

QUICK-GEOMETRY

A RAPID RESPONSE METHOD FOR MATHEMATICALLY MODELING CONFIGURATION GEOMETRY

Alfred F. Vachris, Jr., and Larry S. Yaeger

Grumman Aerospace Corporation

SUMMARY

This paper outlines the philosophy, the development and various applications of the QUICK-GEOMETRY system. This system provides a practical method for developing the geometry models that are essential to the operation of computer-based design and manufacturing systems. QUICK-GEOMETRY is part of Grumman's Rapid Aerospace Vehicle Evaluation System (RAVES).

4

NEED FOR GEOMETRY MODELING

Today, many efforts are focused on the design and operation of computerized systems for the development and production of all types of vehicles (aircraft, ships, automobiles and spacecraft). The results of these efforts are represented by computer-aided-design systems like ODIN, RAVES and IPAD (refs. 1, 2, and 3) and by computer-aided-manufacturing systems like CADAM (ref. 4).

The ability to model geometry is essential to the operation of these systems. From the systems standpoint, vehicle geometry must be modeled to establish a data base for geometry information. From the design standpoint, geometry must be modeled to develop the parametric models for conceptual design studies (ref. 5). From the analysis standpoint, configuration geometry must be carefully modeled to provide inputs for analysis programs (ref. 6). From the manufacturing standpoint, parts must be programmed so that they can be produced by numerically controlled machinery. In addition, surface geometry must be lofted to produce manufacturing templates (ref. 7).

RECENT EVOLUTION IN THE METHODS FOR MODELING GEOMETRY

A complete geometry modeling system would support the variety of geometry needs that are listed above. Today, there are a number of different geometry systems that cover some aspects of the problem (see, for example, refs. 8, 9, and 10) but no system is complete. The QUICK-GEOMETRY system was developed as the initial step toward a total geometry system.

The best way to describe the QUICK-GEOMETRY system is to contrast this system with the other systems now in use. Of particular interest, are the various methods for modeling surface geometry that are used by aerodynamic analysis codes. These different methods can be cataloged according to the various data structures that are used to define surface geometry (see fig. 1). In cataloging the various codes it was convenient to split the geometry modeling into two parts: (1) A description of the geometry of cross-section cuts at select fuselage stations and (2) a description of the surface geometry between cross-section cuts.

Point Geometry

The initial method for modeling geometry used a point geometry data structure that defined the configuration by tables of surface coordinates (refs. 11 through 14). Each table of surface coordinates is defined by digitizing a cross-section cut at a selected fuselage station. There are a number of difficulties with this approach: (1) Since a large number of surface points are required to develop this type of model, the process of measuring, cataloging and checking these surface coordinates becomes a very tedious and time consuming task. (2) Surface geometry is approximated by a large number of flat panels that are generated by connecting surface points from adjacent cross sections. The resulting surface is quite angular, making interpolation very difficult and providing only rough estimates for surface slopes and normals. (3) It is difficult to respond to design changes. Even local design changes require a careful review of how the original surface geometry was broken down into tables of points before the new geometry can be modeled. In many cases it is easier to remodel entire cross sections from the new drawing rather than modifying the earlier model. (4) The formats of point geometry models are very specific so that different analysis codes often require completely new models.

Fixed Pattern Geometry

Of all the problems mentioned, the difficulty with surface interpolation is the most critical one. To get around this problem a number of improvements were made to the point geometry method. To begin with, a rule or a pattern was followed when digitizing cross-section cuts so that a proper framework for surface interpolation could be established by sequentially connecting surface points from adjacent cross sections. Next, the data structure was expanded to provide for the definition of a number of different patterns and to allow the vehicle to be broken up into a series of modules, such that each module contains all the cross sections that could be fit by the same pattern. Finally, the digitizing pattern, which was built up from segments of lines, circles and ellipses, was developed analytically. Once the pattern had been established it was a simple matter to digitize the necessary points from the plan and profile views. Digitizing from body lines rather than from the cross-section cuts that are scattered all over the drawing generates a more consistent model.

The sum total of these improvements produced a new modeling technique centered on the use of a pattern rather than points to define cross sections (refs. 15 and 16). However, like the point geometry method, these patterns are still defined only at specific fuselage stations so that surface geometry must be defined by interpolation. And although surface interpolation can be carried out, there are problems because the interpolation techniques (spline fit and least squares fit) are not an integral part of the pattern.

Moving Pattern Geometry

Again, difficulties with defining surface geometry by interpolation lead to the next development in modeling. Instead of trying to improve interpolation techniques, a more comprehensive modeling system was developed. This system eliminated the need for surface interpolation by providing for the continuous definition of cross-section patterns along the length of the vehicle. In this system all the lines in the plan and profile view are individually modeled by either analytical or numerical (spline fit) techniques. The data structure then completes the model by analytically joining the cross-sections patterns to the body lines. In this way a cross-section pattern can be developed at each fuselage station (see refs. 17, 18 and 19).

The moving pattern technique seems like the correct way to model surface geometry. But the method must be organized into a proper geometry system before it can be evaluated. Since this method resulted from ad hoc modeling techniques to support aerodynamic computations, the codes that use this method have not set up geometry modeling systems.

The QUICK-GEOMETRY system will make an initial step toward the development of a total geometry system by organizing and developing the moving pattern into a proper geometry system.

OBJECTIVES

The overall objective is to develop the moving pattern into a geometry modeling system that could service the complete spectrum of geometry needs to support design, analysis, manufacturing and CAD-CAM.

To make sure that the QUICK-GEOMETRY system could be used as a design tool, a number of specific objectives were identified:

- The method should be capable of generating geometry models at all levels of configuration definition, starting with the initial "back of the envelope" sketch through the final three-view drawing. Then modeling could begin while the design was still taking shape, instead of having to wait until the three-view drawing was complete.

- The method must be capable of adding new detail to the current model as the design is developed. Then the model could be continually updated and so avoid having to remodel for each new phase of development.
- The method should be capable of synthesizing conceptual design models by selecting vehicle components from cataloged shapes and then scaling, rotating and translating them until they are properly positioned on the model. With this capability the initial three-view could then become a by-product of the model.

A second group of objectives were identified to make sure that the system could function as part of a CAD-CAM data bank. These objectives were:

- Common geometry - The model must serve as a central source of input geometry for a number of programs.
- Complete geometry - The model must integrate the various configuration components so that the vehicle is handled as a single unit.
- Computational geometry - The model must be defined by mathematical equations so that the surface coordinates and derivatives are calculated analytically.

GROUND RULES

Realizing that the development of a new geometry modeling system would require a significant effort, a number of ground rules were set up to organize the development of the computer code.

The primary ground rule was that the modeling system should be split into two separate operations. The first phase of the split operation is to define the geometry from the drawing and to collect all the modeling information into a concise model that could then be stored in a data bank. The second phase of the operation is to look up (calculate) surface geometry using the model defined in phase one. The overall organization of the split operation is shown in figure 2. Phase one is called QUICKDEF, while phase two is called QUICKLOK.

Because of the operational split the codes could then be developed and optimized for the separate functions to be performed. The QUICKDEF code must be very flexible to do a proper modeling job and so it requires a number of user-oriented conveniences to help make the program easy to use. These conveniences were built into the QUICKDEF code by first designing a self-explanatory data deck that could handle the job of modeling a difficult configuration, and then developing the subroutines to process and organize this input deck into a concise mathematical model. The definition code uses the cartesian coordinate system shown in figure 3.

On the other hand, the QUICKLOK code must be very automatic so that the surface geometry can be looked up without requiring that the user have specific knowledge of how a particular configuration has been modeled. The code must be very efficient so that it can be used frequently without a cost penalty. The look-up model uses a cylindrical coordinate system which properly supports the moving pattern geometry (fig. 3). The QUICKLOK code was designed around the use of two subroutines that were to provide the communication between the user's program and the math model.

The two subroutines are GEOMIN and CSGEOM. The first subroutine reads the math model into core and sets up the data structure.

To read the model:

```
CALL GEOMIN (IREAD, IRITE, ICRITE)
```

Where: IREAD - input unit
IRITE - write unit
ICRITE - write unit for error messages

The second subroutine is called each time a surface point is to be calculated. To calculate a surface point:

```
CALL CSGEOM (X, H, R, RX, RH, RXX, RXH, RHH, NDERV)
```

Where: X - fuselage station location (see fig. 3)
H - theta location ($-\pi/2 \leq \theta \leq +\pi/2$)
R - radial distance to surface point (X, H)
RX - DR/DX at the surface point
RH - DR/D θ at the surface point
RXX - D^2R/DX^2 at the surface point
RXH - $D^2R/(DXD\theta)$ at the surface point
RHH - $D^2R/D\theta^2$ at the surface point
NDERV - $\pm N$, where N is the order of the derivative to be calculated (0, 1, 2).
+ N, previous call to CSGEOM was at a different surface point. Set up the geometry at this station before computing derivatives.
- N, previous call to CSGEOM was at the same surface point. Calculate derivative without recalculating R.

The quantities X, H, and NDERV are specified by the user, the remaining values are computed by the QUICKLOK code.

As a result of the split operation a number of other codes have been added to the QUICK-GEOMETRY system. They include:

- QUICKCHK - operates the QUICKLOK code to check out the model. Outputs are tables and plots of surface coordinates and slopes.
- QUICKPLT - a general plotting package that will display the model geometry from different points of view and in different display formats.
- QUICKGEN - generates point geometry input models for other computer programs.

To get the modeling techniques started, a number of limitations were put on the geometry model. First of all, only the external surface of the configuration will be modeled. All the internal geometry is ignored. Secondly, the surfaces will be evaluated in polar coordinates so that the surfaces must therefore be expressed as a single valued function of theta. A moving axis, in the plane of symmetry, is provided to lessen the impact of this limitation (see fig. 4). And finally, the basic modeling elements that are used to build up the geometry of the configuration make use of only point and slope boundary conditions; hence, curvature is not matched from one element to another. (Note - Even with these intentional limitations, experience has shown that the QUICK-GEOMETRY system does a very good job of modeling geometry.)

To make sure that the system was not dependent on a particular hardware system, two other ground rules were adopted. First, the modeling system was developed so that it could operate without the use of any special equipment such as digitizers or scopes. Of course, the performance of the system will be improved by the availability of extra hardware. The final ground rule was that the code be developed in machine-independent FORTRAN code so that it could be put up on any system with a minimum of effort. There are no machine tricks in the QUICK-GEOMETRY system. Furthermore, the code was developed on IBM equipment using single precision for the most part, so there will be no precision problems when running on other computer systems. As a result of these ground rules, the use of the QUICK-GEOMETRY system is not limited to organizations having access to large-scale computer facilities.

UNIQUE FEATURES

The QUICK-GEOMETRY system supports a number of unique features that distinguish this geometry system from all others. These features include the data structure, a user-oriented language for geometry modeling and the formulation of a geometry catalog.

DATA STRUCTURE

The most important feature of the QUICK-GEOMETRY system is the data structure. The left-hand side of figure 5 shows the different levels of organization that are used in the data structure. The pattern or surface which has the most complex organization is at the top, leading down to the curve element which is the fundamental building block of the QUICK-GEOMETRY system. The chart in the center of the figure shows how the various parts of the data structure are used to model the configuration and it also provides a functional flow diagram for both the QUICKDEF and the QUICKLOK codes. The flow from top to bottom outlines the different steps taken to translate a drawing into a QUICK-GEOMETRY model, while the path from bottom to top outlines the different computational steps used to construct surface geometry from the math model.

Curve Element

The basic building block of the QUICK-GEOMETRY system is the curve element (see fig. 6). The curve element is similar to the standard lofting conic except that the shoulder point condition has been replaced by a direct selection of a shape equation.

The boundary conditions that are used to determine the coefficients of each shape equation are the coordinates of the origin point, the termination point and the slope control point. The slope control point is a very convenient way of specifying slope conditions in coordinate form and in particular, it allows for the simultaneous specification of a vertical tangent at one end point and a horizontal tangent at the other.

Although there are a variety of shape functions to choose from, it is easy to make the proper selection. Aside from the line and the cubic, all the shapes fall into either the x-parabola or the y-parabola family. The x- or y-prefix indicates that the axis of symmetry for that particular parabola is parallel to the x-axis or y-axis. The x- or y-prefix for the rotated parabola and for the ellipse is very important because it identifies the shape that the curve element will take when the c-coefficient in the shape equation has a zero value.

The shape selection is made by matching the geometry to be modeled to either the x- or y-parabola. If the segment to be modeled has a vertical slope along its length, then it belongs to the x-family; if it has a horizontal tangent, then it belongs to the y-family. If the segment has both a vertical and a horizontal tangent, then it must be modeled as an ellipse. Note that the x-ellipse and y-ellipse generate the same points provided that $c \neq 0$. If the segment is rotated so that its symmetry line is not parallel to either the x-axis or the y-axis, the segment must be modeled as a rotated parabola. Note that for intermediate slope conditions, both rotated parabolas could be defined for the same set of conditions and that each shape would generate a different curve.

Curves

The curve is the next level in the data structure. In the QUICK-GEOMETRY system each curve is the projection of a space curve into either the plan or profile view of a three-view drawing. Every curve is constructed by blending various curve elements together, with user control over matching point and slope continuity between segments (see fig. 7).

<u>TYPE</u>	<u>KEYWORD</u>	<u>FUNCTION</u>
Piece	PIECE	Curve is defined as a unit, with end points and slope control point if necessary.
Aft-Link	ALINK	Curve being defined begins at the end of the previous curve and is tangent to it.
Fore-Link	FLINK	Curve being defined ends at the beginning of the following curve and is tangent to it.
Patch	PATCH	Curve being defined begins at the end of the previous curve and ends at the beginning of the following curve and is tangent to both of the adjoining curves.
Fillet	FILET	End points and slopes of curve being defined are calculated from specified positions on the adjoining curves.

Note that the fillet, in addition to defining a new segment, also controls the domain or use of the curve segments that are adjacent to it.

Surface Element

The next level in the data structure is the surface element which is very similar to the curve element. At a fixed fuselage station the surface element would be defined exactly like the curve element by specifying the coordinates for the origin, the termination and the slope control points. However, as shown in figure 8, the surface element is not fixed at a particular fuselage station, but rather, is continuously defined along some portion of the fuselage. The boundary conditions for the surface element are therefore defined along curves specifying the variation of the origin, the termination and the slope control points.

At each fuselage station the coordinates of the boundary curves are used to calculate the coefficients in the shape equation defining each surface element. Currently, the shape equations for the surface elements are limited to the line and the ellipse. Note, however, that A^2 and B^2 are used as coefficients in the ellipse equation, instead of just A and B, therefore the "ellipse" equation can also define a hyperbola.

Since the shape equation coefficients are calculated continuously along the length of the fuselage, these coefficients are functions of x just like the boundary curves (x is the distance along the fuselage reference line aft of some reference point). The derivatives of these coefficients can also be calculated from the boundary curves. Once these derivatives are known, then the surface shape function can be differentiated to produce the set of surface derivatives shown in figure 8.

Surfaces

The surface or pattern is the top level of the QUICK-GEOMETRY data structure. The pattern is built up from surface elements in two ways: (1) using the same blending controls that were used to build up curves from curve elements (see fig. 7) and (2) using surface elements to control the use of adjacent surface elements.

The right-hand side of figure 9 illustrates the basic concept of the adapting or controlling surface element. The canopy and the fuselage are each defined as a separate surface element while the surface (canopy plus fuselage) is defined by specifying that the canopy and the fuselage are to control the use of each other. The basis of this control is the ability to calculate the intersections of adapting elements, and to then limit the use of each element, so that only external geometry is seen.

The left-hand side of the figure shows the fillet which is a special type of adapting surface element. The fillet is special because it is an optional surface element that can be added to a pattern that has already been modeled. It would be impractical to add a fillet to a pattern using only measurements from a drawing because the measurements cannot match the pattern coordinates which are computed to at least six significant figures. Adding tolerances to the program is not the answer because this will only lead to a tangled mess of logical tests.

A straightforward procedure was developed for adding a fillet. The fillet is defined by giving only one measured coordinate at each end point of the fillet (usually the y -coordinate at one end and the z -coordinate at the other). The program then calculates the missing coordinates from the pattern. Thus, the fillet will properly adhere to the pattern. For example, look at the fillet that is shown under the wing in figure 9. This fillet is defined by the z -coordinate of the inboard end point and by the y -coordinate of the outboard end point.

MODELING LANGUAGE

The QUICKDEF code uses a user-oriented language for translating a drawing into a math model. The basic elements of this language are the keywords that define shapes and blending control and a common input structure that defines both surface and curve elements. Some of these language features are shown in a sample input deck (see fig. 10).

The first six cards after the title card define the single pattern that is used for this model. The pattern shown here has two surface elements: the lower body (BDYLOWER) and the upper body (BDYUPPER). Each surface element is defined by its shape keyword, its blending keyword and the names of its origin, termination and slope control curves. All the names (excluding the shape and blending keywords) are chosen by the user. Selecting functional or mnemonic names will make it easier to follow and model the various lines shown on a three-view drawing.

The next portion of the data deck (starting with the card showing YBDYBOT) models the plan and profile views of the lines that define the surface elements. The particular curve to be modeled is identified by prefixing a y (for plan view) or a z (for profile view) to the name of the control line. The next two cards define one of the curve elements that will build up the curve. The first card specifies the keywords for shape and blending. The second card defines the coordinates for the origin, termination and the slope control points. Depending on the blending control, this data may be provided by pointing to another curve element rather than typing in the coordinate values. A card with a minus one signals the completion of input for the curve being modeled.

The last part of the data deck shows the alias cards which set up a way of sharing curve models instead of having to model the same curve twice. For example: the plan projection of the body top center line (YBDYTOP) and the plan projection of the body bottom center line (YBDYBOT) are the same line on the drawing so they are aliased to share the same curve model. Aliases are especially useful for defining the slope control lines for surface elements. These slope control lines usually specify either a horizontal or a vertical tangent boundary condition at one end point of the surface element. A horizontal tangent is set by aliasing the z-coordinate of the slope control line to the z-coordinate of the end point line and a vertical tangent is set by aliasing the y-coordinate of the slope control line to the y-coordinate of the end point line.

GEOMETRY CATALOG

The flexibility of the QUICK-GEOMETRY data structure and the modeling language makes it possible to collect and catalog the various patterns to form a library of configurations.

Figure 11 illustrates the flexibility of the pattern. This is the most elementary pattern consisting of only two elliptical surface elements. Once the pattern has been defined the actual shape that it will take at any fuselage station depends upon the relative locations of the boundary curves. All the shapes that are displayed come from the same pattern.

Figure 12 shows how the pattern can have a flexible structure, as well as a flexible shape. The adapting surface elements can be joined to build an adapting pattern that will change its external structure depending upon what components are visible. This can be accomplished, for example, by defining the canopy and the wing to be inside the fuselage at stations before they actually appear on the configuration and setting up the mutual use-domain controls among the fuselage, canopy and wing. Then the structure of the pattern will change automatically as canopy and wing appear and then disappear.

Selecting the correct patterns from the library is easy to do since the pattern is described completely in words.

APPLICATIONS

The QUICK-GEOMETRY system is being applied in a number of ways. As part of RAVES it is used as a common source of vehicle geometry which is drawn upon to generate a variety of geometry input data decks (ref. 20). In particular, data decks conforming to the Harris input format (ref. 11), which is a standard at NASA Langley, can be generated. In another RAVES application, QUICK-GEOMETRY has been adapted to a vehicle lofting system (ref. 21). The most sophisticated application to date, has been to integrate the QUICK-GEOMETRY system with a numerical flow code which calculates the steady super/hypersonic inviscid flow around real configurations (ref. 22). In this application vehicle geometry (surface coordinates, slopes and normals) are generated as required by the flow analysis code. A small cross section of models are shown in figure 13.

CONCLUSIONS

The QUICK-GEOMETRY system has developed the moving pattern method into a practical method for modeling geometry. It therefore provides the initial step toward a complete modeling system that can support design, analysis, manufacturing and CAD-CAM.

WHAT IS NEXT ...

The data structure will be expanded:

- (1) To model multiple external and internal surfaces.
- (2) To define wings, fins and tails using natural aerodynamic coordinates (buttline and waterline cuts).
- (3) To define body lines by word patterns.

The design synthesis capability will be developed. Figure 14 shows the early results of a design effort based on the lofting program (ref. 21). The model shown was composed in a half hour using an IBM 2250.

And finally, an interface will be developed to join a surface patch technique to the moving pattern.

REFERENCES

1. Glatt, C. R., and Hague, D. S. : ODIN - Optimal Design Integration System. NASA CR-2492, February 1975.
2. Wennagel, G., Loshigian, H., and Rosenbaum, J.: RAVES - Rapid Aerospace Vehicle Evaluation System. ASME Winter Annual Meeting, Houston, Texas, 1975.
3. Heldenfels, R. R.: Integrated Computer-Aided Design of Aircraft. AGARD Conference Proceedings, No. 147, Vol. 1, 1973.
4. Feder, Aaron : Test Results on Computer Graphics Productivity for Aircraft Design and Fabrication. AIAA Paper 75-967, 1975.
5. NATO, Aircraft Design Integration and Optimization. AGARD Conference Proceedings, No. 147, Vol 1, 1973.
6. Giles, Gary L., Blackburn, Charles L., and Dixon, Sidney C.: Automated Procedures for Sizing Aerospace Vehicle Structures (SAVES). Journal of Aircraft, Vol 9, No. 12, December 1972, pp 812-819.
7. Bezier, P.: Numerical Control Mathematics and Applications. John Wiley and Sons, New York. 1970.
8. Forrest, A. R. : Computational Geometry Proceedings of the Royal Society of London, Vol 321A, pp 187-195, 1971.
9. Sutherland, I. E. : Three-Dimensional Data Input by Tablet. Proceedings of the IEEE, Vol 62, pp 453-461, April 1974.
10. Ahuja, D. V., and Coons, S. A.,: Geometry for Construction and Display. IBM System Journal, Vol 7, No. 3 and 4, 1968.
11. Craidon, Charlotte B. : Description of a Digital Computer Program for Airplane Configuration Plots. NASA TM X-2074, 1970.
12. Harris, Roy V., Jr.: An Analysis and Correlation of Aircraft Wave Drag. NASA TM X-947, 1964.

13. Hess, J. L. and Smith, A. M. O.: Calculation of Non-Lifting Potential Flow About Arbitrary Three-Dimensional Bodies. Douglas Aircraft Company, Rept. No. E.S. 40622, March 1962.
14. Woodward, F. A. : An Improved Method for the Aerodynamic Analysis of Wing-Body-Tail Configurations in Subsonic and Supersonic Flow. NASA CR-2228, May 1973.
15. Gentry, A. E., Smyth, D. N., and Oliver, W. R.: The Mark IV Supersonic-Hypersonic Arbitrary-Body Program. AFFDL-TR-73-159, Vol 1, November 1973.
16. Watman, H. and Meyer, R.: Grumman High Speed Aerodynamic Prediction Program (GAC-HAPP). Grumman Advanced Development Report ADR 01-03-74.3, March 1974.
17. De Jarnette, Fred R.: Surface Fitting Three-Dimensional Bodies. NASA CR-139663, 1974.
18. Gunness, R. C., Jr., Knight, C. J., D'Sylva, E.: Flowfield Analysis of Aircraft Configurations Using a Numerical Solution to the Three-Dimensional Unified Supersonic/Hypersonic Small-Disturbance Equations. NASA CR-1926, February 1972.
19. Kutler, P., Reinhardt, W. A., and Warming, R. R.: Multishocked, Three-Dimensional Supersonic Flowfields With Real Gas Effects. AIAA Journal, Vol 11, No. 5, pp 657-664, May 1973.
20. Vachris, A. : QUICK-GEOMETRY (A9), Grumman Aerospace Corporation. EG/RAVES-UM-210-75, July 1975.
21. Abramson, W. : Fuselage Modeling Program (C3), Grumman Aerospace Corporation. EG/RAVES-UM-102, July 1975.
22. Marconi, F., Yaeger, L., and Hamilton, H.: Computation of High-Speed Inviscid Flows About Real Configurations. NASA SP-347 (Part II), March 1975, pp 1412-1455.

DATA STRUCTURE	DEFINITION	CROSS-SECTION GEOMETRY	SURFACE GEOMETRY	COMPUTER CODES
POINT GEOMETRY	DIGITIZE CROSS SECTIONS	NUMERICAL APPROXIMATION	NUMERICAL APPROXIMATION	CRAIDON HARRIS HESS-SMITH WOODWARD
FIXED PATTERN GEOMETRY	DIGITIZE BODY LINES AT FIXED CROSS SECTIONS	ANALYTICAL COMPUTATION	NUMERICAL APPROXIMATION	GENTRY WATMAN
MOVING PATTERN GEOMETRY	DIGITIZE BODY LINES	ANALYTICAL COMPUTATION	ANALYTICAL COMPUTATION	DE JARNETTE GUNNESS KUTLER MARCONI

Figure 1.- Methods for defining geometry.

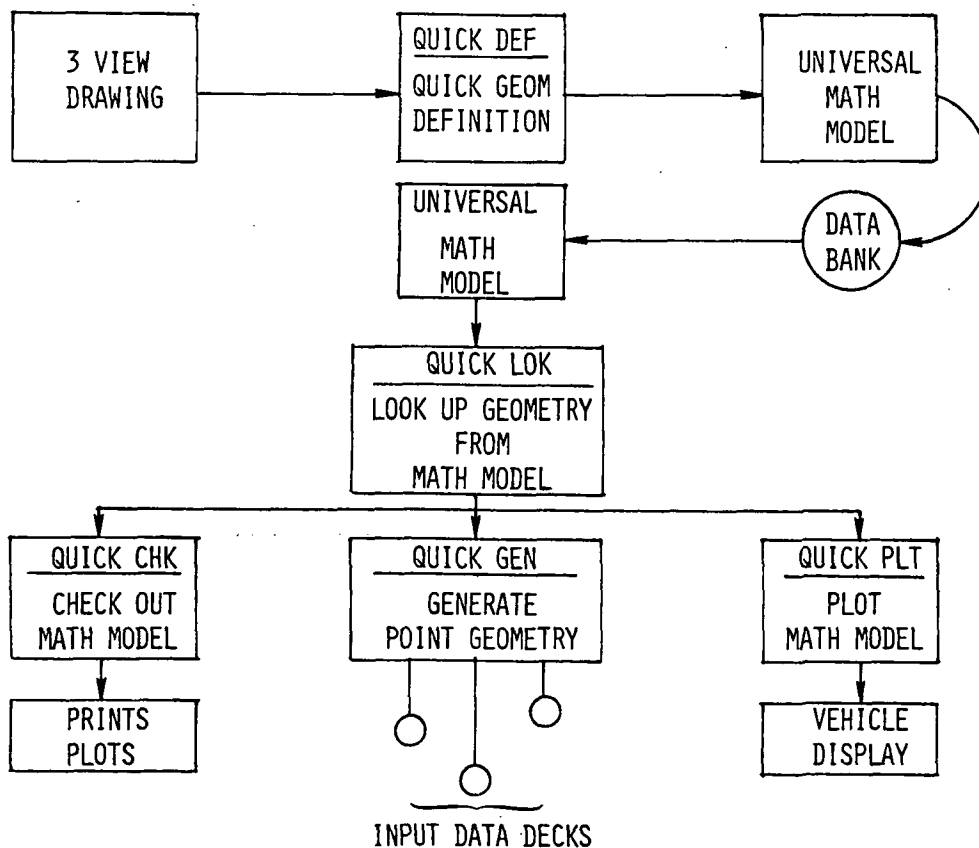


Figure 2.- Split operation.

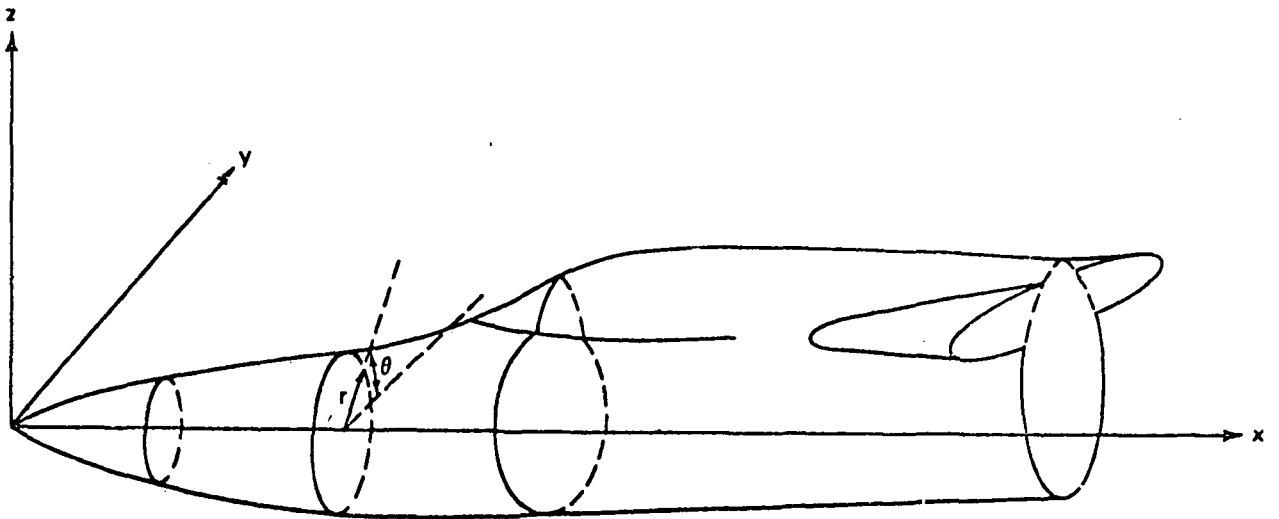


Figure 3.- Cartesian and cylindrical coordinate systems.

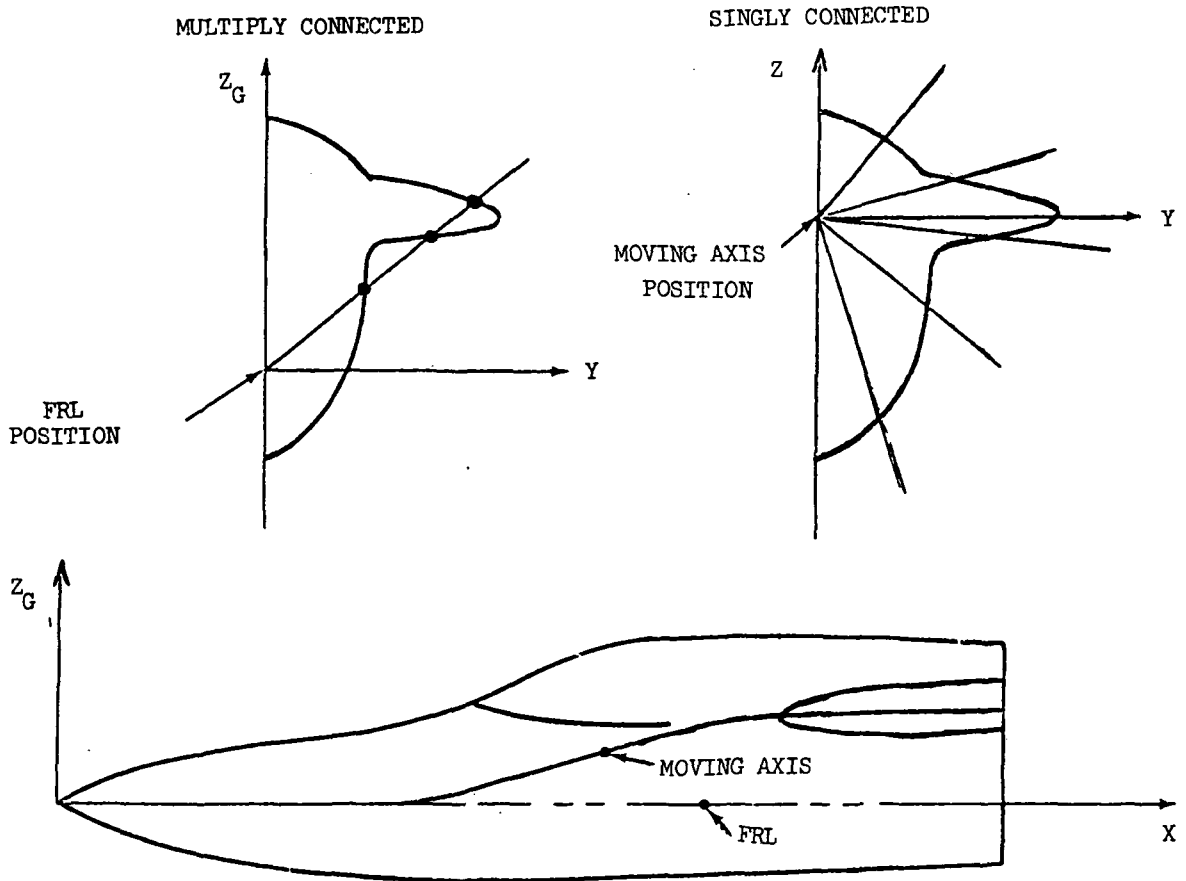


Figure 4.- FRL and moving axis.

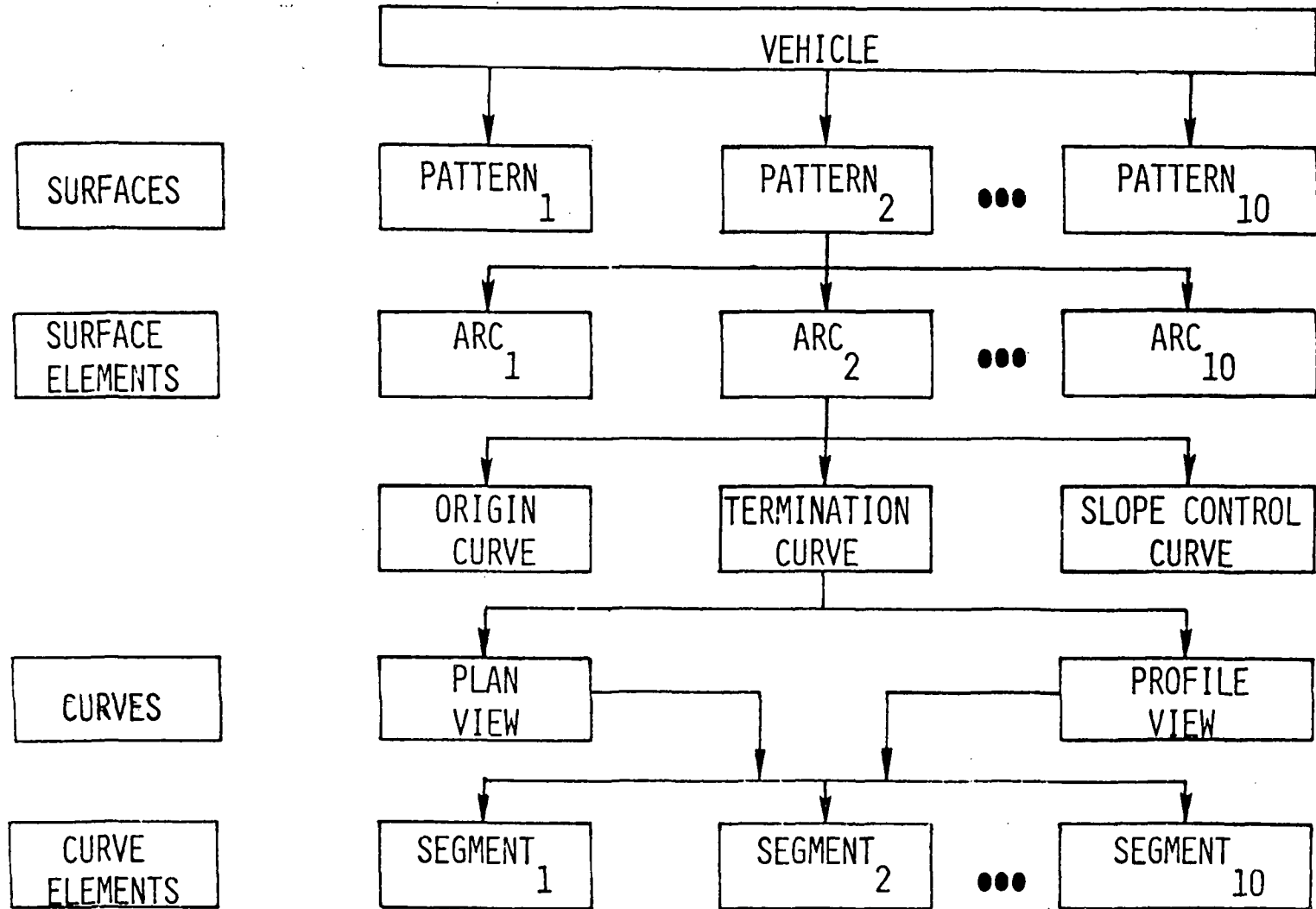
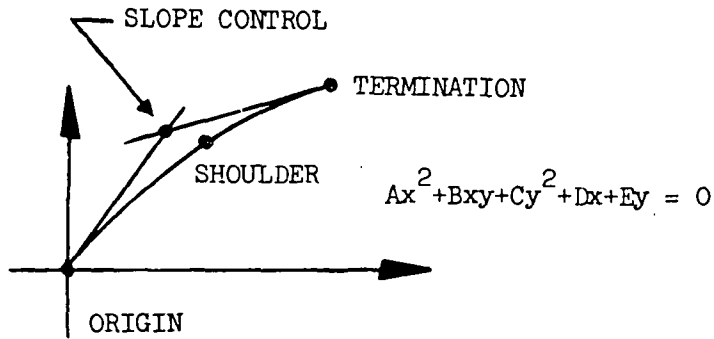
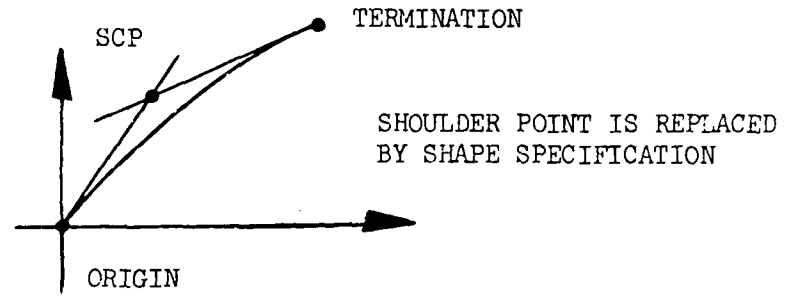


Figure 5.- Data structure.

STANDARD LOFTING CONIC



QUICK CURVE SEGMENT



<u>SHAPE</u>	<u>KEYWORD</u>	<u>EQUATION</u>	X-AXIS PARABOLA	ROTATED X-AXIS PARABOLA	X-ELLIPSE
Line	LINE	$Ax + By = 0$			
x-Parabola	XPAR	$Ax + By + y^2 = 0$			
y-Parabola	YPAR	$Ax + By + x^2 = 0$			
Rotated x-Parabola	RXPA	$Ax + By + Cxy + y^2 = 0$			
Rotated y-Parabola	RYPAR	$Ax + By + Cxy + x^2 = 0$			
x-Ellipse	ELLX	$Ax + By + Cx^2 + y^2 = 0$			
y-Ellipse	ELLY	$Ax + By + Cy^2 + x^2 = 0$			
Cubic	CUBI(C)	$Ax + By + Cx^2 + x^3 = 0$			

Figure 6.- Curve element definition.

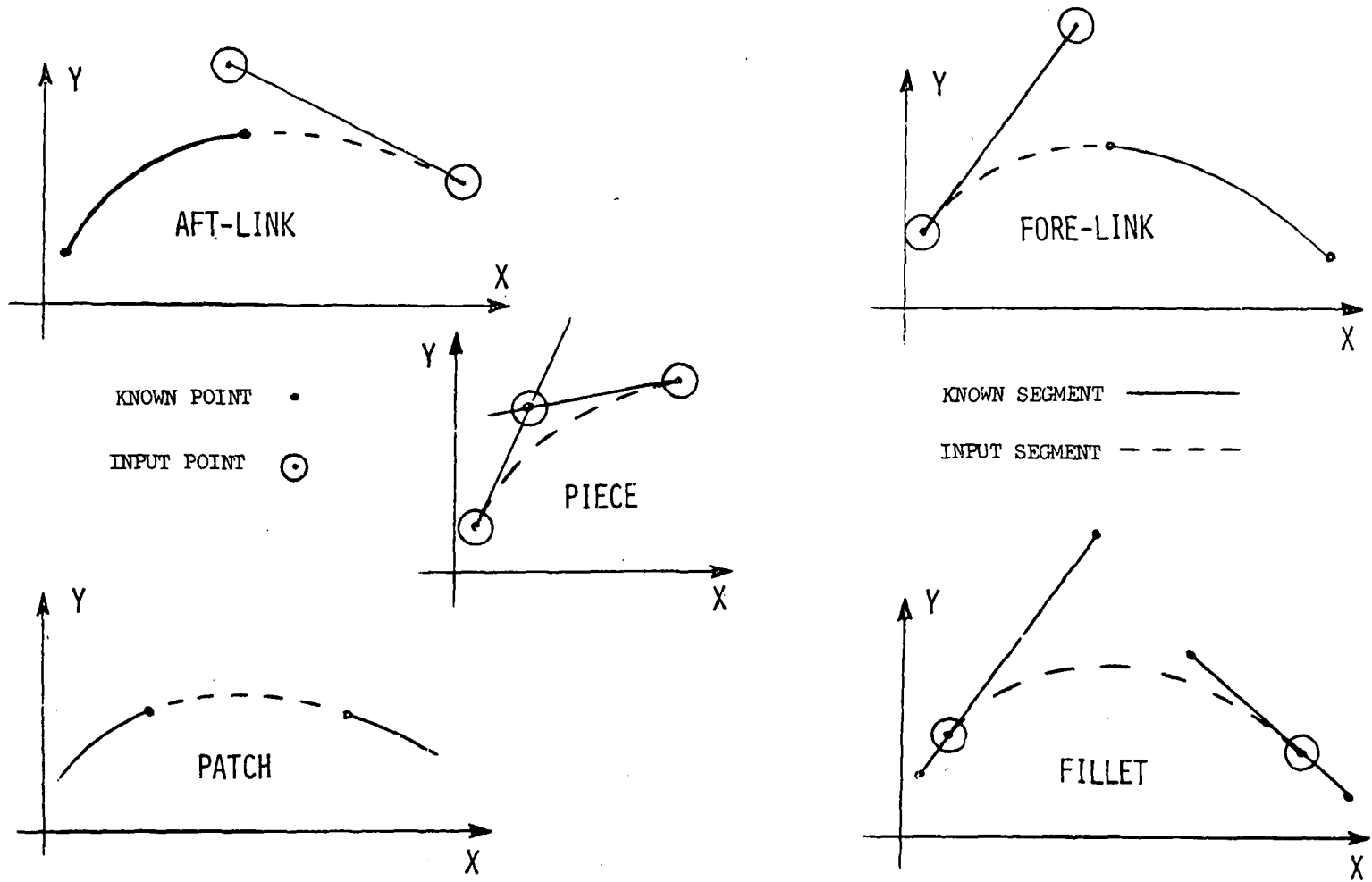
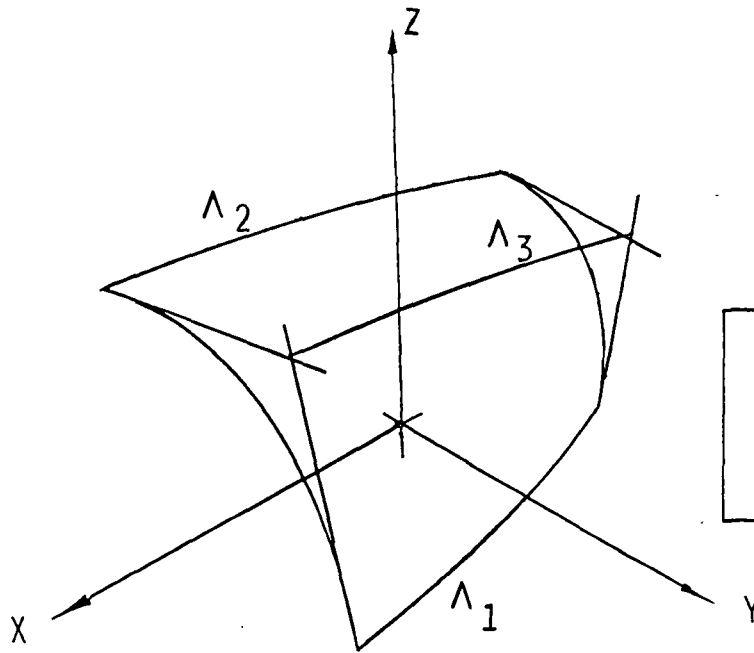


Figure 7.- Blending control for curve elements.



- Λ_1 - ORIGIN CURVE
- Λ_2 - TERMINATION CURVE
- Λ_3 - SLOPE CONTROL CURVE

TYPICAL SHAPE \rightarrow GROWING ELLIPSE

$$\frac{[Y(X) - Y_0(X)]^2}{A^2(X)} + \frac{[Z(X) - Z_0(X)]^2}{B^2(X)} = 1$$

POLAR FORM $Q(R, R_0, \theta, \theta_0, A^2, B^2) = 0$

$$B^2 (R \cos \theta - R_0 \cos \theta_0)^2 + A^2 (R \sin \theta - R_0 \sin \theta_0)^2 - A^2 B^2 = 0$$

WHERE $R_0 = R_0(X)$; $\theta_0 = \theta_0(X)$; $A^2 = A^2(X)$; $B^2 = B^2(X)$

Q IS DIFFERENTIABLE PRODUCING

$$\frac{DR}{DX}, \frac{DR}{D\theta}, \frac{D^2R}{DX^2}, \frac{D^2R}{DXD\theta}, \frac{D^2R}{D\theta^2}$$

Figure 8.- Surface element.

ADAPTING SURFACE ELEMENTS

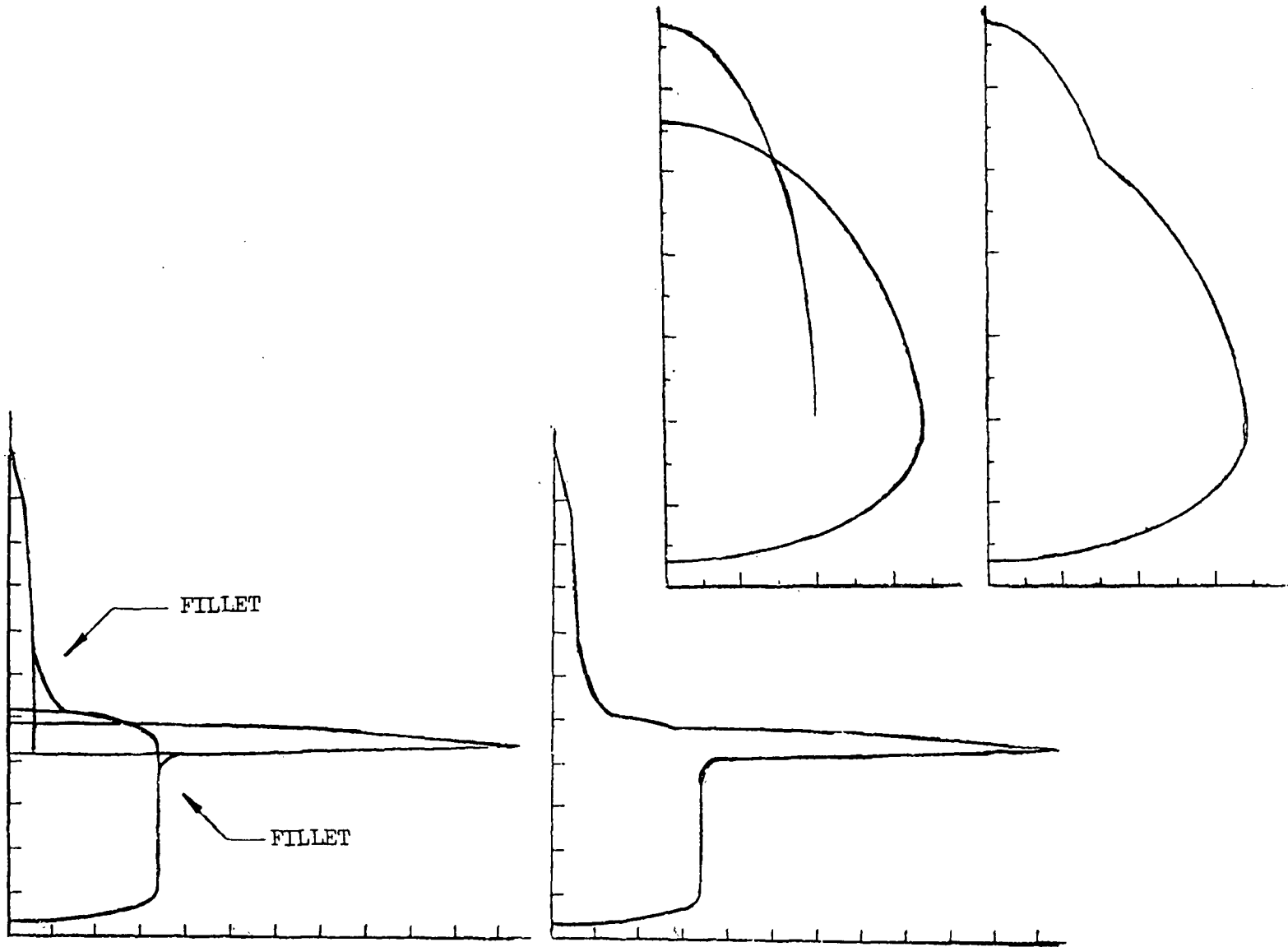
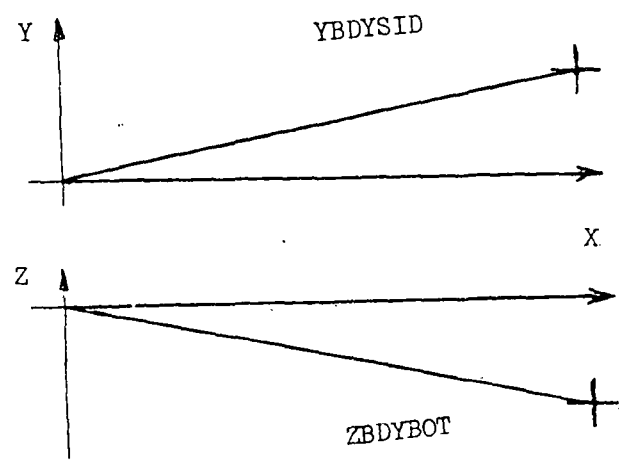
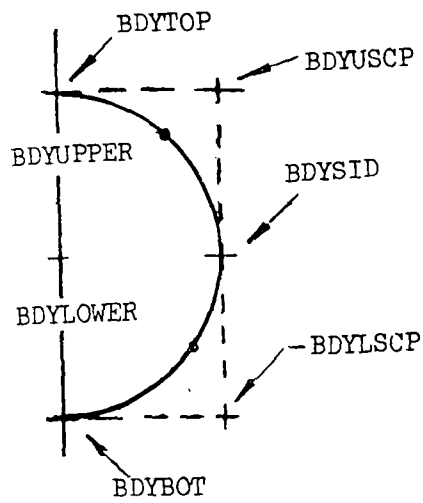


Figure 9.- Constructive geometry.



SCONE20: TWENTY DEGREE SHARP CONE

```

1
1 2
BDYLOWER 1ELLI  PIECE  BDYBOT  BDYSID  BDYLSCP
BDYUPPER 2ELLI  PIECE  BDYSID  BDYTOP  BDYUSCP
1
1 1  MAPAXIS
1 1  0. 20.
YBDYBOT
1 LINE  PIECE KV5
0. 0. 20. 0.
-1
ZBDYBOT
1 LINE  PIECE KV4
0. 0. 15. A-20.
3 LINE  PIECE KV5
10. -4.55 20. -4.55
2 RYPA  FILET KV0
1. 1. 3. 3. 10. 15.
-1
YBDYSID
1 LINE  PIECE KV4
0. 0. 15. A20.
3 LINE  PIECE KV5
10. 4.55 20. 4.55
2 RYPA  FILET KV0
1. 1. 3. 3. 10. 15.
-1
ZBDYSID  YBDYBOT
YBDYTOP  YBDYBOT
ZBDYTOP  YBDYSID
YBDYLSCP YBDYSID
ZBDYLSCP ZBDYBOT
YBDYUSCP YBDYSID
ZBDYUSCP ZBDYTOP
YMAPAXIS YBDYBOT
ZMAPAXIS YBDYBOT

```

Figure 10.- Sample input.

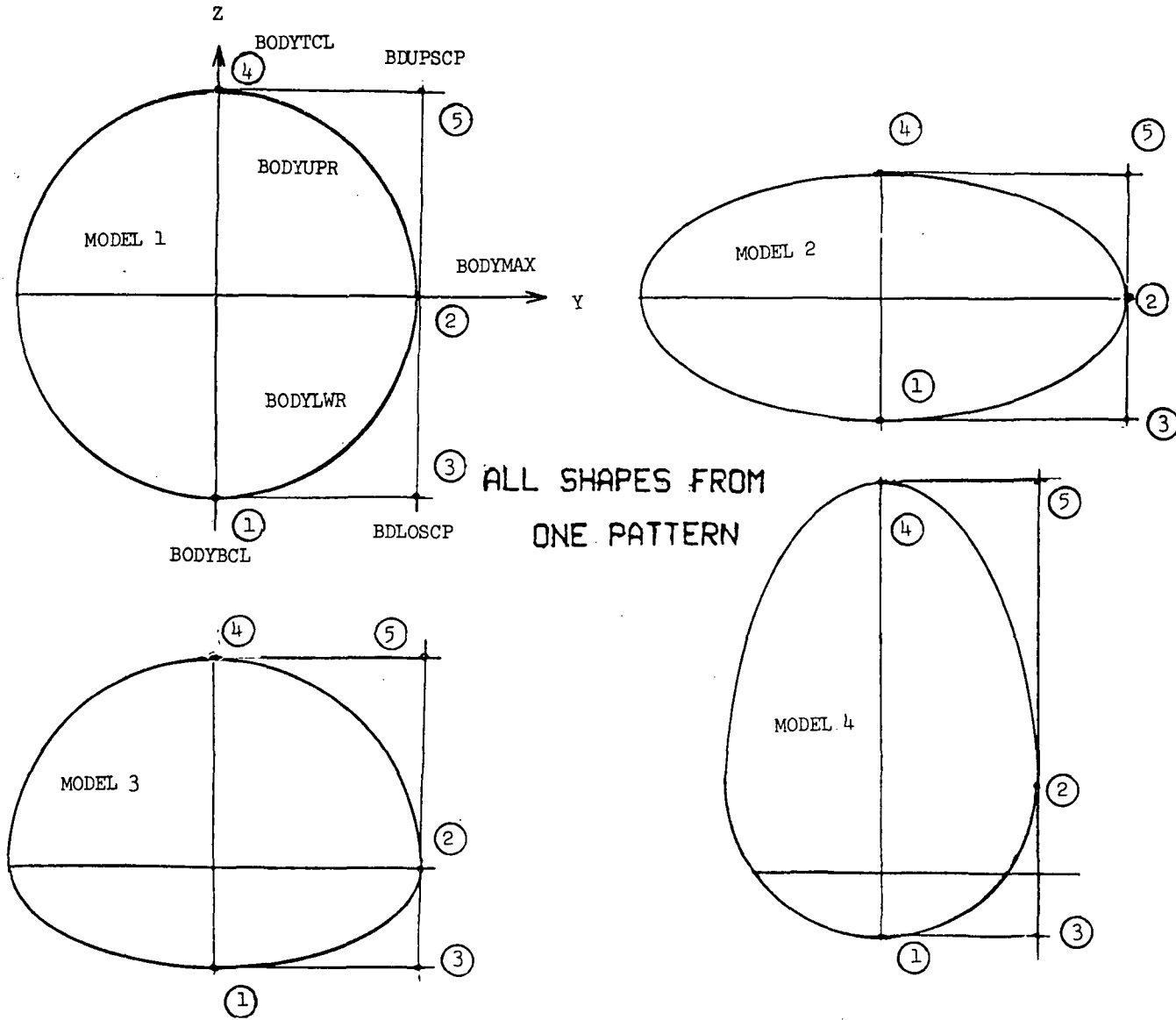


Figure 11.- Flexible pattern.

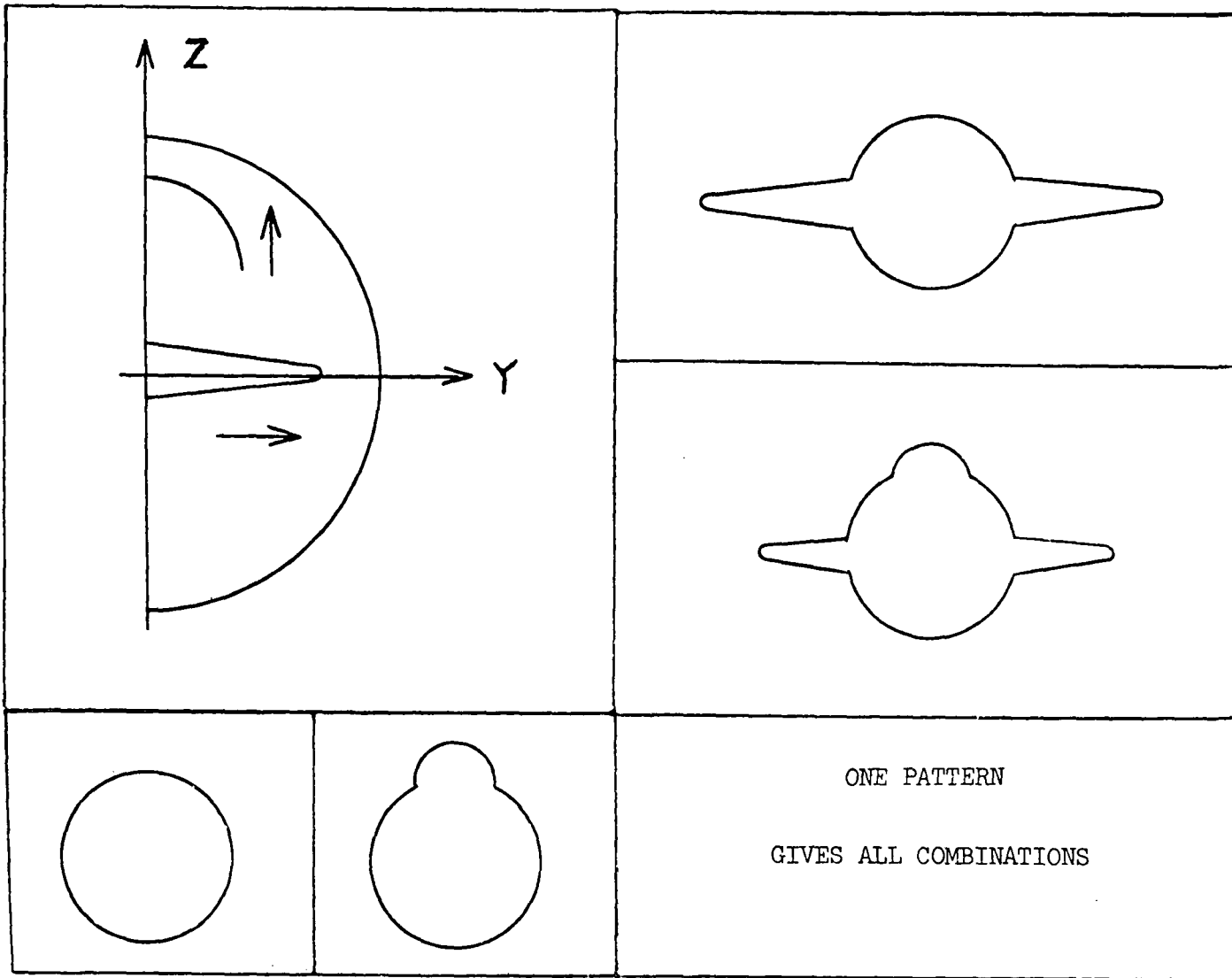


Figure 12.- Adapting pattern.

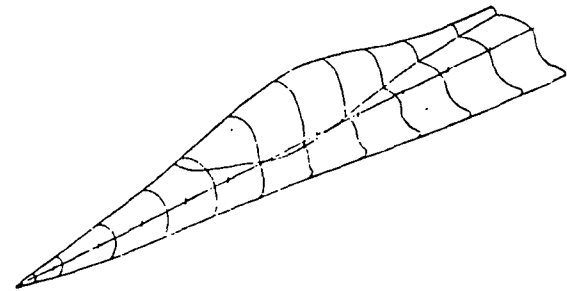
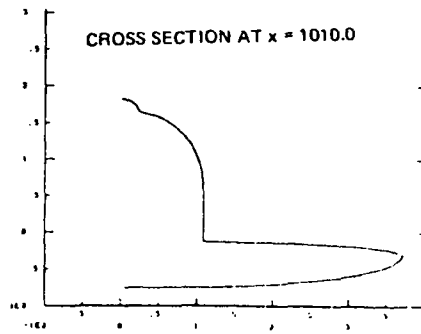
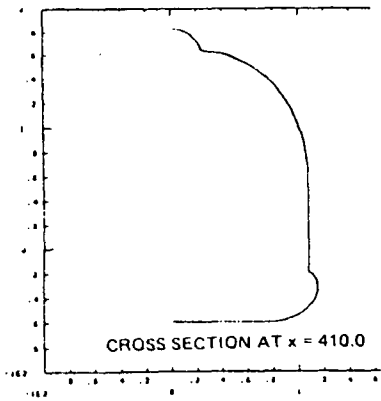
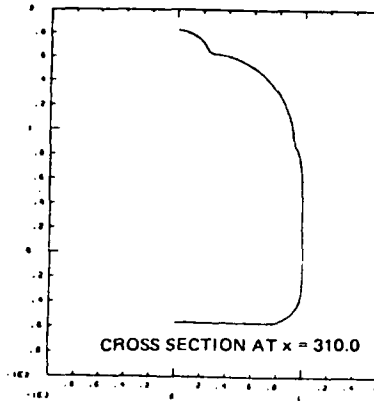
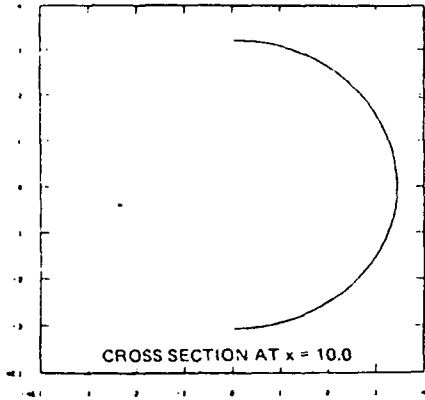
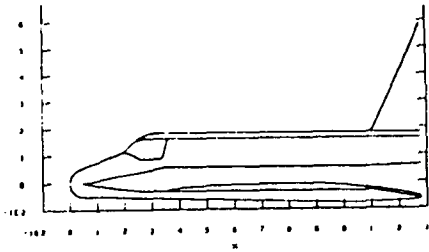


Figure 13.- Quick-geometry models.

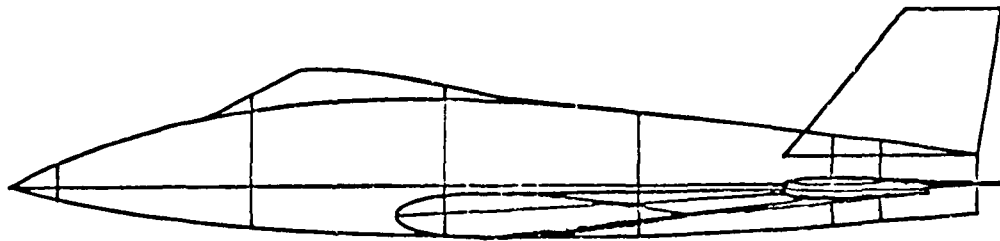
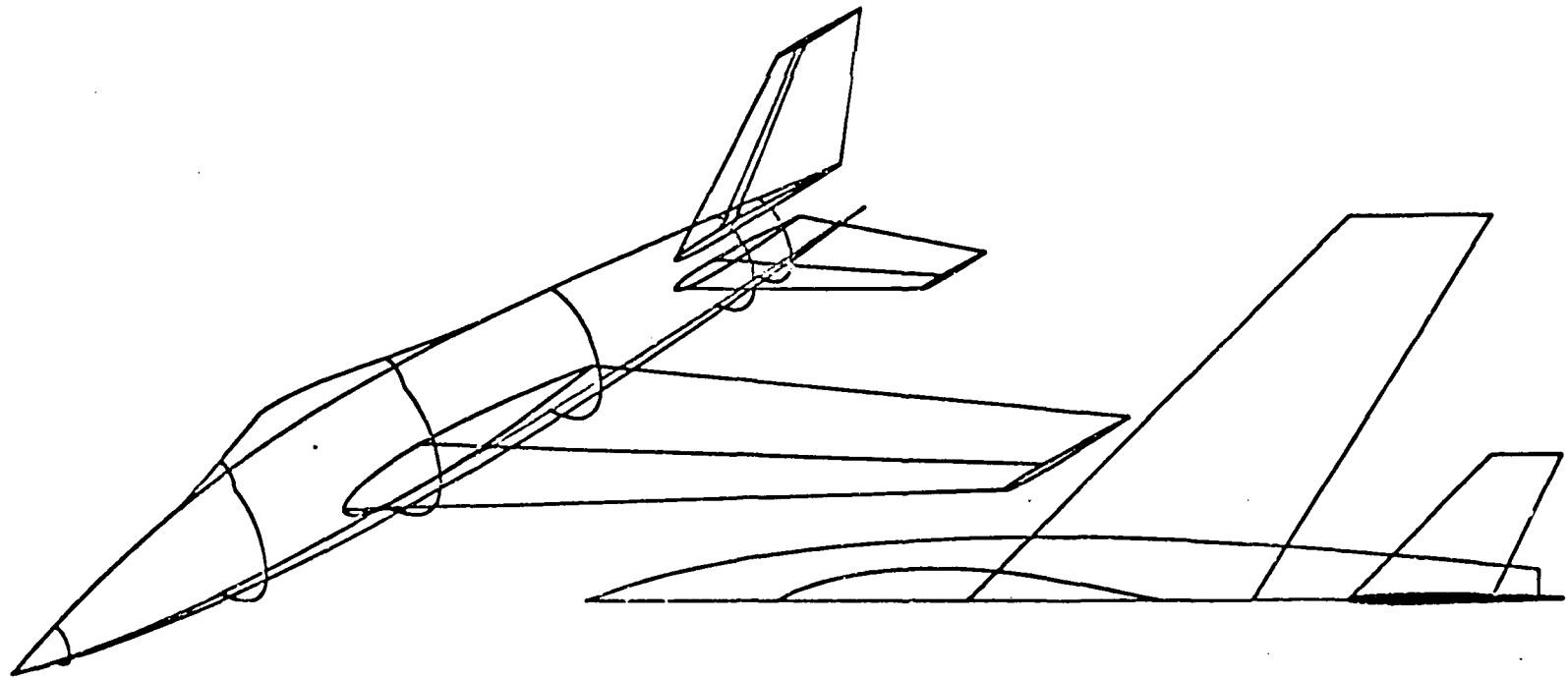


Figure 14.- Design synthesis.