NASA Technical Memorandum 83218

XSECT: A COMPUTER CODE FOR GENERATING FUSELAGE CROSS SECTIONS - USER'S MANUAL

NASA-TM-83218 19840019628

KATHY R. AMES

FEBRUARY 1982

Because of its significant early commercial potential, this information, which has been developed under a U.S. Government program, is being disseminated within the United States in advance of general publication. This information may be duplicated and used by the recipient with the express limitation that it not be published. Release of this information to other domestic parties by the recipient shall be made subject to these limitations.

Foreign release may be made only with prior NASA approval and appropriate export licenses. This legend shall be marked on any reproduction of this information in whole or in part.

Review for general release: February 28, 1984

NASA
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665
# TABLE OF CONTENTS

Summary .................................................. 1
Introduction .............................................. 2
Discussion .................................................. 3
  Processing a Circular Cross Section Without
  Area Balancing .......................................... 3
  Processing a Circular Cross Section With Area
  Balancing ................................................. 4
  Processing the Noncircular Shape ...................... 4
Description of Program Use ............................... 6
  Input Data ................................................ 6
  Accessing and Running the Program ...................... 6
  Output Files Produced by XSECT ......................... 8
Concluding Remarks ....................................... 9
Appendix A - Examples of Program Execution ............ 10
  Example 1 ................................................ 11
  Example 2 ................................................ 18
Appendix B - Special Error Conditions ................... 32
  When Area Balancing Criterion Cannot be Satisfied .... 32
  When R Cannot be Computed for a Given Cross Section . 36
  Other Errors that Stop Program Execution ............... 43
References ................................................ 45
Figures .................................................... 46
SUMMARY

A computer code, XSECT, has been developed to generate fuselage cross sections from a given area distribution and wing definition. The cross sections are generated to match the wing definition while conforming to the area requirement. An iterative procedure is used to generate each cross section. Fuselage area balancing may be included in this procedure if desired.

The code is intended as an aid for engineers who must first design a wing under certain aerodynamic constraints and then design a fuselage for the wing such that the constraints remain satisfied. This report contains the information necessary for accessing and executing the code, which is written in FORTRAN to execute on the Cyber 170 series computers (NOS operating system) and produces graphical output for a Tektronix 4014 CRT. The LRC graphics software is used in combination with the interface between this software and the PLOT 10 software.
INTRODUCTION

Computer code XSECT produces a digital fuselage description in the arbitrary geometry Harris or wave-drag format (ref. 1) using a given area distribution and a wing description. The cross sections are generated to have the area defined by the area distribution and to match the wing description out to a user-defined wing-body junction point. In general, the generated cross sections are circular except for a stub or indentation to fit the wing description; however, a noncircular shape, specifically, a circle with a circular vertical lobe, may be produced to represent a cockpit. The circular cross sections may be produced with or without fuselage area balancing. Fuselage area balancing means that when progressing from one cross section to the next along the fuselage, the change in fuselage area above the wing is equal to the change in fuselage area below the wing (refer to ref. 2).
DISCUSSION

Since there are two types of cross sections, circular and noncircular, and the circular cross sections may be generated with or without area balancing, there are three methods for processing a cross section. These three methods are described in the following sections.

PROCESSING A CIRCULAR CROSS SECTION WITHOUT AREA BALANCING

To generate the points to describe a particular cross section, the program must compute the value of the radius (R) of a circle whose combined areas above and below the wing will equal the desired fuselage area (refer to figure 1). The center of the circle is located at either the wing root camber line or the fuselage camber line defined in the wave-drag file. The value of R is computed iteratively. First, upper and lower limits are computed for R using geometric criteria. Then an initial estimate is made for R, the fuselage area is computed using this estimate, and the computed fuselage area is compared to the desired fuselage area. If the computed area is close enough to the desired area, the current value of R is used to generate the digital description of the fuselage. If not, a new estimate is made for R and the fuselage area is again computed and compared to the desired area. This iterative process is repeated until the fuselage area computed is close enough to the desired fuselage area and, therefore, the correct value of R is found.
(NOTE: For circular fuselage cross sections that do not intersect the wing, no iteration is needed since
\[
R = \sqrt{\frac{\text{fuselage area}}{\pi}}.
\]

PROCESSING A CIRCULAR CROSS SECTION WITH AREA BALANCING

Processing circular cross sections that intersect the wing using area balancing involves a double iteration. Not only must the correct value of R be found, but the center (ZC) of the circle must be located so that the balancing is achieved (refer to figure 2). First an initial estimate is made for ZC and the iterations to find R for the desired fuselage area are performed. After R is found, the area balancing criterion is checked. If this criterion is not satisfied, a new estimate for ZC is made, R is found, and again, the balancing criterion is checked. This iterative procedure continues until the correct value of ZC is found.

PROCESSING THE NONCIRCULAR SHAPE

The noncircular shape may be generated for a group of cross sections to represent a cockpit area. The shape consists of a circle with a circular vertical lobe (refer to figure 3). To exercise the cockpit option, the user must define where the cockpit begins and ends, the upper surface profile of the cockpit \(Z_{\text{MAX}}(x)\), and a constant that controls the location of the center of the upper circle (vertical lobe) with respect to the center of the main circle. \(Z_{\text{MAX}}(x)\) may be defined by typing in the values or by using the crosshair cursor to trace the shape.
The cross sections behind the cockpit are processed first and this processing determines the center (ZC) of each of the cockpit stations. To compute R for a given cockpit station, upper and lower limits for R are first computed using geometric criteria. Then, an initial estimate is made for R and the center of the lobe (Z2) is computed using equation 1, shown in figure 3. Next, the radius of the lobe (R2) is computed using equation 2, also shown in figure 3. The shape is now defined and its area may be computed and checked. If the area is not close enough to the desired area, a new estimate is made for R, a new shape is determined by the process described above, and a new area is computed and checked. This continues until the computed area is correct and, therefore, the desired value of R is found.

(Note: Unlike the circular cross sections, even cross sections that do not intersect the wing require iteration to find R since R cannot be defined in terms of the desired area by a simple equation.)
DESCRIPTION OF PROGRAM USE

INPUT DATA

The input data are provided by the user in a wave-drag geometry input file. The file should contain a wing description and the fuselage description should be that of a circular fuselage. The circular fuselage is specified by $J2 = -1$ which indicates that the fuselage will be described by cross-sectional areas. This wave-drag file is used as input to a preprocessing program (Q2650) which defines necessary wing cross-sectional data at fuselage locations determined by the XFUS locations in the wave-drag file. These data are written by Q2650 to TAPE7 along with a list of the $x$ locations of the cross sections, the fuselage areas at each of the $x$ locations, and the camber ordinates at each $x$ location. The procedure file to run XSECT takes care of this step for the user.

ACCESSING AND RUNNING THE PROGRAM

The procedure file to run the program resides on user number 723796N. To access the procedure file, type

GET,XSECTP/UN-723796N
The contents of XSECTP are shown below:

PROCEDURE XSECTP,FILE1,FILE2.

1 R1=1 - FIRST RUN FOR THIS CONFIGURATION.
2 R1=1 - NOT FIRST RUN FOR THIS CONFIGURATION.
3 T5 - ORIGINAL WAVE DRAG FILE.
4 T7 - FILE OF CROSS SECTION DATA FOR T5.
5 R1=1 - EITHER PRODUCED AND SAVED BY THIS
6 PROCEDURE FILE (R1=1) OR EXISTS ON
7 USER'S ACCOUNT (R1=1).
GET,TAPES=TS.
GET,XSECTB/UN=723796H.
SETTL(500)
* IF FIRST RUN, GENERATE TAPE7 AND SAVE.
IF(R1.EQ.1),LABEL1.
GET,CBC/UN=723796H.
CBC(TAPES)
RETURN,TAPE6,TAPE9,TAPE10,TAPE12.
REWIND,TAPES,TAPE7.
SAVE,TAPE7,T7.
* IF NOT FIRST RUN, GET TAPE7.
ELSE,LABEL1.
GET,TAPE7,T7.
ENDIF,LABEL1.
EXECUTE PROGRAM.
XSECTB.

To run the program for a particular configuration for the first time, type

XSECTP,1,FILE1,FILE2

The 1 indicates that this is the first run for this configuration. FILE1 is the name of the permanent file on the users account on which the wave-drag input data resides. Since this wave drag file is being processed for the first time, program Q2650 is run to get the digital cross-sectional data (TAPE7).

FILE2 is the name of the permanent file on the users account on which these data will be saved. If this particular configuration needs to be processed again, the digital cross-sectional data from the first run may be re-used since it has been saved on FILE2.

This saves time since program Q2650 need not be run prior to program XSECT. To use this option, type

XSECTP,2,FILE1,FILE2

where the 2 indicates this is not the first run for this configuration. FILE1 is the permanent file on which the wave drag data resides, and FILE2 is the permanent file on which the digital cross-sectional data have been saved.
OUTPUT FILES PRODUCED BY XSECT

The following output files are produced by program XSECT and exist as local files after program execution is complete.

TAPE8 - contains the digital description of the fuselage cross sections in wave drag format. (See figures 6 and 10 for examples.)

TAPE10 - table of information about each cross section. One line is entered in the table for each cross section produced. The entries occur in the order that the cross sections are generated. (See figures 5 and 9 for examples.)

SAVPVF - the plot vector file containing a plot of each cross section generated. The plots are on the file in the order that the cross sections were produced. This plot vector file may be viewed using PLIST. To use PLIST, make sure SAVPVF is a local file and type

```
GET,PLIST/UN-LIBRARY
PLIST
```

PLIST gives instructions on how to view the plots in random or sequential order. (See figures 4 and 8 for example plots.)

TAPE9 - traces the iterations used to find the radius of each cross section (used for debugging during program development).

TAPE11 - traces the iterations used to find the correct center of each cross section when area balancing (used for debugging during program development).
CONCLUDING REMARKS

A computer code has been developed to produce a digital fuselage description in the arbitrary geometry wave-drag format using a given area distribution and a wing description. This code is designed to automate a process which has been performed manually in the past. Since the results of a particular attempt at a fuselage fit are available much more quickly by using the automated process, the user has time to make several attempts and select the best rather than being forced to accept the first satisfactory results.
APPENDIX A - EXAMPLES OF PROGRAM EXECUTION

The following examples explain the interactive prompts and possible user responses required by running a very simple case of the program (Example 1) and a complicated case (Example 2). These cases can be repeated at the terminal as indicated.

All questions requiring a yes-no response should be answered by typing "Y" for yes and "N" for no.

User responses are checked for validity whenever possible so that typing errors should not abnormally terminate the program. An invalid response will either cause the question to be repeated or a default option to be taken.
Example 1 illustrates program execution for a very simple case. It requires the least user input from the terminal. The wing described in the wave-drag file is defined to the center line, no area balancing is requested, and the cockpit option is not requested. The specific input files used for this case are the wave-drag file, MACK, and the cross-sectional data file, QINP1. The user may repeat this example by typing

```
GET,MACK,QINP1,XSECTP/UN*723796N
SAVE,MACK,QINP1
XSECTP,2,MACK,QINP
```

The first input required is the wing-body junction point (YJCT). This y value is chosen so that all of the wing-fuselage from y = 0 to y = YJCT will be included in the digital fuselage description. YJCT must be one of the wing y leading-edge values defined in the wave-drag file.

```
INPUT WING-BODY JUNCTION PT (YJCT)
```
Next, the user indicates how the wing and fuselage are defined.

1. Wing defined from \( y = 0 \), fuselage area includes area to be placed around wing

2. Wing defined from \( y = 0 \), fuselage area includes area of wing to \( Y_{JCT} \)

3. Wing defined from \( y = (Y_{JCT}) \), fuselage area includes all area out to \( Y_{JCT} \)

The fuselage type is selected next.

INPUT 'FUSELAGE TYPE'
1 - CIRCULAR
2 - CIRCULAR WITH VERTICAL LOBE FOR COCKPIT

The data just entered are echoed to the screen and the user is asked if it is okay to continue. To change a response to any of the previous questions, the user answers "NO" and the questions are repeated. Otherwise, a "YES" response causes program execution to continue.

WING-SODY JCT PT - 6.000
INPUT IS TYPE 1
FUSELAGE DESIRED IS TYPE 1

OKAY TO CONTINUE? (Y/N) ? Y
Next, the screen is erased and a table is displayed listing each cross section by x station number and its corresponding x location. The user should make a hard copy of this table since cross sections are referred to by x station number in the future. A carriage return will cause program execution to continue. The screen will be erased before the next input prompt.

<table>
<thead>
<tr>
<th>X STATION NO.</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>5.000</td>
</tr>
<tr>
<td>3</td>
<td>10.000</td>
</tr>
<tr>
<td>4</td>
<td>15.000</td>
</tr>
<tr>
<td>5</td>
<td>20.000</td>
</tr>
<tr>
<td>6</td>
<td>25.000</td>
</tr>
<tr>
<td>7</td>
<td>30.000</td>
</tr>
<tr>
<td>8</td>
<td>35.000</td>
</tr>
<tr>
<td>9</td>
<td>40.000</td>
</tr>
<tr>
<td>10</td>
<td>45.000</td>
</tr>
<tr>
<td>11</td>
<td>50.000</td>
</tr>
<tr>
<td>12</td>
<td>55.000</td>
</tr>
<tr>
<td>13</td>
<td>60.000</td>
</tr>
<tr>
<td>14</td>
<td>65.000</td>
</tr>
<tr>
<td>15</td>
<td>70.000</td>
</tr>
<tr>
<td>16</td>
<td>75.000</td>
</tr>
<tr>
<td>17</td>
<td>80.000</td>
</tr>
<tr>
<td>18</td>
<td>85.000</td>
</tr>
<tr>
<td>19</td>
<td>90.000</td>
</tr>
<tr>
<td>20</td>
<td>95.000</td>
</tr>
<tr>
<td>21</td>
<td>100.000</td>
</tr>
<tr>
<td>22</td>
<td>105.000</td>
</tr>
<tr>
<td>23</td>
<td>110.000</td>
</tr>
<tr>
<td>24</td>
<td>115.000</td>
</tr>
<tr>
<td>25</td>
<td>120.000</td>
</tr>
<tr>
<td>26</td>
<td>125.000</td>
</tr>
<tr>
<td>27</td>
<td>130.000</td>
</tr>
<tr>
<td>28</td>
<td>135.000</td>
</tr>
<tr>
<td>29</td>
<td>140.000</td>
</tr>
<tr>
<td>30</td>
<td>145.000</td>
</tr>
<tr>
<td>31</td>
<td>150.000</td>
</tr>
<tr>
<td>32</td>
<td>155.000</td>
</tr>
<tr>
<td>33</td>
<td>160.000</td>
</tr>
<tr>
<td>34</td>
<td>165.000</td>
</tr>
<tr>
<td>35</td>
<td>170.000</td>
</tr>
<tr>
<td>36</td>
<td>175.000</td>
</tr>
<tr>
<td>37</td>
<td>180.000</td>
</tr>
<tr>
<td>38</td>
<td>185.000</td>
</tr>
<tr>
<td>39</td>
<td>190.000</td>
</tr>
<tr>
<td>40</td>
<td>195.000</td>
</tr>
<tr>
<td>41</td>
<td>200.000</td>
</tr>
<tr>
<td>42</td>
<td>205.000</td>
</tr>
<tr>
<td>43</td>
<td>210.000</td>
</tr>
<tr>
<td>44</td>
<td>215.000</td>
</tr>
<tr>
<td>45</td>
<td>220.000</td>
</tr>
<tr>
<td>46</td>
<td>225.000</td>
</tr>
<tr>
<td>47</td>
<td>230.000</td>
</tr>
<tr>
<td>48</td>
<td>235.000</td>
</tr>
<tr>
<td>49</td>
<td>240.000</td>
</tr>
<tr>
<td>50</td>
<td>245.000</td>
</tr>
<tr>
<td>51</td>
<td>250.000</td>
</tr>
<tr>
<td>52</td>
<td>255.000</td>
</tr>
<tr>
<td>53</td>
<td>260.000</td>
</tr>
</tbody>
</table>

PLEASE MAKE HARD COPY FOR FUTURE REFERENCE
HIT CR TO CONTINUE
At this point, the user must indicate whether or not area balancing is desired.

**DO YOU WISH TO AREA BALANCE? (Y/N)**

? N

By selecting no area balancing, the user is required to indicate how the fuselage cross sections should be centered. They may be centered at the wing-root camber line or they may be centered at the ZFUS locations defined in the wave-drag file. By selecting option 1 below, cross sections ahead of and behind the wing are centered to follow the fuselage camber of the input file but are adjusted so there is no discontinuity at the first and last wing station.

**HOW DO YOU WANT CROSS SECTIONS CENTERED?**

1 - TO FOLLOW WING CAMBER
2 - TO FOLLOW CAMBER OF FUSELAGE IN INPUT FILE

? 1

Again, the user is asked if it is okay to continue. If not, the option to request area balancing is repeated. If so, processing continues.

**OKAY TO CONTINUE? (Y/N)**

? Y

The user now indicates the total number of points desired to describe each fuselage cross section. The minimum is 8 and the maximum is 30.

**ENTER TOTAL NO. OF PTS TO DESCRIBE EACH CROSS SECTION (MIN 8, MAX 30)**

? 20
Next, the user indicates how these points should be distributed around the cross section. First, how many points should be placed on the part of the fuselage above the wing? This number includes the point located at the wing-body intersection. The minimum is 3: 1 for the wing-body intersection, 1 for the top point, and at least 1 point in between.

**ENTER NO. ABOVE WING, INCLUDING PT OF INTERSECTION OF WING & FUS (MIN 3)**

? 6

Next, how many points should be placed on the upper wing surface? This number does not include the wing-body intersection point, but does include the point at the wing-body junction point (YJCT). The minimum number is 1.

**ENTER NO. ON UPPER WING, INCLUDING PT AT Y-YJCT (MIN 1)**

? 4

Next, the user indicates the number of points to be placed on the lower wing surface, following the same directions as those for the upper wing surface, and then the number of points on the part of the fuselage below the wing, following the same directions as those for the fuselage above the wing.

**ENTER NO. ON LOWER WING, INCLUDING PT AT Y-YJCT (MIN 1)**

? 4

**ENTER NO. BELOW WING, INCLUDING PT OF INTERSECTION OF WING & FUS (MIN 3)**

? 6
Each response is checked for violation of the minimum and/or maximum. If the number entered is invalid, the question is repeated. Also, if the sum of the distribution of points around the fuselage does not equal the total number of points requested, the following message is displayed and the sequence of questions repeated.

```
SUM OF PTS DISTRIBUTED NOT = TOTAL PTS
III TRY AGAIN!
```

After valid data have been entered, the data are echoed and the user must again indicate whether or not it is okay to continue.

```
TOTAL FUS PTS = 20 DISTRIBUTED
ABOVE WING (UPPER FUS) 6
UPPER WING 4
LOWER WING 4
BELOW WING (LOWER FUS) 6

OKAY TO CONTINUE? (Y/N)
```

A "NO" response requires that all of the number of points information be re-entered and a "YES" response causes the screen to be erased and the processing of each cross section to begin.

As a cross section is completed, a message is displayed on the screen so the user can keep track of the processing. The cross sections will probably not be processed in sequential order. (The order of processing depends on the options selected and the location of the leading and trailing edges of the wing.)
PROGRAM EXECUTION COMPLETE

After execution is complete, the output files described previously exist as local files. Figure 4 shows 3 examples of the cross sectional plots contained on the plot vector file SAVPVF. An example of TAPE10, the table of cross section information, is shown in figure 5, and TAPE8, the digital description of the cross sections in wave-drag format, is shown in figure 6.
Example 2 illustrates program execution for a complicated case. It requires a considerable amount of user input from the terminal. The wing is defined only to the wing-body junction point (YJCT = 25.926 in this case). Both the area balancing and cockpit options are requested. The input files used for this case are the wave-drag file, F16AM, and the cross-sectional data file, F16AMX. The user may repeat this example by typing

```
GET,F16AM,F16AMX,XSECTP/UN-723796N
SAVE,F16AM,F16AMX
XSECTP,2,F16AM,F16AMX
```

The user enters YJCT, type of wing-fuselage data, and type of fuselage desired as in example 1. This time, input type 3 and fuselage type 2 are selected.

```
INPUT WING-BODY JUNCTION PT (YJCT)
? 25.926

ENTER TYPE OF INPUT
  1 - WING DEF FROM Y=0, FUS AREA NOT INCLUDING WING
  2 - WING DEF FROM Y=0, FUS AREA INCLUDES WING TO Y=YJCT
  3 - WING DEF FROM Y=YJCT, FUS AREA INCLUDES WING TO Y=YJCT
? 3

INPUT FUSELAGE TYPE
  1 - CIRCULAR
  2 - CIRCULAR WITH VERTICAL LOBE FOR COCKPIT
? 2

WING-BODY JCT PT = 25.926
INPUT IS TYPE 3
FUSELAGE DESIRED IS TYPE 2

OKAY TO CONTINUE? (Y/N)
? Y
```
Again, a table of x station numbers and corresponding x values is displayed.

<table>
<thead>
<tr>
<th>X STATION NO.</th>
<th>X</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-38.000</td>
</tr>
<tr>
<td>2</td>
<td>-25.000</td>
</tr>
<tr>
<td>3</td>
<td>-12.500</td>
</tr>
<tr>
<td>4</td>
<td>0.030</td>
</tr>
<tr>
<td>5</td>
<td>12.500</td>
</tr>
<tr>
<td>6</td>
<td>25.000</td>
</tr>
<tr>
<td>7</td>
<td>37.500</td>
</tr>
<tr>
<td>8</td>
<td>50.000</td>
</tr>
<tr>
<td>9</td>
<td>62.500</td>
</tr>
<tr>
<td>10</td>
<td>75.000</td>
</tr>
<tr>
<td>11</td>
<td>87.500</td>
</tr>
<tr>
<td>12</td>
<td>100.000</td>
</tr>
<tr>
<td>13</td>
<td>112.500</td>
</tr>
<tr>
<td>14</td>
<td>125.000</td>
</tr>
<tr>
<td>15</td>
<td>137.500</td>
</tr>
<tr>
<td>16</td>
<td>150.000</td>
</tr>
<tr>
<td>17</td>
<td>162.500</td>
</tr>
<tr>
<td>18</td>
<td>175.000</td>
</tr>
<tr>
<td>19</td>
<td>187.500</td>
</tr>
<tr>
<td>20</td>
<td>200.000</td>
</tr>
<tr>
<td>21</td>
<td>212.500</td>
</tr>
<tr>
<td>22</td>
<td>225.000</td>
</tr>
<tr>
<td>23</td>
<td>237.500</td>
</tr>
<tr>
<td>24</td>
<td>250.000</td>
</tr>
<tr>
<td>25</td>
<td>262.500</td>
</tr>
<tr>
<td>26</td>
<td>275.000</td>
</tr>
<tr>
<td>27</td>
<td>287.500</td>
</tr>
<tr>
<td>28</td>
<td>300.000</td>
</tr>
<tr>
<td>29</td>
<td>312.500</td>
</tr>
<tr>
<td>30</td>
<td>325.000</td>
</tr>
<tr>
<td>31</td>
<td>337.500</td>
</tr>
<tr>
<td>32</td>
<td>350.000</td>
</tr>
<tr>
<td>33</td>
<td>362.500</td>
</tr>
<tr>
<td>34</td>
<td>375.000</td>
</tr>
<tr>
<td>35</td>
<td>387.500</td>
</tr>
<tr>
<td>36</td>
<td>400.000</td>
</tr>
<tr>
<td>37</td>
<td>412.500</td>
</tr>
<tr>
<td>38</td>
<td>425.000</td>
</tr>
<tr>
<td>39</td>
<td>437.500</td>
</tr>
</tbody>
</table>

PLEASE MAKE HARD COPY FOR FUTURE REFERENCE
HIT CR TO CONTINUE

Since the wing is not defined to the center line, the definition must be extended by the program to \( y = 0 \) so fuselage cross sections with wing stub will have some way of matching the wing definition at \( y = Y_{JCT} \). The user must guide this extension.

First, the number of additional airfoils desired in the extended definition is specified. This number includes an airfoil located at \( y = 0 \). The user may select from 1 to 6 additional airfoils.

ENTER NO. OF AIRFOILS TO EXTEND WING DEF TO Y=0, INCLUDING Y=0
(1 <= NO. OF AIRFOILS <= 6)
? 1
Next, the user indicates how the leading-edge x and y values and chord length for each airfoil should be defined. The user may type in the values or have the program extrapolate the values linearly using the first two airfoils defined in the original wing definition.

**ENTER HOW TO DEFINE X, Y AND C VALUES**

1 - PROGRAM EXTRAPOLATES VALUES FROM GIVEN DEF  
2 - USER INPUT X, Y AND C FOR EACH AIRFOIL

? 2

By selecting option 2, the x and y leading-edge values and chord length for the one additional airfoil requested must be entered.

**AFTER EACH PROMPT ENTER X, Y AND C, BEGINNING WITH Y=0**

EX = 0, 0.50

? 62.82, 0.141 511

At this point, the leading- and trailing-edge extensions are computed. The slope, \( \frac{dy}{dx} \), of the extended part of the leading edge can never be negative and the slope of the extended part of the trailing edge can never be positive. These restrictions result from the method used in the program to extend the wing definition to \( y = 0 \). The restricting conditions are checked, and if either is violated, the following message is displayed and the leading and trailing edges will automatically be extended straight across to \( y = 0 \) using the x values of the leading and trailing edges at the first original airfoil (airfoil defined at \( y = Y_{JCT} \)).
Whether or not the extended edges are invalid, the user is asked if it is okay to continue.

**OKAY TO CONTINUE? (Y/N)**

? Y

A "YES" answer accepts the computed extended definition and processing continues. A "NO" answer enables the user to again select how to define the leading and trailing edges.

If option 1 to define the leading-edge values and chord lengths is selected, no additional input is required. The extension of the leading and trailing edges is computed. The slopes are checked as in option 2 and if either is invalid, no message is printed but both the leading and trailing edges are automatically extended straight across to $y = 0$ using the $x$ values of the leading and trailing edges at the first original airfoil. Next, the leading-edge definition is checked for extension beyond the fuselage nose. If this occurs, the following message is displayed.

**LEADING EDGE BEYOND FUSELAGE NOSE**

**OKAY TO CONTINUE? (Y/N)**

? N

By answering "NO" to "OKAY TO CONTINUE?", the following message is displayed

**YOU MUST ENTER X, Y AND C'S**

and instructions indicating how to input the leading edge values and chord lengths are displayed (same prompts as when option 2 is selected). In other words, the extrapolation of the $x$ and $y$ values of the leading edge has produced a wing whose definition
extends beyond the fuselage nose. If this is not acceptable, the user must define the leading edge. If it is acceptable, a "YES" response to "OKAY TO CONTINUE?" causes processing to continue. The trailing edge is checked for extension past the tail of the fuselage in a similar manner.

Once the x and y values of the leading and trailing edges have been computed and accepted, the user determines how the z definition (camber) of the extended wing should be computed. The thickness distribution of the first original airfoil is used and there are two ways of defining the camber.

**ENTER HOW TO EXTEND WING CAMBER**

1 - TO FOLLOW FUS CAMBER FROM INPUT
2 - TO FOLLOW CAMBER DEF OF AIRFOIL AT Y\*YJCT

Option 1 causes the airfoil at \( y = 0 \) to follow the camber definition of the fuselage from the original input (ZFUS values). The camber value for airfoils between \( y = 0 \) and the airfoil at \( Y_{JCT} \) is linearly interpolated between the two airfoils. Option 2 causes all extended airfoils to have the same camber definition as the airfoil at \( Y_{JCT} \).

Since a fuselage with a cockpit section has been selected, some information about the noncircular cross sections to represent the cockpit is needed.

First, the constant that is used to control the placement of ZC is entered. (See Processing the Noncircular Shape.) A value of 0.8 is suggested, but any valid number may be used between
0.1 and 1.4 (the input will be checked). This constant will be used in the processing of every cockpit cross section.

ENTER CONSTANT CONTROLLING VERTICAL POSITION OF COCKPIT (.1<CONSTANT<1.4 - TRY .8)

9.8

Next, the x station numbers of the cross sections where the cockpit should begin and end are entered.

ENTER BEGINNING, ENDING STATION FOR COCKPIT

7, 7, 16

The program checks that there is at least one cross section between the beginning and ending stations. If not, the following message is displayed and the question is repeated.

NO X STATIONS BETWEEN COCKPIT BEGINNING AND ENDING STATIONS

Also, the program checks that the cockpit ending station is less than or equal to the wing ending station. If not, the following message is displayed and the question is repeated.

WING ENDS BEFORE COCKPIT ENDS

Finally, the cockpit information is echoed and the user indicates whether or not it is okay to continue. By answering "NO", the constant and beginning and ending cockpit stations may be re-entered. A "YES" answer will cause processing to continue.

8 COCKPIT STATIONS BETWEEN STATION NO. 7 X = 37.50 AND STATION NO. 16 X = 168.00

OKAY TO CONTINUE? (Y/N)

Y

23
Next, the user indicates whether or not area balancing is desired.

**DO YOU WISH TO AREA BALANCE? (Y/N)**

? Y

By answering yes, further input is required. The user now selects the key station. This is the cross section that must be processed first to initiate the area balancing.

**ENTER NO. OF KEY STATION**

? 20

The key station number is checked to insure that the wing is defined at that cross section. If not, the following message is displayed and the question is repeated.

**THERE IS NO WING DEFINED AT KEY STATION NO. 2**

**TRY AGAIN!**

Since the cockpit option was selected, the program also checks that the key station is not located in the cockpit area. It can be located at the cockpit ending station but not before. If this restriction is violated, the following message is displayed and the key station number must be re-entered.

**KEY STATION CANNOT BE LOCATED IN COCKPIT AREA**

**TRY AGAIN!**

Next, the user enters the percentage of the fuselage area that should be located above the wing at the key station.

**ENTER PERCENT AREA ABOVE WING (0.0<PERCENT<100.0)**

? 60.
If this number is not between 0.0 and 100.0, the question is repeated. The percentage of area required below the wing is computed (100 - percentage above) and the data are echoed to the screen. A "NO" response causes the option to request area balancing to be repeated. A "YES" response causes processing to continue.

```
BALANCING WITH KEY STATION NO. 20 X = 200.00
PERCENT ABOVE = 60.0000  PERCENT BELOW = 40.0000
OKAY TO CONTINUE? (Y/N)  ? Y
```

As in the first example, the user enters the total number of points to describe each cross section, followed by how the points should be distributed on the different parts of the cross section.

```
ENTER TOTAL NO. OF PTS TO DESCRIBE EACH CROSS SECTION (MIN 10, MAX 30)? 20
ENTER NO. ABOVE WING, INCLUDING PT OF INTERSECTION OF WING & FUS (MIN 5)  ? 8
ENTER NO. ON UPPER WING, INCLUDING PT AT Y-YJCT (MIN 1)
? 4
ENTER NO. ON LOWER WING, INCLUDING PT AT Y-YJCT (MIN 1)
? 4
ENTER NO. BELOW WING, INCLUDING PT OF INTERSECTION OF WING & FUS (MIN 3)
? 4
```

Notice that the minimum number of points for the fuselage above the wing has increased to five (therefore, the overall minimum has increased to ten). This is because the cockpit option has been selected and more points are needed to represent the vertical lobe. One point is allowed for the top point, one for the intersection point of the two circles, one for the wing-body intersection point, and one on each circular arc. The user specifies how these upper fuselage points should be distributed.
between the upper lobe (cockpit) and the part of the main circle that is above the wing. The points on the cockpit include the top point and the intersection point of the two circles and at least one point in between these two.

**HOW TO DIVIDE UPPER FUS PTS FOR COCKPIT STATIONS?**
ENTER NO. ON COCKPIT (MIN 3)

The points on the main body include the wing-body intersection point and at least one point between this intersection point and the intersection point of the two circles.

**ENTER NO. ON MAIN BODY (MIN 2)**

If the sum of this distribution is not equal to the number of points requested above the fuselage, the following message is displayed and the distribution must be re-entered.

**DISTRIBUTION OF UPPER FUS PTS HAS INCORRECT SUM**

Finally, the overall point distribution and the upper fuselage point distribution are printed and the user is asked if it is okay to continue.

**TOTAL FUS PTS: 20 DISTRIBUTED**
- ABOVE UING (UPPER FUS) 8
- UPPER UING 4
- LOWER UING 4
- BELOW UING (LOWER FUS) 4
- UPPER FUS PTS DISTRIBUTED
  - 4 ON COCKPIT
  - 4 ON BODY

**OKAY TO CONTINUE? (Y/N)**

? Y
A "NO" answer requires that the number of points information be re-entered. A "YES" response causes a screen erasure and continuation of processing.

At this point, the cross sections behind (and possibly ahead of) the cockpit are processed and a message is displayed to the screen as each cross section is completed.

```
X STATION NO.  20 COMPLETE
X STATION NO.  19 COMPLETE
X STATION NO.  18 COMPLETE
X STATION NO.  17 COMPLETE
X STATION NO.  16 COMPLETE
X STATION NO.  21 COMPLETE
X STATION NO.  22 COMPLETE
X STATION NO.  23 COMPLETE
X STATION NO.  24 COMPLETE
X STATION NO.  25 COMPLETE
X STATION NO.  26 COMPLETE
X STATION NO.  27 COMPLETE
X STATION NO.  28 COMPLETE
X STATION NO.  29 COMPLETE
X STATION NO.  30 COMPLETE
```

After these cross sections have been processed, the screen is erased and the upper surface profile of the cockpit area must be defined before the cockpit stations can be processed.

Now, a table of the cross sections to be processed with the vertical lobe is printed. This is a list of the cross sections between the beginning and ending stations (since the beginning and ending stations are processed as circular cross sections). The user then selects one of the two options for defining \( z_{\text{MAX}} \).
THE FOLLOWING CROSS SECTIONS LIE BETWEEN THE COCKPIT BEGINNING AND ENDING STATIONS AND REQUIRE THE UPPER SURFACE PROFILE (ZMAX) TO BE INPUT

<table>
<thead>
<tr>
<th>NO.</th>
<th>X SECTION NO.</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>50.00</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>62.50</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>75.00</td>
</tr>
<tr>
<td>4</td>
<td>11</td>
<td>87.50</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>100.00</td>
</tr>
<tr>
<td>6</td>
<td>13</td>
<td>112.50</td>
</tr>
<tr>
<td>7</td>
<td>14</td>
<td>125.00</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>150.00</td>
</tr>
</tbody>
</table>

HOW DO YOU WANT TO DEFINE ZMAX?
1 - TYPE IN Z VALUES
2 - USE CROSSHAIR CURSOR TO SELECT POINTS

? 1

By selecting option 1, the user must numerically input the value of ZMAX for each cross section.

AFTER EACH PROMPT, ENTER ZMAX VALUE FOR EACH COCKPIT STATION, BEGINNING WITH NO. 1

? 12.029
? 17.636
? 23.399
? 27.604
? 30.310
? 31.498
? 30.875
? 28.694

As each value is entered, the program checks that ZMAX is above the wing upper surface at that cross section. If not, the following message is displayed and the ZMAX value for that cross section must be re-entered.

ZMAX REQUESTED < WING UPPER SURFACE
ZMAX = -15.000 WING = -1.1833
RE-ENTER ZMAX FOR THIS CROSS SECTION
A similar check is made to insure that $Z_{\text{MAX}}$ is above the center of the fuselage ($Z_C$) at that cross section and a similar message is displayed if $Z_{\text{MAX}}$ is less than $Z_C$. After all the $Z_{\text{MAX}}$ values have been entered, they are listed and the user is asked if it is okay to continue. A "NO" response allows the user to select the other option and/or re-enter the cockpit data. A "YES" response causes the cockpit stations to be processed.

**CURRENT UPPER SURFACE VALUES**

<table>
<thead>
<tr>
<th>NO.</th>
<th>$Z_{\text{MAX}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12.029</td>
</tr>
<tr>
<td>2</td>
<td>17.636</td>
</tr>
<tr>
<td>3</td>
<td>23.399</td>
</tr>
<tr>
<td>4</td>
<td>27.604</td>
</tr>
<tr>
<td>5</td>
<td>30.918</td>
</tr>
<tr>
<td>6</td>
<td>31.428</td>
</tr>
<tr>
<td>7</td>
<td>30.875</td>
</tr>
<tr>
<td>8</td>
<td>28.694</td>
</tr>
</tbody>
</table>

**OKAY TO CONTINUE? (Y/N) ? Y**

By selecting option 2, interactive graphics is used to input the upper surface profile. First, a side view of the entire configuration (except the cockpit area) is displayed on the upper portion of the screen. A box is drawn around the cockpit area and 1 or 2 stations on either side of this area. This boxed-in area is then enlarged and displayed on the lower portion of the screen. Vertical dashed lines are drawn on the enlargement to indicate the location of the cross sections requiring a $Z_{\text{MAX}}$ value. These cross sections are labeled by station number and the $x$ value at that station (refer to figure 7-a). Then, the cross-hair cursor appears on the screen. Beginning with the leftmost cross section, the cursor is positioned to trace the upper surface
of the cockpit. When the cursor is in the proper position for the first cross section, the user types "x" and the point is read. The value is checked for $Z_{\text{MAX}} > $ wing surface and $Z_{\text{MAX}} > Z_C$ as in option 1, and if either condition is violated, a message is displayed and the user must re-enter that point. If the point is okay, a plus is drawn at the point and the cursor re-appears, ready to accept the next point (refer to figure 7-b). This continues until the entire upper surface is defined. After the last point is accepted, a dashed line is drawn connecting the $Z_{\text{MAX}}$ values, as in figure 7-c. Then the terminal bell rings and the user types a carriage return to continue program execution. After the carriage return, the screen is erased and the $Z_{\text{MAX}}$ values that were entered are listed. If it is okay to continue, the cockpit stations will be processed. If not, the user may select the other option and re-enter the $Z_{\text{MAX}}$ values or the $Z_{\text{MAX}}$ values may be re-entered using this option again.

<table>
<thead>
<tr>
<th>X STATION NO.</th>
<th>X</th>
<th>ZMAX</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>50.00</td>
<td>12.0291</td>
</tr>
<tr>
<td>9</td>
<td>62.50</td>
<td>17.6361</td>
</tr>
<tr>
<td>10</td>
<td>75.00</td>
<td>23.3989</td>
</tr>
<tr>
<td>11</td>
<td>87.50</td>
<td>27.6841</td>
</tr>
<tr>
<td>12</td>
<td>100.00</td>
<td>30.8961</td>
</tr>
<tr>
<td>13</td>
<td>112.50</td>
<td>31.4970</td>
</tr>
<tr>
<td>14</td>
<td>125.00</td>
<td>30.8749</td>
</tr>
<tr>
<td>15</td>
<td>150.00</td>
<td>28.6944</td>
</tr>
</tbody>
</table>

OKAY TO CONTINUE? (Y/N)  
Y Y

30
Again, a message is displayed as the processing of each cross section is complete.

<table>
<thead>
<tr>
<th>X STATION NO.</th>
<th>COMPLETE</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>14</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>13</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>12</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>11</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>10</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>9</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>8</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>7</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>6</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>5</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>4</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>3</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>2</td>
<td>COMPLETE</td>
</tr>
<tr>
<td>1</td>
<td>COMPLETE</td>
</tr>
</tbody>
</table>

PROGRAM EXECUTION COMPLETE

After execution is complete, the output files produced by the program exist as local files. Examples of the cross section plots produced on the plot vector file SAVPVF are shown in Figure 8. Figures 8 (a) and (b) show cross sections in the cockpit area and (c) and (d) show cross sections behind the cockpit. TAPE10 is illustrated in figure 9 and TAPE8 in figure 10.
APPENDIX B - SPECIAL ERROR CONDITIONS

Explained in this section are special error conditions that will either stop program execution or require responses from the user before execution can continue.

WHEN AREA BALANCING CRITERION CANNOT BE SATISFIED

This problem occurs when the fuselage area is decreasing and there is not enough area left above or below the wing to satisfy the required decrease. The program will issue warning messages indicating that balancing cannot be achieved, and will offer the user options for continuing processing.

Shown below is a warning message that results from a preliminary check on the remaining upper and lower areas and the required decrease in upper and lower area to achieve balancing.

```
**CURRENT X STATION NO. • 41 , X • 450.00
BALANCING REQUIREMENT UNABLE TO BE SATISFIED
NECESSARY CHANGE ABOVE • CHANGE BELOW • -4.300
BEST BALANCE POSSIBLE IS CHANGE ABOVE • -8.559 CHANGE BELOW • -.042
HOW DO YOU WANT TO CONTINUE?
1 - ATTEMPT BEST POSSIBLE BALANCE
2 - SELECT NEW KEY STATION AND/OR INITIAL BALANCE CONDITION AND START OVER
3 - START OVER WITHOUT AREA BALANCING
4 - STOP PROGRAM
?
```

The cross section being processed is identified for all messages. In this case, the total area should decrease by 8.6, meaning that the upper and lower areas should each decrease by 4.3. However, the amount of area below the wing in the previous cross section is 0.042 so the closest possible balance would decrease the lower area by 0.042 and place the rest of the
decrease above the wing. As a result of this preliminary check, one of four options may be taken.

Under the first option, the program attempts to process the cross section and, after doing so, prints a message indicating why the balancing failed (during what part of processing) and information about the balance that was achieved.

There are four possible reasons why balancing failed, two indicating that the program has run out of area below the wing — "BALANCING FAILED BECAUSE FUSELAGE SHIFTED TOO HIGH" and "BALANCING FAILED BECAUSE MINIMUM LOWER AREA REACHED" and two indicating that the program has run out of area above the wing — "BALANCING FAILED BECAUSE FUSELAGE SHIFTED TOO LOW" and "BALANCING FAILED BECAUSE MINIMUM UPPER AREA REACHED". There are two messages for each case because there are two areas in the processing where failure may be detected. The messages about the fuselage being shifted too high or low indicate that ZC was shifted too far from the wing during processing to produce the required total area. The messages about minimum upper or lower area being reached indicate that after the maximum number of iterations allowed to find ZC, an upper or lower bound for ZC was found and the iterations have run up against one of these bounds. All of these conditions provide the user with the same information about the best balance that was achieved and the same options for continuing processing, as shown in the following example.
Again, the current cross section is identified. The computed changes in upper and lower area are given along with the accuracy (error tolerance criterion) that was computed. Also, the actual areas above and below are given. The first option allows the user to accept these processing results and move on to the next cross section. The remaining options are the same as those for the preliminary check and selecting one of them produces the same results.

Under the second option, the user selects a new key station and/or initial balance condition and area balancing processing is restarted. First, the current balancing conditions are displayed.

Next, the number of the new key station must be entered (as in this case, the key station may remain the same).

Next, the new distribution of area for the key station must be specified by entering the percentage of fuselage area desired above the wing. A hint is given indicating whether the percentage should be increased or decreased (based on the assumption that the
key station has not been changed). The user may choose to change
the key station and leave the requested distribution unchanged.

\( \text{(TRY DECREASING PERCENT AREA ABOVE FOR KEY STATION)} \)

\( \text{ENTER NEW PERCENT AREA ABOVE WING (0.0-100.0)} \)

\( ? 90. \)

The new conditions are displayed and the user responds "YES" if it
is okay to continue. A "NO" answer enables the user to reenter
the balancing conditions.

\( \text{BALANCING WITH KEY STATION NO. 30 X = 325.00} \)
\( \text{PERCENT ABOVE} = 90.00 \quad \text{PERCENT BELOW} = 10.00 \)

\( \text{OKAY TO CONTINUE? (Y/N)} \)

\( ? Y \)

If option 3 is chosen, the program will reprocess the
configuration from the beginning as if area balancing had not been
selected. Therefore, the user must indicate how the fuselage
cross sections should be centered.

\( \text{HOW DO YOU WANT CROSS SECTIONS CENTERED?} \)
\( 1 - \text{TO FOLLOW WING CAMBER} \)
\( 2 - \text{TO FOLLOW CAMBER OF FUSELAGE IN INPUT FILE} \)

\( ? 1 \)
Following a screen erasure, the cross sections are processed.

The last option stops program execution in case the user does not wish to continue processing. The following message is displayed.

```
111 CURRENT X STATION NO. • 15 , X • 150.00
PROGRAM STOPPED BY USER REQUEST
```

WHEN R CANNOT BE COMPUTED FOR A GIVEN CROSS SECTION

The generation of both circular and double lobe shapes involves computing the value of R - the radius of the circle (or main body on the double lobe shape) that produces the shape with the required area. In some cases, the correct value of R cannot be found. This happens because upper and lower bounds on the value of R are computed using geometric criteria and the value of R that produces the correct area does not lie within these limits. In fact, in some cases, the upper limit is computed to be less than the lower limit. Since this represents the case of a cross section that is geometrically impossible to process the program is stopped with the following message.

```
AREA REQUIREMENT CANNOT BE SATISFIED
LOWER BOUND ON R > UPPER BOUND ON R
CENTER OF FUSELAGE PROBABLY TOO FAR FROM WING
OR BAD COCKPIT PARAMETERS
```
This problem has occurred on circular cross sections ahead of or behind the wing where the fuselage was balanced to an extreme and on double lobe cross sections where the combination of parameters controlling the shape produced an impossible situation.

If the limits for the value of R are normal (upper bound greater than lower bound) the iteration scheme to compute R is performed. If, after the maximum number of iterations to find R, the value that produces the desired area has not been found, it is usually because the iteration has been bounded by the geometric limits on R and the correct value of R lies outside of these limits. When this occurs, the following message is displayed.

```
AREA REQUIREMENT NOT SATISFIED WITHIN DESIRED ACCURACY
X STATION NO. 15 X = 150.00
AREA DESIRED = 1839.600 AREA COMPUTED = 2332.059 ACCURACY = 26.77 PERCENT
RADIUS = 21.21760
UPPER BOUND ON RADIUS = 38.19167
LOWER BOUND ON RADIUS = 21.21760
IS THIS AREA ACCEPTABLE? (Y/N)
? N
```

This informs the user of the area produced by the best possible value of R and the percent error between this area and the desired area. It also informs the user of which limit restricted the proper value of R from being computed. If the accuracy is small enough, the user may wish to accept this computation and move on to the next cross section by responding "YES" to "IS THIS AREA ACCEPTABLE?". In this case, the digital description is generated using these values. If the area produced is not acceptable and the cross section is circular, nothing can be done to improve the computation so a "NO" response causes program execution to stop with the following message.

```
PROGRAM STOPPED BY USER REQUEST
```
If the area produced is not acceptable and the cross section
is a cockpit station (double lobe), then it may be possible to
change some of the parameters defining the shape to allow the
correct value of R to be found. In this case, a "NO" response
generates the following prompt.

SELECT AN OPTION
1 - CHANGE PARAMETERS ON THIS SECTION AND TRY AGAIN
2 - RE-ENTER ENTIRE PROFILE (ZMAX) OF COCKPIT AND TRY AGAIN
3 - STOP PROGRAM
? 1

By selecting option 1, the user may change either the value
of ZMAX for this cross section or the value of the constant used
on this cross section (the original value of the constant will be
restored before processing the next cross section). The new value
may be entered by the user or the program can compute the new
value.

If the user chooses to enter a new value for ZMAX, the
program provides the hint that decreasing ZMAX might produce
better results.

WHICH PARAMETER TO CHANGE?
1 - ZMAX
2 - CONSTANT CONTROLLING POSITION OF CENTER OF COCKPIT CIRCLE
? 1

DO YOU WANT TO INPUT CHANGE OR USE PROGRAM DEFAULTS?
1 - INPUT CHANGE
2 - USE PROGRAM DEFAULT
? 1

CURRENT ZMAX = 50.495
ENTER NEW ZMAX
(TRY DECREASING ZMAX)
? 45.
Using this lower value of $Z_{\text{MAX}}$, the program reprocesses the cross section but the results still are not good enough and the following message is displayed.

```
AREA REQUIREMENT NOT SATISFIED WITHIN DESIRED ACCURACY
X STATION NO.  15  X =  150.00
AREA DESIRED = 1829.600 AREA COMPUTED = 1893.514 ACCURACY = 2.93 PERCENT
RADIUS = 19.60002
UPPER BOUND ON RADIUS = 32.69644
LOWER BOUND ON RADIUS = 19.60002
IS THIS AREA ACCEPTABLE? (Y/N) Y
```

Since the accuracy is within 3 percent, the user chooses to accept these computations so that processing can move to the next cross section and the completion message is displayed for this cross section.

```
X STATION NO.  15 COMPLETE
```

The next illustration is an example of the user choosing to use the program default for changing $Z_{\text{MAX}}$. The default option increases or decreases (according to which limit restricted $R$) $Z_{\text{MAX}}$ by ten percent of the value of $R$. 

39
The change produced by the program is not enough so the processing fails again.

At this point, the user decides to enter a new value for the constant so the program gives the hint that increasing the constant may produce better results.
The increased value of 0.95 still is not good enough and the processing fails for the third time.

AREA REQUIREMENT NOT SATISFIED WITHIN DESIRED ACCURACY
X STATION NO. 13 X* 112.50
AREA DESIRED = 1934.580 AREA COMPUTED = 2954.005 ACCURACY = 52.70 PERCENT
RADIUS = 24.00113
UPPER BOUND ON RADIUS = 44.05129
LOWER BOUND ON RADIUS = 24.00113
IS THIS AREA ACCEPTABLE? (Y/N)

The user now chooses to use the program default for changing the constant. The default option increases or decreases (according to which limit restricted R) the constant by 0.1.

SELECT AN OPTION
1 - CHANGE PARAMETERS ON THIS SECTION AND TRY AGAIN
2 - RE-ENTER ENTIRE PROFILE (ZMAX) OF COCKPIT AND TRY AGAIN
3 - STOP PROGRAM

? 1
WHICH PARAMETER TO CHANGE?
1 - ZMAX
2 - CONSTANT CONTROLLING POSITION OF CENTER OF COCKPIT CIRCLE

? 2
DO YOU WANT TO INPUT CHANGE OR USE PROGRAM DEFAULTS?
1 - INPUT CHANGE
2 - USE PROGRAM DEFAULT

? 2
CURRENT CONSTANT = .95
NEW CONSTANT = 1.05

If the value of the constant is too close to the limits of 0.1 and 1.4 to be changed by 0.1, one of the following messages is displayed and the option to select which parameter to change is repeated.
Returning to the example, the new constant of 1.05 still is not satisfactory and the processing fails for the fourth time.

<table>
<thead>
<tr>
<th>AREA REQUIREMENT NOT SATISFIED WITHIN DESIRED ACCURACY</th>
<th>AREA DESIRED</th>
<th>AREA COMPUTED</th>
<th>ACCURACY</th>
<th>RADIUS</th>
<th>UPPER BOUND ON RADIUS</th>
<th>LOWER BOUND ON RADIUS</th>
<th>IS THIS AREA ACCEPTABLE? (Y/N)</th>
</tr>
</thead>
<tbody>
<tr>
<td>X STATION NO. 13 X = 112.50</td>
<td>1934.500</td>
<td>2794.039</td>
<td>44.43</td>
<td>22.83034</td>
<td>42.54745</td>
<td>22.83034</td>
<td>N</td>
</tr>
</tbody>
</table>

At this point, the user decides that the area is still not acceptable and selects to change the entire upper surface profile of the cockpit area.

SELECT AN OPTION
1 - CHANGE PARAMETERS ON THIS SECTION AND TRY AGAIN
2 - RE-ENTER ENTIRE PROFILE (ZMAX) OF COCKPIT AND TRY AGAIN
3 - STOP PROGRAM

The new profile data will be entered in the same manner it was entered the first time. If the user chooses to type in the ZMAX values, the current values will be listed first. If the option to use the crosshair cursor is selected, the current profile will be drawn on the blowup with a dashed line before the cursor appears to accept the new input.
If the user does not wish to attempt to change the parameters to solve the problem, option 3 may be selected to stop program execution. The following message is displayed.

**PROGRAM STOPPED BY USER REQUEST**

**OTHER ERRORS THAT STOP PROGRAM EXECUTION**

The following errors occur during the processing of the cross sections and cause program execution to stop. Along with the error message, information identifying the cross section currently being processed is given

*** CURRENT X STATION NO. = ___, X = ____

<table>
<thead>
<tr>
<th>ERROR</th>
<th>EXPLANATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO WING DEFINITION IN INPUT FILE</td>
<td>Self-explanatory (J1=0 in wave-drag file)</td>
</tr>
<tr>
<td>YJCT NOT A VALUE IN YWING ARRAY</td>
<td>YJCT selected is not located at a semi-span station defined in the wing definition</td>
</tr>
<tr>
<td>EXTENSION OF WING TO CENTER HAS AREA &gt; AREA DESIRED FOR FUS</td>
<td>the extended wing definition used up all of the fuselage area from the area distribution.</td>
</tr>
</tbody>
</table>
a cockpit station is being generated and R has been found. When preparing to generate the digital description, it is discovered that both wing surfaces intersect the shape on the upper lobe and this is not allowed. To remedy this, try changing the balancing requested before the cockpit is processed or try changing the cockpit parameters.
REFERENCES


Figure 1. - Circular cross section without area balancing.
SOLUTION

SELECT R AND ZC SO THAT

1. \( A_U + A_L - A_{TOT} \)

2. \( \frac{dA_U}{dx} = \frac{dA_L}{dx} \)

Figure 2. - Circular cross section with area balancing.
Figure 3. - Noncircular shape option.
(a) Cross section located at fuselage station 7.

Figure 4. - Typical cross section plots found on SAVPVF (Example 1).
(b) Cross section located at fuselage station 28.

Figure 4. – Continued.
(c) Cross section located at fuselage station 43.

Figure 4. - Concluded.
Figure 5. - TAPE10 produced by running Example 1.
Figure 6. - TAPE8 produced by running Example 1.
Figure 7. - Graphically defining $Z_{\text{MAX}}$ (Example 2).

(a) Before user input.
(b) During user input.

Figure 7. - Continued.
(c) After user input.

Figure 7. - Concluded.
(a) Cross section located at fuselage station 8.

Figure 8. - Typical cross-section plots found on SAVPVF (Example 2).
(b) Cross section located at fuselage station 12.

Figure 8. - Continued.
X STATION NO. 16, X = 160.0

(c) Cross section located at fuselage station 16.

Figure 8. - Continued.
X STATION NO. 30, X = 325.0

(d) Cross section located at fuselage station 30.

Figure 8. - Concluded.
Figure 9. - TAPE10 produced by running Example 2.
Figure 10. - TAPES produced by running Example 2.
A computer code, XSECT, has been developed to generate fuselage cross sections from a given area distribution and wing definition. The cross sections are generated to match the wing definition while conforming to the area requirement. An iterative procedure is used to generate each cross section. Fuselage area balancing may be included in this procedure if desired.

The code is intended as an aid for engineers who must first design a wing under certain aerodynamic constraints and then design a fuselage for the wing such that the constraints remain satisfied. This report contains the information necessary for accessing and executing the code, which is written in FORTRAN to execute on the Cyber 170 series computers (NOS operating system) and produces graphical output for a Tektronix 4014 CRT. The LRC graphics software is used in combination with the interface between this software and the PLOT 10 software.

<table>
<thead>
<tr>
<th>Key Words (Suggested by Author(s))</th>
</tr>
</thead>
<tbody>
<tr>
<td>wave-drag</td>
</tr>
<tr>
<td>supersonic aerodynamics</td>
</tr>
<tr>
<td>supersonic area rule</td>
</tr>
</tbody>
</table>

Security Classif. (of this report) Unclassified

Security Classif. (of this page) Unclassified

No. of Pages 64

Price A04