 4 m=1,np
      k=1+k
      q=abs(sind(aj(n,j)*u+uq(n,j)))
      eps=b1(n,j) + b2(n,j)*q**b3(n,j)
      atest=sind(am(n,j)*u + uy(n,j))
      btest=cosd(an(n,j)*u + uz(n,j))
      aaa=sign(1.,atest)
      bbb=sign(1.,btest)
      yy=aaa*a(n,j)*abs(sind(am(n,j)*u+uy(n,j)))**eps+ytp(n,j)
      zz=bbb*b(n,j)*abs(cosd(an(n,j)*u+uz(n,j)))**eps+ztp(n,j)
      yc(k,j)=aa(n,j)*(yy*cosd(psi(n,j))+zz*sind(psi(n,j)))+yt(n,j)
      zc(k,j)=bb(n,j)*(yy*-sind(psi(n,j))+zz*cosd(psi(n,j)))+zt(n,j)
```

```
      if(yc(k,j).lt.ymin(n,j)) ymin(n,j)=yc(k,j)
      if(yc(k,j).gt.ymax(n,j)) ymax(n,j)=yc(k,j)
      if(zc(k,j).lt.zmin(n,j)) zmin(n,j)=zc(k,j)
      if(zc(k,j).gt.zmax(n,j)) zmax(n,j)=zc(k,j)
c  y-zmin, y-zmax define minimum and maximum values of y and z
c  for each segment of each defined cross section...for use
c  with topological transform
      u=u+uu
  4   continue
  5   continue
      npp=np+npp
  3   continue
      write(*,101)npp
 101  format(i9)
  1   continue
      jjj=jq
      do 64 j=nsa,nsb-1
      i=1+i
c
      id=ia
      loop=0
      jb=j
      mb=1
      ib=0
      if(ngroup(ia).eq.0) go to 100
 102  if(ngroup(ia).ne.jb) go to 103
      ib=1+ib
      yminn(ib,mb)=1.e06
      ymaxx(ib,mb)=-1.e06
      zminn(ib,mb)=1.e06
      zmaxx(ib,mb)=-1.e06
      do 104 ma=ngroup(ia+1),ngroup(ia+3)
      if(ymin(ma,jb).lt.yminn(ib,mb)) yminn(ib,mb)=ymin(ma,jb)
      if(ymax(ma,jb).gt.ymaxx(ib,mb)) ymaxx(ib,mb)=ymax(ma,jb)
      if(zmin(ma,jb).lt.zminn(ib,mb)) zminn(ib,mb)=zmin(ma,jb)
      if(zmax(ma,jb).gt.zmaxx(ib,mb)) zmaxx(ib,mb)=zmax(ma,jb)
      npnn(ib,jb)=npn(ma,jb)+npnn(ib,jb)
      write(*,220)ib,jb,npnn(ib,jb),mb,
     &ia,ngroup(ia),loop
 220  format(7i5)
 104  continue
c  npnn is number of cross-section points for each cluster
```

```
      ia=ia+5
      if(ngroup(ia).eq.jb) go to 102
      if(nwi(kz).ne.0.and.j.ne.(nsb-1).and.loop.eq.1) ia=id
  103 if(nsb-1.eq.j) then
        if(loop.eq.1) go to 100
  201   ib=0
        mb=2
        jb=nsb
        loop=1
        do 212 jbj=1,15
        npnn(jbj,jb)=0
  212   continue
        go to 102
      end if
      if(nwi(kz).eq.0) go to 100
      if(loop.eq.1) go to 100
      if(ib.eq.0) go to 100
      id=ia
      do 202 jvv=id,2000,5
      if(ngroup(jvv).eq.nsb) then
        ia=jvv
        go to 201
      end if
  202 continue
      write(*,203)
  203 format(6h error)
  100 continue
c  ngroup defines clustering of cross-section segments for use by
c  topological transform - example input: 1,1,1,3,0,1,4,1,8,0,2,1,2,8,
c  use 0 to separate clusters ... cluster pairs define array elements
c  for cross-section segments - must be no gaps in any cited
c  cross section or cited independent block
c
c  size function can be specified for y and z axis independently
c  through parameters cy1,cz1,cy2,cz2,... or for both y and z
c  axes through c1,c2,...
c  similarly, for the transition function as ay1,az1,ay2,az2,...
c  or a1,a2,...
      do 65 jj=jjj,kq
      xx=(x(jj)-xl(nsa))/(xl(nsb)-xl(nsa))
      xa=(x(jj)-xl(j))/(xl(j+1)-xl(j))
      jjj=jj
```

```
      if(xa.gt.1.) go to 64
      if(xx.gt.1.) go to 64
      ayz=abs(az1(i))+abs(az2(i))+abs(az3(i))+abs(az4(i))+abs(az5(i))
      if(ayz.lt.1.e-06) then
        lflag=0
      else
        lflag=1
        go to 12
      end if
      eps=2./(a1(i)+a2(i)*xa**a3(i)+a4(i)*xa**a5(i))
      go to 13
12    epsz=2./(az1(i)+az2(i)*xa**az3(i)+az4(i)*xa**az5(i))
      epsy=2./(ay1(i)+ay2(i)*xa**ay3(i)+ay4(i)*xa**ay5(i))
13    cyz=abs(cz1(kz))+abs(cz2(kz))+abs(cz3(kz))+abs(cz4(kz))+
     &abs(cz5(kz))
      if(cyz.lt.1.e-06) then
        kflag=0
      else
        kflag=1
        go to 10
      end if
      s=c1(kz)+c2(kz)*xx**c3(kz)+c4(kz)*xx**c5(kz)
      go to 11
10    sy=cy1(kz)+cy2(kz)*xx**cy3(kz)+cy4(kz)*xx**cy5(kz)
      sz=cz1(kz)+cz2(kz)*xx**cz3(kz)+cz4(kz)*xx**cz5(kz)
11    if(lflag.eq.1)then
        epy=epsy
        epz=epsz
      else
        epy=eps
        epz=eps
      end if
      ty=(1.0-xa**epy)**(1./epy)
      tz=(1.0-xa**epz)**(1./epz)
      kqq=1+kqq
      do 66 n=1,k
      if(kflag.eq.1) then
        syy=sy
        szz=sz
      else
        syy=s
        szz=s
```

```
      end if
      y(n,kqq)=syy*((1.-ty)*yc(n,j+1)+ty*yc(n,j))
      z(n,kqq)=szz*((1.-tz)*zc(n,j+1)+tz*zc(n,j))
c  ntop(n)=1 means that a topological transform is to be
c  performed upon independent block n
c  to include topological transform of lofted model simply add
c  h parameters to namelist input:
      if(ntop(kz).eq.0) go to 80
      gf=0.
      gb=0.
      nx=0
      do 110 ic=1,15
      nx=npnn(ic,j)+nx
      if(nx.ge.n) go to 111
 110  continue
 111  icc=ic
c  icc determines the operational cross-section cluster
      ymm=ymaxx(icc,1)-yminn(icc,1)
      zmm=zmaxx(icc,1)-zminn(icc,1)
      if(ymm.lt.1.e-06) then
        yf=.5
        go to 115
      end if
      yf=abs((y(n,kqq)-yminn(icc,1))/(ymaxx(icc,1)-yminn(icc,1)))
 115  if(zmm.lt.1.e-06) then
        zf=.5
        go to 116
      end if
      zf=abs((z(n,kqq)-zminn(icc,1))/(zmaxx(icc,1)-zminn(icc,1)))
 116  epf=hf5(icc,j)+hf6(icc,j)*yf**hf7(icc,j)+hf8(icc,j)*zf**hf9(icc,j)
      gf=hf1(icc,j)*abs(1.-abs(hf2(icc,j)-yf)**(2./epf))**(0.5*epf)
     &  +hf3(icc,j)*abs(1.-abs(hf4(icc,j)-zf)**(2./epf))**(0.5*epf)
     & +hf10(icc,j)
      if(mb.ne.2) go to 114
      nx=0
      do 112 ic=1,15
      nx=npnn(ic,nsb)+nx
      if(nx.ge.n) go to 113
 112  continue
 113  icc=ic
      ymm=ymaxx(icc,2)-yminn(icc,2)
      zmm=zmaxx(icc,2)-zminn(icc,2)
```

```
      if(ymm.lt.1.e-06) then
        yb=.5
        go to 117
      end if
      yb=abs((y(n,kqq)-yminn(icc,2))/(ymaxx(icc,2)-yminn(icc,2)))
 117  if(zmm.lt.1.e-06) then
        zb=.5
        go to 118
      end if
      zb=abs((z(n,kqq)-zminn(icc,2))/(zmaxx(icc,2)-zminn(icc,2)))
 118  epb=hb5(icc,nsb)+hb6(icc,nsb)*yb**hb7(icc,nsb)+
     &hb8(icc,nsb)*zb**hb9(icc,nsb)
      gb=hb1(icc,nsb)*abs(1.-abs(hb2(icc,nsb)-yb)**(2./epb))**(0.5*epb)
     &  +hb3(icc,nsb)*abs(1.-abs(hb4(icc,nsb)-zb)**(2./epb))**(0.5*epb)
     & +hb10(icc,nsb)
 114  continue
c  limits on topological transform can span either width of independent
c  block (specify nwi=1) or width of segment - for width of independent
c  block: repeat hf parameters for each incorporated cross-section.
c  Need include hb parameters for cross-section nsb only for nwi=1 or 0.
      if(nwi(kz).eq.1) then
        xax=xx
      else
        xax=xa
      end if
      xtt=gf*(1.-xax**h(icc,j))+gb*xax**h(icc,j)
      gf=0.
      gb=0.
 80   xt(n,kqq)=x(jj)+xtt
      xtt=0.
c  nblock allows independent blocks to be stored on separate groups of
c  fortran logical units- nblock defines last block of each group
      if(kz.le.nblock(jv)) go to 70
      jv=1+jv
      jww=jww+3
 70   jx=60+jww
      if(list3.ne.0.and.ngroup(1).eq.0) then
        write(jx,192)x(kqq),y(n,kqq),z(n,kqq)
 192    format(3e15.6)
        go to 66
      else if(list3.ne.0.and.ngroup(1).ne.0) then
        write(jx,192)xt(n,kqq),y(n,kqq),z(n,kqq)
```

```
        go to 66
      end if
      if(ngroup(1).eq.0.) then
        write(jx,190)x(kqq)
        write(jx+1,190)z(n,kqq)
        write(jx+2,190)y(n,kqq)
190     format(e16.6)
        go to 66
      else
        write(jx,190)xt(n,kqq)
        write(jx+1,190)z(n,kqq)
        write(jx+2,190)y(n,kqq)
      end if
66    continue
65    continue
64    continue
      nsa=nsb+1
60    continue
      stop
      end
```

```
c  CROSS-SECTION CODE
c
      dimension an(90,5), am(90,5), alim(90,5), uy(90,5), uz(90,5)
      dimension aj(90,5), b1(90,5), b2(90,5), b3(90,5), a(90,5)
      dimension b(90,5), psi(90,5), yt(90,5), zt(90,5), y(720,5)
      dimension z(720,5), aa(90,5), bb(90,5), nbreak(90,5), ang(90,5)
      dimension nseg(5), ytp(90,5), ztp(90,5), uq(90,5)
      namelist/lista/an,am,alim,uy,uz,aj,b1,b2,b3,a,b,psi,yt,zt,
     &ang,nseg,nplot,npoint,aa,bb,nbreak,ytp,ztp,uq
      read(25,lista)
c  nseg= number of curve segments
c  npoint= number of plot points
c  ang= angular extent of each segment to be used to define point
c  distribution between segments    summation(ang)=360
c  alim= angular stop points for each segment for the sin and cos
c  arguments independent of parameters am and an
c  allow for axial scaling before and after rotation
c  each segment to be rotated and translated independently
      do 1 j=1,nplot
      k=0
      jj=1
      do 3 n=1,nseg(j)
      np=npoint*ang(n,j)/360.
c  np is number of plot points in each segment
c  always make uy= uz
      uu=(alim(n,j)-uy(n,j))/(np-1)
c  uu= angular increment size
      u=0.
      do 4 m=1,np
      k=1+k
      q=abs(sind(aj(n,j)*u+uq(n,j)))
      eps=b1(n,j) + b2(n,j)*q**b3(n,j)
      atest=sind(am(n,j)*u + uy(n,j))
      btest=cosd(an(n,j)*u + uz(n,j))
      aaa=sign(1.,atest)
      bbb=sign(1.,btest)
      yy=aaa*a(n,j)*abs(sind(am(n,j)*u+uy(n,j)))**eps+ytp(n,j)
      zz=bbb*b(n,j)*abs(cosd(an(n,j)*u+uz(n,j)))**eps+ztp(n,j)
      y(k,j)=aa(n,j)*(yy*cosd(psi(n,j))+zz*sind(psi(n,j)))+yt(n,j)
      z(k,j)=bb(n,j)*(yy*-sind(psi(n,j))+zz*cosd(psi(n,j)))+zt(n,j)
c
c  nbreak used to set number of segments to be joined to form individual
```

```
c  curves... nbreak defines last segment in each curve
c  j term allow routine to be used for multiple cross section problem
c
      if(n.le.nbreak(jj,j)) go to 5
      jj=jj+1
  5   kk=79+jj
      write(kk,6)z(k,j),y(k,j)
  6   format(2e16.6)
      u=u+uu
  4   continue
  3   continue
  1   continue
      stop
      end
```

Appendix B

Source Code Input Modules

The curve parameters for figures 2 through 4 (as defined in sect. 2.1.22 in main body of text) are included within this appendix. In addition, the appendix contains a listing of the code input (namelist) modules for a number of the figures of this report. These represent sufficient information to demonstrate the performance of the surface code.

DATA FOR FIGURES 2 THROUGH 4

CURVE	a1	a2	a3	a4	a5
1	0.01	20.0	0.5	0.0	---
2	0.01	4.0	0.3	1.0	4.0
3	0.01	1.0	4.0	4.0	1.0
4	0.01	1.0	10.0	4.0	4.0
5	0.01	20.0	50.0	0.0	---

INPUT DATA SET FIGURE 8A

```
&lista npoint=360,nseg=4,ang=90,90,90,90,alim=180,360,180,360,
uy=0,180,0,180,uz=0,180,0,180,uq=0,0,0,0,b1=1,1,1,1,b2=0,0,0,0,
b3=1,1,1,1,am=3,1,3,1,an=1,1,1,1,aj=1,1,1,1,a=.1,1,.1,1,
b=1,1,1,1,aa=1,1,1,1,bb=1,1,1,1,psi=90,90,-90,-90,
zt=1,1,-1,-1,yt=0,0,0,0,ztp=0,0,0,0,ytp=0,0,0,0,nbreak=2,4,nplot=1,
&end
```

INPUT DATA SET FIGURE 8B

```
&lista npoint=360,nseg=2,ang=180,180,alim=360,61.36,uy=45,360,
uz=45,360,uq=0,0,0,0,b1=1,1,b2=0,0,b3=1,1,am=1,-1,an=1,-1,
aj=1,1,a=1,.75,b=1,.75,aa=1,1,bb=1,1,psi=67.5,59.32,zt=0,0,
yt=0,.2789,ztp=0,0,ytp=0,0,nbreak=2,nplot=1,
&end
```

INPUT DATA SET FIGURE 8C

```
&lista npoint=240,nbreak=8,nplot=1,nseg=8,
ang=30,60,30,60,30,60,30,60,
alim=87.45,180,177.45,270,267.45,360,357.45,90,
uy=2.55,-90,92.55,0,182.55,90,272.55,-180,
uz=2.55,-90,92.55,0,182.55,90,272.55,-180,
b1=1,2,1,2,1,2,1,2,b2=8*0,b3=8*1,am=8*1,an=8*1,aj=8*1,
bb=1,.0106,1,.272,1,.0106,1,.272,aa=1,.272,1,.0106,1,.272,1,.0106,
a=.169,1,.169,1,.169,1,.169,1,b=.169,1,.169,1,.169,1,.169,1,
psi=0,45,0,45,0,45,0,45,zt=0,0,0,-.362,0,0,0,.362,
yt=0,.362,0,0,0,-.362,0,0,
&end
```

INPUT DATA SET FIGURE 8D

```
&lista aj=1,1,1,-1,1,1,1,-1,1,1,1,-1,aa=90*1,bb=90*1,
am=1,-1,1,1,1,-1,1,1,1,-1,1,1,1,an=1,-1,1,1,1,-1,1,1,1,-1,1,1,
ang=110,40,0,40,5,40,0,40,5,40,0,40,
uy=27.15,90,0,0,7.15,90,0,0,17.15,90,0,0,
uz=27.15,90,0,0,7.15,90,0,0,17.15,90,0,0,
uq=0,0,0,90,0,0,0,90,0,0,0,90,
alim=366,0,90,90,16,0,90,90,26,0,90,90,
b1=1,1.8,2,1.8,1,1.8,2,1.8,1,1.8,2,1.8,
b2=0,.4,0,.4,0,.4,0,.4,0,.4,0,.4,b3=15*1,
a=1,2,.0141,2,1,2,.0141,2,1,2,.0141,2,nseg=12,nplot=1,
b=1,2,.0141,2,1,2,.0141,2,1,2,.0141,2,npoint=360,
psi=0,51,-39,51,0,61,-29,61,0,71,-19,71,nbreak=12,
yt=0,-1.154,.409,-1.134,0,-.694,1.062,-.675,0,-.213,1.683,-.195,
zt=0,2.549,3.796,2.547,0,2.711,3.668,2.705,0,2.790,3.428,2.788,
&end
```

INPUT DATA SET
FIGURE 9A

```
&lista ncount=1,xl=0,10,ns=1,anx=60,npoint=30,nseg=1,1,
ang=30*360,alim=30*360,b1=30*1,b3=30*1,am=30*1,an=30*1,
a=30*1,b=30*1,aa=30*1,bb=30*1,nblock=1,a1=1,a3=1,a5=1,
yt=30*1000,zt=30*0,cz1=1,cz3=1,cz5=1,
cy1=1,cy2=.01,cy3=.5,cy4=-.01,cy5=3,
&end
```

INPUT DATA SET
FIGURE 9B

```
&lista ncount=1,xl=0,10,ns=1,anx=60,npoint=30,nseg=1,1,
ang=30*360,alim=30*360,b1=30*1,b3=30*1,am=30*1,an=30*1,
a=30*1,b=30*1,aa=30*1,bb=30*1,nblock=1,a1=1,a3=1,a5=1,
yt=30*1000,zt=30*0,cz1=1,cz2=2,cz3=2,cz4=0,cz5=1,
cy1=1,cy2=.01,cy3=.5,cy4=-.01,cy5=3,
&end
```

```
INPUT DATA SET
FIGURE 9C

&lista ncount=1,xl=0,10,ns=1,anx=60,npoint=30,nseg=1,1,
ang=30*360,alim=30*360,b1=30*1,b3=30*1,am=30*1,an=30*1,
a=30*1,b=30*1,aa=30*1,bb=30*1,nblock=1,a1=1,a3=1,a5=1,
yt=30*0,zt=30*0,cz1=4,cz2=-3,cz3=.5,cz4=0,cz5=1,
cy1=1,cy2=3,cy3=2,cy4=0,cy5=1,
&end


INPUT DATA SET FIGURE 10

&lista ncount=2,nblock=1,2,xl=2,4,4,6,ns=1,1,anx=50,50,npoint=120,120,
ang=4*90,11*0,90,45,45,90,45,45,9*0,90,45,45,90,45,45,9*0,4*90,
alim=90,180,270,360,11*0,90,135,180,270,315,360,9*0,
90,135,180,270,315,360,9*0,90,180,270,360,11*0,
uy=0,90,180,270,11*0,0,90,135,180,270,315,9*0,
0,90,135,180,270,315,9*0,90,180,270,11*0,
uz=0,90,180,270,11*0,0,90,135,180,270,315,9*0,
0,90,135,180,270,315,9*0,90,180,270,11*0,
nseg=4,6,6,4,b1=60*2,b3=60*1,am=60*1,an=60*1,aj=60*1,
a=60*1,b=60*1,aa=60*1,bb=60*1,a1=1,1,a3=1,1,a5=1,1,
c1=1,1,c3=1,1,c5=1,1,list3=0,psi=60*45,h=60*1,
hf1=30*-1,0,1,1,0,1,1,hf2=30*0,0,0,1,0,1,0,
hf5=30*1,1,1,1,1,1,1,hf6=30*2,hf7=30*1,1,1,1,1,1,1,
hf9=30*1,1,1,1,1,1,1,hb1=0,-1,-1,0,-1,-1,9*0,15*0,15*0,
hb2=0,0,1,0,1,0,hb5=1,1,1,1,1,1,9*0,15*0,15*2,
hb7=1,1,1,1,1,1,9*0,15*0,15*1,hb9=1,1,1,1,1,1,9*0,15*0,15*1,
ngroup=1,1,1,4,0,2,1,2,1,0,2,2,2,2,0,2,3,2,3,0,2,4,2,4,0,
2,5,2,5,0,2,6,2,6,0,3,1,3,1,0,3,2,3,2,0,3,3,3,3,0,3,4,3,4,0,
3,5,3,5,0,3,6,3,6,0,4,1,4,4,0,nwi=1,ntop=1,1,
&end

INPUT DATA SET FIGURE 12

&lista
ncount=2,
xl=0,1.732,1.732,1.882,2.032,
ns=1,2,
anx=20,20,
```

```
npoint=240,240,
nseg=1,1,9,12,9,
ang=360,14*0,360,14*0,78.,45,45,6,45,45,6,45,45,6*0,
78.,45,0,45,6,45,0,45,6,45,0,45,3*0,
78.,45,45,6,45,45,6,45,45,6*0,
uy=15*0,15*0,20,90,0,0,90,0,10,90,0,6*0,
27.15,90,0,0,7.15,90,0,0,17.15,90,0,0,3*0,
32,90,0,12,90,0,22,90,0,6*0,
uz=15*0,15*0,20,90,0,0,90,0,10,90,0,6*0,
27.15,90,0,0,7.15,90,0,0,17.15,90,0,0,3*0,
32,90,0,12,90,0,22,90,0,6*0,
alim=360,14*0,360,14*0,360,0,90,10,0,90,20,0,90,6*0,
366,0,90,90,16,0,90,90,26,0,90,90,3*0,
372,0,90,22,0,90,32,0,90,6*0,
b1=1,14*0,1,14*0,1,1.8,1.8,1,1.8,1.8,1,1.8,1.8,6*0,
1,1.8,2,1.8,1,1.8,2,1.8,1,1.8,2,1.8,3*0,
1,1.8,1.8,1,1.8,1.8,1,1.8,1.8,6*0,
b2=15*0,15*0,0,.4,.4,0,.4,.4,0,.4,.4,6*0,
0,.4,0,.4,0,.4,0,.4,0,.4,0,.4,3*0,
0,.4,.4,0,.4,.4,0,.4,.4,6*0,
b3=75*1,
am=30*1,1,-1,1,1,-1,1,1,-1,1,6*1,
1,-1,1,1,1,-1,1,1,1,-1,1,1,3*1,
1,-1,1,1,-1,1,1,-1,1,6*1,
an=30*1,1,-1,1,1,-1,1,1,-1,1,6*1,
1,-1,1,1,1,-1,1,1,1,-1,1,1,3*1,
1,-1,1,1,-1,1,1,-1,1,6*1,
aj=30*1,1,1,-1,1,1,-1,1,1,-1,6*0,
1,1,1,-1,1,1,1,-1,1,1,1,-1,3*0,
1,1,-1,1,1,-1,1,1,-1,6*0,
a=1,14*0,1,14*0,1,2,2,1,2,2,1,2,2,6*0,
1,2,.0141,2,1,2,.0141,2,1,2,.0141,2,3*0,
1,2,2,1,2,2,1,2,2,6*0,
b=1,14*0,1,14*0,1,2,2,1,2,2,1,2,2,6*0,
1,2,.0141,2,1,2,.0141,2,1,2,.0141,2,3*0,
1,2,2,1,2,2,1,2,2,6*0,
aa=75*1,
bb=75*1,
psi=15*0,15*0,0,45,45,0,55,55,0,65,65,6*0,
0,51,-39,51,0,61,-29,61,0,71,-19,71,3*0,
0,57,57,0,67,67,0,77,77,
yt=15*0,15*0,0,-1.414,-1.414,0,-.9734,-.9734,0,-.5032,-.5032,6*0,
```

```
0,-1.154,.409,-1.134,0,-.694,1.062,-.675,0,-.213,1.683,-.195,3*0,
0,-.881,-.881,0,-.406,-.406,0,.080,.080,
zt=15*0,15*0,0,2.414,2.414,0,2.623,2.623,0,2.752,2.752,6*0,
0,2.549,3.796,2.547,0,2.711,3.668,2.705,0,2.790,3.428,2.788,3*0,
0,2.655,2.655,0,2.768,2.768,0,2.797,2.797,0,
uq=15*0,15*0,0,0,90,0,0,90,0,0,90,6*0,
0,0,0,90,0,0,0,90,0,0,0,90,3*0,
0,0,90,0,0,90,0,0,90,6*0,
nblock=1,2,
list3=0,
a1=2,2,2,
a2=0,0,0,
a3=1,1,1,
a4=0,0,0,
a5=1,1,1,
c1=0,1,
c2=1,0,
c3=1,1,
c4=0,0,
c5=1,1,
&end
```

INPUT DATA SET FIGURE 14

```
&lista
ncount=5,nblock=1,2,3,4,5,ns=2,2,1,3,2,
xl=1.5,2,3.9,3.9,4.07,4.36,4.36,5.46,5.46,5.96,6.33,6.86,6.86,7.1,7.27,
anx=42,8,19,24,7,npoint=60,180,60,120,180,
nseg=1,1,1,4,6,4,1,1,8,8,8,8,12,8,12,
ang=360,14*0,360,14*0,360,14*0,
60,120,60,120,11*0,6*60,9*0,60,120,60,120,11*0,
360,14*0,360,14*0,
240,6,18,18,36,18,18,6,7*0,240,6,18,18,36,18,18,6,7*0,
240,6,18,18,36,18,18,6,7*0,240,6,18,18,36,18,18,6,7*0,
12*30,3*0,30,60,30,60,30,60,30,60,7*0,12*30,3*0,
alim=360,14*0,360,14*0,360,14*0,
179,270,361,90,11*0,173.5,90,270,360,360,90,9*0,180,270,360,90,11*0,
360,14*0,360,14*0,
270,180,180,180,304.5,360,90,360,7*0,235.53,180,180,180,304.5,360,90,
360,7*0,235.53,235.6,235.7,235.8,304.1,304.2,304.3,304.47,7*0,
235.53,235.6,235.7,235.8,304.1,304.2,304.3,304.47,7*0,
```

```
90,360,270,180,90,0,270,180,90,360,270,180,3*0,87.45,180,177.45,270,
267.45,360,357.45,90,7*0,90,360,270,180,90,0,270,180,90,360,270,
180,3*0,
uy=15*0,15*0,15*0,
1,0,179,-180,11*0,6.5,0,90,180,180,0,9*0,0,0,180,-180,11*0,
0,14*0,0,14*0,
-90,90,90,90,235.5,270,0,270,7*0,-55.53,90,90,90,235.5,270,0,270,7*0,
-55.53,235.53,235.6,235.7,235.8,304.1,304.2,304.3,7*0,
-55.53,235.53,235.6,235.7,235.8,304.1,304.2,304.3,7*0,
0,270,360,90,0,90,180,90,180,270,180,270,3*0,2.55,-90,92.55,0,
182.55,90,272.55,-180,7*0,0,270,360,90,0,90,180,90,180,270,180,
270,3*0,
uz=15*0,15*0,15*0,
1,0,179,-180,11*0,6.5,0,90,180,180,0,9*0,0,0,180,-180,11*0,
0,14*0,0,14*0,
-90,90,90,90,235.5,270,0,270,7*0,-55.53,90,90,90,235.5,270,0,270,7*0,
-55.53,235.53,235.6,235.7,235.8,304.1,304.2,304.3,7*0,
-55.53,235.53,235.6,235.7,235.8,304.1,304.2,304.3,7*0,
0,270,360,90,0,90,180,90,180,270,180,270,3*0,2.55,-90,92.55,0,182.55,
90,272.55,-180,7*0,0,270,360,90,0,90,180,90,180,270,180,270,3*0,
b1=1,14*0,1,14*0,1,14*0,
1,2,1,2,11*0,1,2,2,1,2,2,9*0,1,2,1,2,11*0,
1,14*0,1,14*0,
1,2,1,2,1,2,1,2,7*0,1,2,1,2,1,2,1,2,7*0,8*1,7*0,8*1,7*0,
1,2,2,1,2,2,1,2,2,1,2,2,3*0,1,2,1,2,1,2,1,2,7*0,1,2,2,1,2,2,1,2,2,
1,2,2,3*0,
b2=15*0,15*0,15*0,
15*0,15*0,15*0,
15*0,15*0,
15*0,15*0,15*0,15*0,
15*0,15*0,15*0,
b3=1,14*0,1,14*0,1,14*0,
1,1,1,1,11*0,6*1,9*0,1,1,1,1,11*0,
1,14*0,1,14*0,
8*1,7*0,8*1,7*0,8*1,7*0,8*1,7*0,
12*1,3*0,8*1,7*0,12*1,3*0,
am=1,14*0,1,14*0,1,14*0,
1,1,1,1,11*0,6*1,9*0,1,1,1,1,11*0,
1,14*0,1,14*0,
1,1,3,1,1,1,3,1,7*0,1,1,3,1,1,1,3,1,7*0,8*1,7*0,8*1,7*0,
12*1,3*0,8*1,7*0,12*1,3*0,
an=1,14*0,1,14*0,1,14*0,
```

```
1,1,1,1,11*0,6*1,9*0,1,1,1,1,11*0,
1,14*0,1,14*0,
8*1,7*0,8*1,7*0,8*1,7*0,8*1,7*0,
12*1,3*0,8*1,7*0,12*1,3*0,
aj=1,14*0,1,14*0,1,14*0,
1,1,1,1,11*0,6*1,9*0,1,1,1,1,11*0,
1,14*0,1,14*0,
8*1,7*0,8*1,7*0,8*1,7*0,8*1,7*0,
12*1,3*0,8*1,7*0,12*1,3*0,
a=1,14*0,1,14*0,1,14*0,
.265,1,.265,1,11*0,1,.735,1,1,1,.735,9*0,1,1,1,1,11*0,
.265,14*0,.265,14*0,
.265,.0354,-.015,.0707,.265,.0707,.015,.0354,7*0,.265,.0001,-.0001,
.0707,.265,.0707,.0001,.0001,7*0,8*.265,7*0,8*.217,7*0,
.217,.238,.238,.217,.238,.238,.217,.238,.238,.217,.238,.238,3*0,
.169,1,.169,1,.169,1,.169,1,7*0,.12,.307,.307,.12,.307,.307,
.12,.307,.307,.12,.307,.307,3*0,
b=1,14*0,1,14*0,1,14*0,
.265,1,.265,1,11*0,1,.735,1,1,1,.735,9*0,1,1,1,1,11*0,
.265,14*0,.265,14*0,
.265,.0354,.15,.0707,.265,.0707,.15,.0354,7*0,.265,.0001,.0001,.0707,
.265,.0707,.0001,.0001,7*0,8*.265,7*0,8*.217,7*0,
.217,.238,.238,.217,.238,.238,.217,.238,.238,.217,.238,.238,3*0,
.169,1,.169,1,.169,1,.169,1,7*0,.12,.307,.307,.12,.307,.307,.12,
.307,.307,.12,.307,.307,3*0,
aa=.0001,14*0,.265,14*0,.265,14*0,
1,.0001,1,.0001,11*0,.265,1,.0071,.265,.0071,1,9*0,.265,.0001,.265,
.0001,11*0,
1,14*0,1,14*0,
8*1,7*0,8*1,7*0,8*1,7*0,8*1,7*0,
12*1,3*0,1,.272,1,.0106,1,.272,1,.0106,7*0,12*1,3*0,
bb=.0001,14*0,.265,14*0,.265,14*0,
1,.636,1,.636,11*0,.265,1,.735,.265,.735,1,9*0,.265,.735,.265,.735,
11*0,
1,14*0,1,14*0,
8*1,7*0,8*1,7*0,8*1,7*0,8*1,7*0,
12*1,3*0,1,.0106,1,.272,1,.0106,1,.272,7*0,12*1,3*0,
psi=15*0,15*0,15*0,
0,45,0,45,11*0,0,46.1,45,0,45,43.9,9*0,0,45,0,45,11*0,
0,14*0,0,14*0,
0,45,0,45,0,45,0,45,7*0,0,45,0,45,0,45,0,45,7*0,8*0,7*0,8*0,7*0,
0,45,45,0,45,45,0,45,45,0,45,45,3*0,0,45,0,45,0,45,0,45,7*0,
```

```
0,45,45,0,45,45,0,45,45,0,45,45,3*0,
zt=15*0,15*0,15*0,
0,-.715,0,.715,11*0,0,-.775,-.785,0,.785,.775,9*0,0,-.785,0,.785,11*0,
0,14*0,0,14*0,
0,.025,0,-.1,0,.1,0,-.025,7*0,0,-.15,-.15,-.1,0,.1,.15,.15,7*0,
8*0,7*0,8*0,7*0,
0,-.1685,-.1685,0,-.3855,-.3855,0,.1685,.1685,0,.3855,.3855,3*0,
0,0,0,-.362,0,0,0,.362,7*0,0,-.217,-.217,0,-.337,-.337,0,
.217,.217,0,.337,.337,3*0,
yt=15*0,15*0,15*0,
0,.005,0,.005,11*0,0,-.5,.005,0,.005,-.5,9*0,0,0,0,0,0,11*0,
0,14*0,0,14*0,
0,-.29,-.315,-.365,-.1965,-.365,-.315,-.29,7*0,0,-.218,-.218,-.268,
-.1,-.268,-.218,-.218,7*0,8*0,7*0,8*0,7*0,
0,.3885,.3885,0,-.1685,-.1685,0,-.3855,-.3855,0,.1685,.1686,3*0,
0,.362,0,0,0,-.362,0,0,7*0,0,.337,.337,0,-.217,-.217,0,
-.337,-.337,0,.217,.217,3*0,
a1=3,2,4,.5,2,.0001,2,.001,2,2,
a2=0,0,1,2.5,0,2,0,2,0,0,
a3=1,1,1,1,1,.2,1,.3,1,1,
a4=0,0,0,0,0,0,0,0,0,0,
a5=1,1,1,1,1,1,1,1,1,1,
c1=5*1,c3=5*1,c5=5*1,
hf1=180*0,0,.18,.18,0,0,0,0,.18,.18,0,0,0,3*0,0,.18,0,0,0,.180,0,
0,7*0,0,.18,.18,0,0,0,0,.18,.18,0,0,0,3*0,
hf2=180*0,0,1,1,0,0,0,0,0,0,0,0,0,0,3*0,0,1,0,0,0,0,0,0,7*0,0,1,
1,0,0,0,0,0,0,0,0,0,3*0,
hf3=45*0,0,.12,0,.12,11*0,0,.12,.12,0,.12,.12,9*0,
0,.12,0,.12,11*0,90*0,0,0,0,0,0,.18,.18,0,0,0,0,.18,.18,3*0,
0,0,0,.18,0,0,0,.18,7*0,0,0,0,0,0,.18,.18,0,0,0,0,.18,.18,3*0,
hf4=45*0,0,0,0,1,11*0,0,0,0,0,1,1,9*0,0,0,0,1,11*0,90*0,
0,0,0,0,0,0,0,0,0,0,1,1,3*0,0,0,0,0,0,0,0,1,7*0,
0,0,0,0,0,0,0,0,0,0,1,1,3*0,
hf5=45*2,2,2,2,2,11*2,2,2,2,2,2,2,9*2,2,2,2,2,11*2,90*0,
12*2,3*0,8*2,7*0,12*2,3*0,
hf7=225*1,
hf9=225*1,
hb3=45*0,0,-.08,0,-.08,11*0,0,-.08,-.08,0,-.08,-.08,9*0,
0,-.08,0,-.08,11*0,
hb4=45*0,0,0,0,1,11*0,0,0,0,0,1,1,9*0,0,0,0,1,11*0,
hb5=45*2,2,2,2,2,11*2,2,2,2,2,2,2,9*2,2,2,2,2,11*2,90*0,45*2,
hb7=225*1,
```

```
hb9=225*1,
h=225*1,
ngroup=4,1,4,1,0,4,2,4,2,0,4,3,4,3,0,4,4,4,4,0,5,1,5,1,0,5,2,5,2,0,
5,3,5,3,0,5,4,5,4,0,5,5,5,5,0,5,6,5,6,0,6,1,6,1,0,6,2,6,2,0,6,3,6,3,0,
6,4,6,4,0,13,1,13,1,0,13,2,13,2,0,13,3,13,3,0,13,4,13,4,0,
13,5,13,5,0,13,6,13,6,0,13,7,13,7,0,13,8,13,8,0,13,9,13,9,0,
13,10,13,10,0,13,11,13,11,0,13,12,13,12,0,14,1,14,1,0,14,2,14,2,0,
14,3,14,3,0,14,4,14,4,0,14,5,14,5,0,14,6,14,6,0,14,7,14,7,0,
14,8,14,8,0,15,1,15,1,0,15,2,15,2,0,15,3,15,3,0,15,4,15,4,0,
15,5,15,5,0,15,6,15,6,0,15,7,15,7,0,15,8,15,8,0,15,9,15,9,0,
15,10,15,10,0,15,11,15,11,0,15,12,15,12,0,
nwi=0,1,0,0,1,ntop=0,1,0,0,1,
&end
```

DISTRIBUTION

ADMINISTRATOR
DEFENSE TECHNICAL INFORMATION CENTER
ATTN DTIC-DDA (12 COPIES)
CAMERON STATION, BUILDING 5
ALEXANDRIA, VA 22304-6145

DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS
ATTN LIBRARY
WASHINGTON, DC 20234

DEPARTMENT OF COMMERCE
NATIONAL BUREAU OF STANDARDS
CENTER FOR RADIATION RESEARCH
WASHINGTON, DC 20234

DEPARTMENT OF ENERGY
ALBUQUERQUE OPERATIONS OFFICE
PO BOX 5400
ALBUQUERQUE, NM 87115

DIRECTOR
US ARMY BALLISTIC RESEARCH LABORATORY
ATTN ED DAVISSON
ATTN PAUL DEITZ,
SLCBR-VL-V
ATTN SLCBR-DD-T (STINFO)
ATTN SLCBR-VL, VULNERABILITY/
LETHALITY DIV
ATTN SLCBR-IB, INTERIOR BALLI DIV
ATTN SLCBR-LF, LAUNCH & FLIGHT DIV
ATTN SLCBR-TB, TERMINAL BALLISTICS DIV
ABERDEEN PROVING GROUND, MD 21005

US ARMY ELECTRONICS TECHNOLOGY
& DEVICES LABORATORY
ATTN SLCET-DD
FT MONMOUTH, NJ 07703

COMMANDER
US ARMY AVIATION SYSTEMS COMMAND
(AVSCOM)
US ARMY AVIATION RESEARCH &
DEVELOPMENT COMMAND
ATTN DRDAV-N, DIR FOR ADVANCED SYSTEMS
ATTN DRDAV-E, DIR FOR SYSTEMS
ENGINEERING & DEVELOPMENT
4300 GOODFELLOW RD
ST LOUIS, MO 63120

DIRECTOR
US ARMY MATERIEL SYSTEMS ANALYSIS
ACTIVITY
ATTN AMXSY-MP
ABERDEEN PROVING GROUND, MD 21005

COMMANDER
US ARMY MISSILE & MUNITIONS
CENTER & SCHOOL
ATTN ATSK-CTD-F
REDSTONE ARSENAL, AL 35809

DIRECTOR
US ARMY MISSILE LABORATORY
USAMICOM
ATTN DRSMI-RPR, REDSTONE SCIENTIFIC
INFO CENTER
ATTN DRSMI-RPT, TECHNICAL
INFORMATION DIV
REDSTONE ARSENAL, AL 35809

NAVAL WEAPONS CENTER
ATTN GRANT LABARRE
CODE 3381
CHINA LAKE, CA 93555

COMMANDER
NAVAL WEAPONS CENTER
ATTN 38, RESEARCH DEPT
ATTN 381, PHYSICS DIV
CHINA LAKE, CA 93555

DIRECTOR
NAVAL RESEARCH LABORATORY
ATTN 2600, TECHNICAL INFO DIV
WASHINGTON, DC 20375

SUPERINTENDENT
NAVAL POSTGRADUATE SCHOOL
ATTN LIBRARY, CODE 2124
MONTEREY, CA 93940

HQ, USAF/SAMI
WASHINGTON, DC 20330

AF AERO-PROPULSION LABORATORY
WRIGHT-PATTERSON AFB, OH 45433

SUPERINTENDENT
HQ, US AIR FORCE ACADEMY
ATTN TECH LIB
USAF ACADEMY, CO 80840

NASA LANGLEY RESEARCH CENTER
ATTN RAYMOND L. BARGER
ATTN MARY S. ADAMS
HAMPTON, VA 23665

DISTRIBUTION (cont'd)

DIRECTOR
NASA
GODDARD SPACE FLIGHT CENTER
ATTN 250, TECH INFO DIV
GREENBELT, MD 20771

DIRECTOR
NASA
ATTN TECHNICAL LIBRARY
JOHN F. KENNEDY SPACE
CENTER, FL 32899

DIRECTOR
NASA
LANGLEY RESEARCH CENTER
ATTN TECHNICAL LIBRARY
HAMPTON, VA 23665

DIRECTOR
NASA
LEWIS RESEARCH CENTER
ATTN TECHNICAL LIBRARY
CLEVELAND, OH 44135

AMES LABORATORY
DEPT OF ENERGY
IOWA STATE UNIVERSITY
ATTN ENVIRONMENTAL SCIENCES
ATTN HIGH ENERGY PHYSICS
ATTN EXPERIMENTAL NUCLEAR
ATTN SOLID STATE PHYSICS
ATTN NUCLEAR SCIENCES CATEGORY
AMES, IA 50011

BROOKHAVEN
DEPT OF ENERGY
ASSOCIATED UNIVERSITIES, INC
ATTN TECHNICAL INFORMATION DIV
ATTN PHYSICS DEPT, 5103
UPTON, LONG ISLAND, NY 11973

CALMA COMPANY, RESEARCH
AND DEVELOPMENT CENTER
ATTN GARY CROCKER
9805 SCRANTON RD
SAN DIEGO, CA 92121-1765

COMPUTER GRAPHICS LABORATORIES
CHRYSLER BUILDING
ATTN DICK LUNDIN
405 LEXINGTON AVE
NEW YORK, NY 10174

ENGINEERING SOCIETIES LIBRARY
ATTN ACQUISITIONS DEPARTMENT
345 EAST 47TH STREET
NEW YORK, NY 10017

EVANS AND SUTHERLAND INTERACTIVE
SYSTEMS DIV
ATTN ALYN P. ROCKWOOD
PO BOX 8700
580 ARAPEEN DR.
SALT LAKE CITY, UT 84108

GENERAL ELECTRIC, RESEARCH
AND DEVELOPMENT
ATTN W. P. WANG
PO BOX 43
SCHENECTADY, NY 12301

IBM THOMAS J. WATSON RESEARCH CENTER
ATTN RIDA T. FAROUKI
PO BOX 218
YORKTOWN HEIGHTS, NY 10598

INGRID B. CARLBOM
73 RESERVOIR STREET
BETHEL, CT 06801

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
4800 OAK GROVE DRIVE
ATTN TECHNICAL LIBRARY
PASADENA, CA 91103

LOS ALAMOS NATIONAL LAB
MAIL STOP D415
ATTN MELVIN L. PRUEITT
LOS ALAMOS, NM 87545

PIXAR
ATTN PAUL S. HECKBERT
PO BOX 13719
SAN RAFAEL, CA 94913

PDA ENGINEERING
ATTN MALCOLM S. CASALE
1560 BROOKHOLLOW DR.
SANTA ANA, CA 92705-5475

RENSSELAER POLYTECHNIC INSTITUTE
ECSE DEPT.
ATTN WM. RANDOLPH FRANKLIN
TROY, NY 12180

SANDIA LABORATORIES
LIVERMORE LABORATORY
PO BOX 969
LIVERMORE, CA 94550

SANDIA NATIONAL LABORATORIES
PO BOX 5800
ALBUQUERQUE, NM 87185

XEROX
PALO ALTO RESEARCH CENTER
ATTN RANDALL B. SMITH
3333 COYOTE HILL RD.
PALO ALTO, CA 94304

DIRECTOR
PROPULSION LABORATORY
RESEARCH & TECHNOLOGY LABORATORIES
AVSCOM
LEWIS RESEARCH CENTER, MS. 106-2
21000 BROOKPARK ROAD
CLEVELAND, OH 44135

DIRECTOR
RESEARCH & TECHNOLOGY
LABORATORIES (AVSCOM)
AMES RESEARCH CENTER
MOFFETT FIELD, CA 94035

NATIONAL OCEANIC & ATMOSPHERIC ADM
ENVIRONMENTAL RESEARCH LABORATORIES
ATTN LIBRARY, R-51, TECH REPORTS
BOULDER, CO 80302

ARIZONA STATE UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
ATTN GREGORY M. NIELSON
TEMPE, AZ 85287

BRIGHAM YOUNG UNIVERSITY
ATTN THOMAS W. SEDERBERG
PROVO, UT 84602

UNIVERSITY OF CALIFORNIA
SCHOOL OF ENGINEERING
LOS ANGELES, CA 90024

UNIVERSITY OF CALIFORNIA
DEPT. OF ELECTRICAL ENGINEERING
COMPUTER SCIENCE DIVISION
ATTN CARLO H. SEQUIN
BERKELEY, CA 94720

CALIFORNIA INSTITUTE OF TECHNOLOGY
SYNCHROTRON LABORATORY 206-49
ATTN JEFFREY GOLDSMITH
PASADENA, CALIFORNIA 91125

CALIFORNIA INSTITUTE OF TECHNOLOGY
COMPUTER SCIENCE (256-80)
ATTN ALAN H. BARR
PASADENA, CA 91125

CALIFORNIA STATE UNIVERSITY
SACRAMENTO
ATTN DWIGHT FREUND
SCIENCE BLDG., RM. 202
6000 J. STREET
SACRAMENTO, CA 95819

UNIVERSITY OF CALIFORNIA
BERKELEY COMPUTER GRAPHICS LABORATORY
COMPUTER SCIENCE DIVISION
ATTN BRIAN A. BARSKY
BERKELEY, CA 94720

CARNEGIE MELLON UNIVERSITY
DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
ATTN ZOLTAN CENDES
PITTSBURGH, PA 15213

CATHOLIC UNIVERSITY OF AMERICA
CARDINAL STATION
PO BOX 232
WASHINGTON, DC 20017

UNIVERSITY OF CONNECTICUT
DEPT. OF COMPUTER SCIENCE
AND ENGINEERING
U-155
ATTN JOHN A. ROULIER
260 GLENBROOK RD.
STORRS, CT 06268

UNIVERSITY OF ILLINOIS
ELEC AND COMP ENG DEPT.
ATTN S. W. LEE
ATTN ANDREW F. PETERSON
ATTN RAJ MITTRA
1406 WEST GREEN ST.
URBANA, ILL 61801

UNIVERSITY OF KANSAS
ATTN SWAPAN CHAKRABARTI
LAWRENCE, KS 66044

UNIVERSITY OF KANSAS
DEPT. OF COMPUTER SCIENCE
ATTN JAMES R. MILLER
LAWRENCE, KS 66045-2192

UNIVERSITY OF MICHIGAN
DEPT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
ATTN YUNG - CHIA LEE
ANN ARBOR, MI 48109-1109

UNIVERSITY OF NEBRASKA
ATTN EZEKIEL BAHAR
LINCOLN, NB 68588

NORTH CAROLINA STATE UNIVERSITY
DEPT OF COMPUTER SCIENCE
ATTN G. YATES FLETCHER
BOX 8206
RALEIGH, NC 27695-8206

DISTRIBUTION (cont'd)

UNIVERSITY OF NORTH CAROLINA
DEPT. OF COMPUTER SCIENCE
ATTN TURNER WHITTED
CHAPEL HILL, NC 27514

OHIO STATE UNIVERSITY
ELECTRO SCIENCE LAB
ATTN W. D. BURNSIDE
ATTN RONALD MARHEFKA
ATTN PRABHAKAR PATHAK
ATTN LEON PETERS
ATTN NAN WANG
1320 KINNEAR ROAD
COLUMBUS, OH 43212

UNIVERSITY OF ROCHESTER
DEPT. OF ELECTRICAL ENGINEERING
ATTN ARISTIDES A. G. REQUICHA
ROCHESTER, NY 14627

US ARMY LABORATORY COMMAND
ATTN TECHNICAL DIRECTOR, AMSLC-TD

INSTALLATION SUPPORT ACTIVITY
ATTN LEGAL OFFICE, SLCIS-CC

USAISC
ATTN RECORD COPY, ASNC-ADL-TS
ATTN TECHNICAL REPORTS BRANCH, ASNC-ADL-TR
(2 COPIES)

HARRY DIAMOND LABORATORIES
ATTN D/DIVISION DIRECTORS
ATTN LIBRARY, SLCIS-IM-TL (3 COPIES)
ATTN LIBRARY, SLCIS-IM-TL (WOODBRIDGE)
ATTN P. B. JOHNSON, SLCHD-ST-A
ATTN H. S. LESSER, SLCHD-IT-EB
ATTN H. C. LISINSKI, SLCHD-ST-AD
ATTN H. ROBERTS, SLCHD-ST-RC
ATTN M. J. VRABEL, SLCHD-ST-AD (20 COPIES)

DEPARTMENT OF THE ARMY
Harry Diamond Laboratories
2800 Powder Mill Rd
Adelphi, MD 20783-1197

An Equal Opportunity Employer

DEPARTMENT OF THE ARMY
HARRY DIAMOND LABORATORIES, ASNC-LAB-TR
2800 POWDER MILL ROAD
ADELPHI, MD 20783-1197

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE. $300

DA Label 18-1, Sep 83
Edition of Oct 74 will be used until exhausted

SUPERINTENDENT
NAVAL POSTGRADUATE SCHOOL
ATTN LIBRARY, CODE 2124
MONTEREY, CA 93940

HDL-TR-2140