

**NASA TECHNICAL NOTE**



**NASA TN D-6766**

c.1

NASA TN D-6766

**LOAN COPY: RETURN  
AFWL (DOUL)  
KIRTLAND AFB, N. M.**

0133604



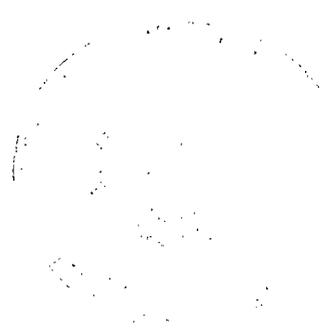
TECH LIBRARY KAFB, NM

**FORTRAN PROGRAM FOR GENERATING  
A TWO-DIMENSIONAL ORTHOGONAL MESH  
BETWEEN TWO ARBITRARY BOUNDARIES**

*by William D. McNally*

*Lewis Research Center*

*Cleveland, Ohio 44135*





0133604

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                               |                                                                |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------------|
| 1. Report No.<br><b>NASA TN D-6766</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 2. Government Accession No.                                   | 3. Recipient's Catalog No.                                     |
| 4. Title and Subtitle<br><b>FORTRAN PROGRAM FOR GENERATING A TWO-DIMENSIONAL ORTHOGONAL MESH BETWEEN TWO ARBITRARY BOUNDARIES</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                               | 5. Report Date<br><b>April 1972</b>                            |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                               | 6. Performing Organization Code                                |
| 7. Author(s)<br><b>William D. McNally</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                               | 8. Performing Organization Report No.<br><b>E-6644</b>         |
| 9. Performing Organization Name and Address<br><b>Lewis Research Center<br/>National Aeronautics and Space Administration<br/>Cleveland, Ohio 44135</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                               | 10. Work Unit No.<br><b>764-74</b>                             |
| 12. Sponsoring Agency Name and Address<br><b>National Aeronautics and Space Administration<br/>Washington, D.C. 20546</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                               | 11. Contract or Grant No.                                      |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                               | 13. Type of Report and Period Covered<br><b>Technical Note</b> |
| 15. Supplementary Notes                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |                                                               | 14. Sponsoring Agency Code                                     |
| 16. Abstract<br><p>A FORTRAN IV program is presented which computes and plots coordinates for a two-dimensional orthogonal mesh in the region between the walls of a flow channel. The program is designed for a channel containing a body about which flow passes and which spans the channel from one wall to the other. However, the condition that the channel contain an immersed body can be easily removed from the program. Input to the program consists of spline points of the channel walls and the body geometry. Output includes printed and plotted coordinates of the generated orthogonal mesh and angles of the mesh with the horizontal plane.</p> |                                                               |                                                                |
| 17. Key Words (Suggested by Author(s))<br><b>Orthogonal mesh; Channel flow; Finite-difference mesh; Turbomachinery analysis; Meridional plane flow</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                | 18. Distribution Statement<br><b>Unclassified - unlimited</b> |                                                                |
| 19. Security Classif. (of this report)<br><b>Unclassified</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | 20. Security Classif. (of this page)<br><b>Unclassified</b>   | 21. No. of Pages<br><b>45</b>                                  |
|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |                                                               | 22. Price*<br><b>\$3.00</b>                                    |



# CONTENTS

|                                                    | Page |
|----------------------------------------------------|------|
| SUMMARY . . . . .                                  | 1    |
| INTRODUCTION . . . . .                             | 2    |
| DESCRIPTION OF METHOD . . . . .                    | 3    |
| USE OF PROGRAM . . . . .                           | 10   |
| Numerical Examples . . . . .                       | 10   |
| Input . . . . .                                    | 14   |
| Dictionary of input variables . . . . .            | 15   |
| Special instructions for preparing input . . . . . | 16   |
| Output . . . . .                                   | 18   |
| DESCRIPTION OF PROGRAM . . . . .                   | 20   |
| Description of Subroutines . . . . .               | 20   |
| Subroutine INPUT . . . . .                         | 20   |
| Subroutine MESH . . . . .                          | 20   |
| Subroutine OUTPUT . . . . .                        | 21   |
| Subroutine OUPLOT . . . . .                        | 21   |
| Subroutine ROOT . . . . .                          | 21   |
| Subroutine CROSS . . . . .                         | 22   |
| Subroutine SPLINE . . . . .                        | 22   |
| Subroutine SPLINT . . . . .                        | 22   |
| Dictionary of Variables . . . . .                  | 23   |
| Program Listing . . . . .                          | 27   |
| CONCLUDING REMARKS . . . . .                       | 37   |
| APPENDIX - LEWIS LIBRARY SUBROUTINES . . . . .     | 39   |
| REFERENCES . . . . .                               | 43   |

# FORTRAN PROGRAM FOR GENERATING A TWO-DIMENSIONAL ORTHOGONAL MESH BETWEEN TWO ARBITRARY BOUNDARIES

by William D. McNally

Lewis Research Center

## SUMMARY

A FORTRAN IV program is presented which computes and plots the coordinates for a two-dimensional orthogonal mesh in the region between the walls of a flow channel. The program is designed for a channel containing a body about which flow passes and which spans the channel from one wall to the other. However, the condition that the channel contain an immersed body can be easily removed from the program.

Input to the program is brief. It consists of arrays of point coordinates describing the two channel walls and the front and rear edges of the submerged body. Information is also given for the number of orthogonal mesh lines desired in the throughflow and normal directions. Output consists of the coordinates of the generated orthogonal mesh points as well as the angles of the mesh at these points with the horizontal plane. One of the routines in the program also generates a microfilm plot of the channel and the submerged body, as well as a plot of the orthogonal mesh. All routines in the program except the plot routine are in general FORTRAN IV code and could be easily transferred to other computing equipment. The plot routine makes use of Lewis in-house subroutines and equipment and would require recoding at another facility.

The geometry of each of the channel walls input to the program must be smooth enough that it can be easily fit with a cubic spline curve; that is, it should not contain any sharp discontinuities in slope. The space between these walls is divided into equal increments across the channel, and spline curves through these points constitute the "horizontal", or throughflow, orthogonal lines. The "vertical", or normal, orthogonals are obtained by means of a simple "predictor-corrector" type process in which normals to adjacent orthogonal lines are used. The program is restricted to channels which are essentially horizontal; that is, at no point should either of the channel walls be inclined more than  $45^{\circ}$  to the horizontal.

This report includes a listing of the program, with an explanation of the input required and the output generated. Numerical examples are also included. Running times are under 0.1 minute on IBM 7094 or 360-67 equipment.

## INTRODUCTION

Finite difference solutions to partial differential fluid flow equations require that a finite difference mesh be placed upon the solution region. And in most cases, the terms in the finite difference equations will be simplified if this mesh is orthogonal. This report describes a FORTRAN IV program which computes and plots coordinates for a two-dimensional orthogonal mesh in the region between the walls of a flow channel.

The primary application for this program at NASA Lewis Research Center is for analytical solutions for flow in turbomachinery passages containing rotor or stator blade rows. For this reason, the program is designed for a channel containing a solid body about which the flow passes and which spans the channel from one wall to the other. The input to the program is designed so that the mesh spacing can be varied in the regions upstream of the immersed blade row, in the general space occupied by the blade row, and downstream of it. In this way a region of particular interest can be covered with a finer mesh than is used in the other two regions.

This "immersed body" feature of the program can be easily removed, however, since the only way the body geometry affects the generated mesh is by designating the points of separation of the three mesh-size regions. The resulting program would have wide application in many fields where finite difference solutions are desired between the walls of an arbitrarily shaped channel.

Input to the program consists primarily of spline points describing the channel walls and the edges of the submerged body. These arrays of points must be smooth enough that they are capable of being accurately fitted with cubic spline curves. Therefore, they cannot contain any sharp discontinuities in slope.

The program divides the space between the walls into equal increments across the channel. Spline curves are then fit through the resulting points to obtain the throughflow, or streamwise or "horizontal", orthogonals. The normal, or "vertical", orthogonals are obtained by means of a simple "predictor-corrector" technique in which normals to adjacent orthogonal lines are used. This technique is analogous to the second-order Runge-Kutta method for solving ordinary differential equations known as the improved Euler method or Heun's method (ref. 1).

The program is restricted to channels which are essentially horizontal; that is, the channel walls should not be inclined by more than about  $45^{\circ}$  with the horizontal plane. For axial turbomachinery this is not a serious restriction. This limitation could be removed in a more general recoding of the program. The program could also be extended to two-dimensional geometries with walls having noncontinuous slopes, as well as to some three-dimensional geometries.

Output consists of printed coordinates of the generated orthogonal mesh points, as well as the angles of the mesh lines at these points with the horizontal plane. One of the

program's subroutines generates a microfilm plot of the channel input geometry and of the calculated orthogonal mesh. All routines in the program except the plot routine are in general FORTRAN IV code and could be easily transferred to other computing equipment. The plot routine makes use of Lewis in-house subroutines and equipment and would require recoding at another facility. Run times for average mesh sizes are under 0.1 minute on IBM 7094 or 360-67 equipment.

This report includes a description of the method used to generate the orthogonal mesh, as well as a description of each of the subroutines in the program. A complete listing is also included. The input and output are thoroughly described, and numerical examples are given. The Lewis plot routines are described in the appendix.

## DESCRIPTION OF METHOD

The program makes use of four principal subroutines - INPUT, MESH, OUTPUT, and OUPLOT. The bulk of the work is done in MESH, which uses four auxiliary routines - ROOT, CROSS, SPLINE, AND SPLINT. About 400 lines of code are used in these eight routines. The calling arrangement is illustrated in figure 1.

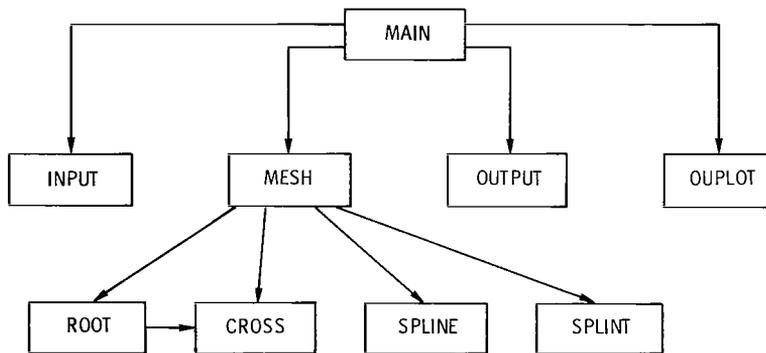


Figure 1. - Calling sequence of subroutines.

The INPUT routine reads in data, the OUTPUT routine prints results, and OUPLOT makes a microfilm plot of the generated mesh. So the bulk of the mesh generation algorithm is contained in MESH, which is described in the following paragraphs.

The mesh generation algorithm begins with input data similar to those shown in figure 2. Two sets of coordinates (ZBOT, RBOT and ZTOP, RTOP) describe the upper and lower channel walls, and two others (ZLE, RLE and ZTE, RTE) describe the front and rear of the confined body. The coordinates defining the body are not used as part

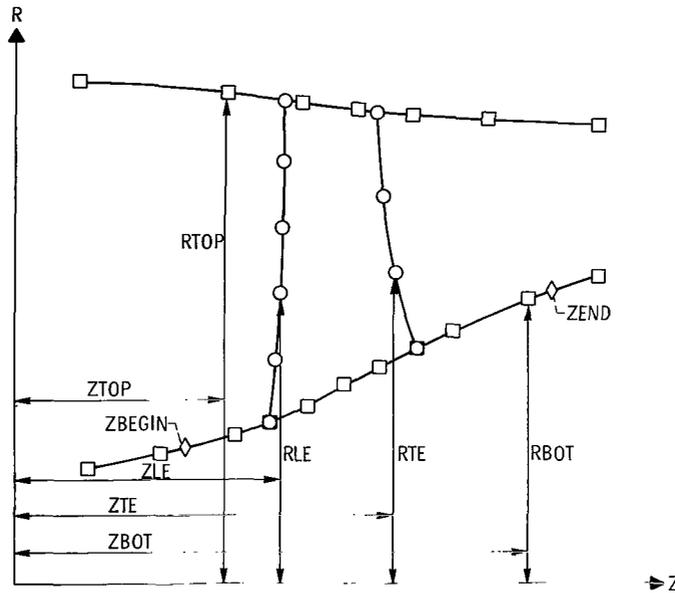


Figure 2. - Schematic representation of input variables.

of the mesh generation algorithm. It is the shape of the channel walls which fully determines the orthogonal mesh. The intersections of the edges of the body with the bottom channel wall ( $ZLE(1)$  and  $ZTE(1)$ ) are the points which separate the three regions of varying mesh size. These regions correspond generally to the areas upstream of the body, on the body, and downstream of it. The Z-coordinates,  $ZBEGIN$  and  $ZEND$ , define the extremes of the orthogonal mesh where its left and right boundaries intersect the bottom channel wall. These two parameters, along with the number of mesh spaces requested in each region, determine the mesh spacing upstream and downstream of the central body. The two points where the left and right mesh boundaries intersect the upper channel wall must remain arbitrary, since they will be defined by the mesh generation algorithm.

The MESH subroutine begins by extending lines vertically from each of the input points on the lower boundary to the upper channel wall (see fig. 3). Each of these lines is then divided into equal increments, the number depending upon the number of mesh lines desired in the horizontal, or streamwise, direction. The resulting array of points along the radial lines is called RRAD. Spline curves are fit through the points in the RRAD array to give the horizontal orthogonals. These are shown in figure 4.

The procedure for calculating the vertical orthogonal links between adjacent horizontal orthogonals is illustrated in figure 5. This procedure is analogous to the technique for solving ordinary differential equations known as the improved Euler method or Heun's method (ref. 1). In this technique, the solution at the next unknown point is

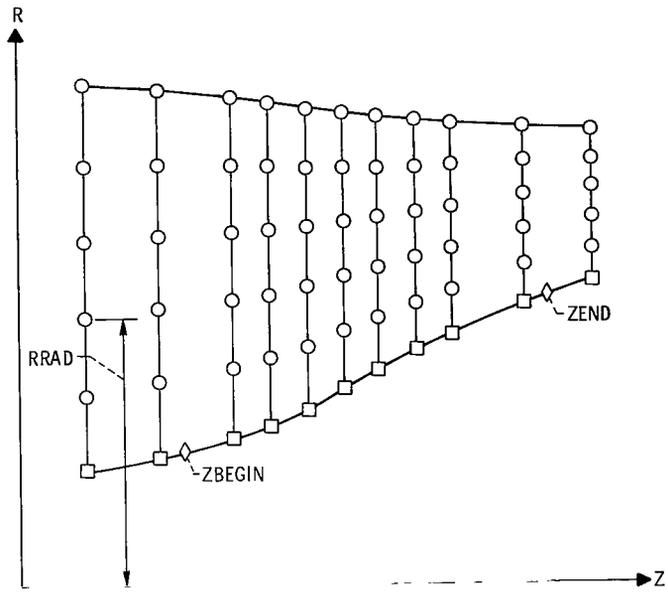


Figure 3. - RRAD coordinates on vertical lines constructed from input points on lower boundary.

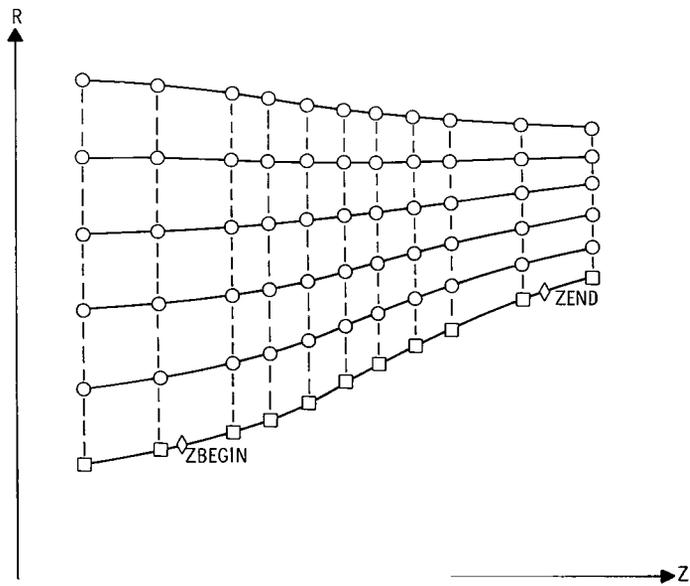


Figure 4. - "Horizontal" orthogonal splines fitted through RRAD points.

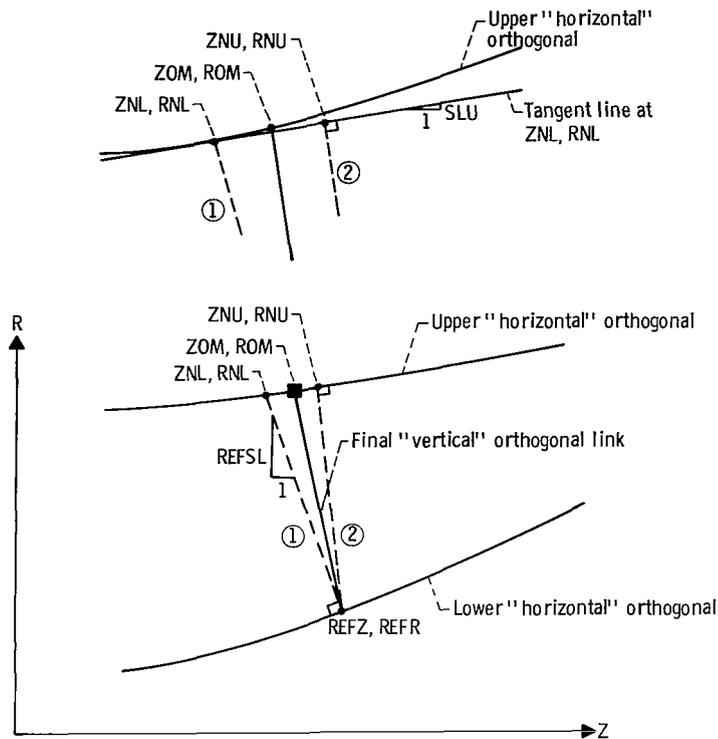


Figure 5. - Calculation procedure for a "vertical" orthogonal link.

first predicted from the solution at a known point by using the Euler method. A second solution is then calculated by using a new slope obtained from the predicted point. The final solution is a corrected combination of these two solutions.

In this program, a similar process is used to construct the vertical links between adjacent horizontal orthogonals. These links are constructed one at a time, moving from left to right between each pair of adjacent horizontal orthogonals (see fig. 6). When all links are calculated for one pair of orthogonals, the algorithm moves vertically to the next pair and again constructs links, moving from left to right.

Each link begins at a known (previously calculated) mesh point on the lower horizontal orthogonal with coordinates  $REFZ, REFR$ . A normal to this point is constructed (line ① in fig. 5), and the intersection  $(ZNL, RNL)$  is calculated on the upper orthogonal curve. The subroutines `ROOT` and `CROSS` are used to locate this intersection. (This process will be explained in more detail in the following paragraph.) The slope of the upper orthogonal at the point of intersection  $(ZNL, RNL)$  is computed; this is  $SLU$ . Line ② in figure 5 is then constructed in such a way that it is perpendicular to the  $SLU$  line and also passes through the reference point  $(REFZ, REFR)$ . The coordinates of the intersection of line ② with the  $SLU$  line are calculated; these are  $ZNU$  and  $RNU$ . This point is a small distance away from the upper orthogonal curve due to

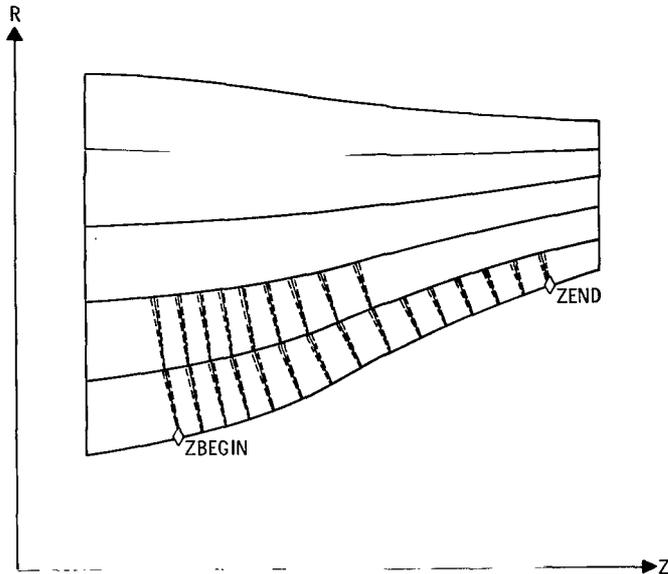


Figure 6. - Process for generating "vertical" orthogonal links.

the change in slope of that curve in moving from point ZNL, RNL to point ZNU, RNU. The final location of the new orthogonal mesh point on the upper curve has a Z-coordinate (ZOM) which is the average of ZNL and ZNU. The corresponding R-coordinate (ROM) is calculated by spline interpolation, using the value of ZOM and coordinates in the RRAD array. The process of constructing vertical links is continued until the upper channel wall is reached by all vertical orthogonals. This completes the generation of the mesh.

Let us return to the procedure by which the intersection of line ① in figure 5 with an upper horizontal orthogonal curve is calculated. The MESH subroutine establishes Z-coordinate bounds (BNDL and BNDR) on the upper orthogonal curve between which the intersection is certain to occur (see fig. 7). These bounds are calculated by using the slope of line ① and an approximate distance between the two horizontal orthogonals. One of these bounds is always equal to REFZ (see fig. 7). The other is established by estimating how far the intersection with the upper orthogonal is away (in the Z-direction) from REFZ and then doubling that distance for a factor of safety. Once the two bounds are set, ROOT is called to find the intersection.

The ROOT subroutine calculates the intersection of a straight line with a curve (the upper horizontal orthogonal in this application). It does this by systematically choosing Z-coordinates, calculating the distance in the R-direction between the straight line (line ① in fig. 7) and the orthogonal curve at each of these coordinates, and converging to the intersection. ROOT begins with a Z-coordinate halfway between the bounds BNDL and BNDR and computes the distance between the corresponding R-coordinates of

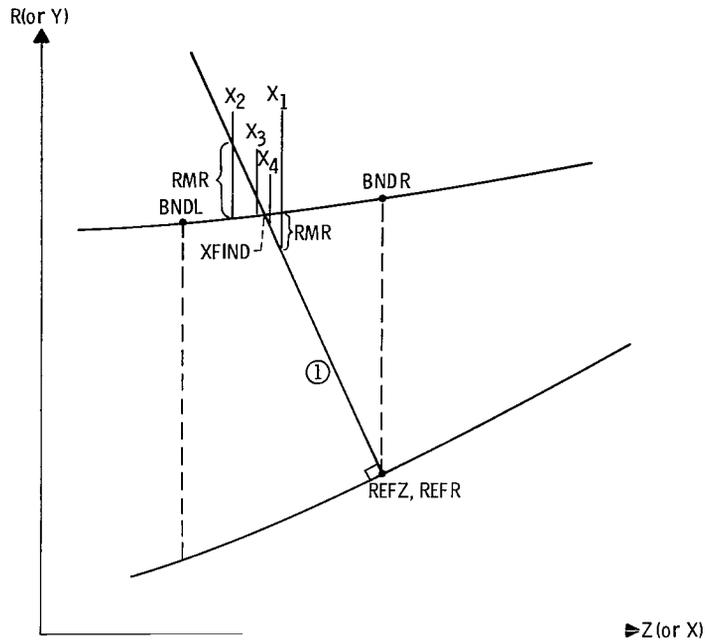


Figure 7. - Intersection of line ① with upper orthogonal by ROOT subroutine.

line ① and the upper curve at the given value of Z. This distance is called RMR (see fig. 7). Based on the sign of RMR, ROOT decides on which side of the chosen Z-coordinate the intersection lies, and moves the left or right bound inward accordingly. The resulting interval is again cut in half and a new RMR computed. This process of dividing the remaining interval is repeated 20 times by the ROOT subroutine. Each time the remaining interval is cut in half, so that after 20 iterations the intersection is accurately located.

CROSS is the routine called 20 times during each call on ROOT. CROSS calculates RMR for a given value of Z (see fig. 8). On the straight line (line ① in fig. 8) normal to the lower orthogonal, the R-coordinate (RLINE) is computed for the given value of Z. For the same Z, the R-coordinate (RCURV) and slope (SL) are computed on the upper orthogonal curve. Note that RLINE can be of greater magnitude than RCURV, as it is for the dashed-line value of Z in figure 8. RMR is computed as the difference between RCURV and RLINE.

An example of a completed orthogonal mesh is shown in figure 9. The body geometry has once again been superimposed on the mesh region in this figure. Notice that it did not affect the generation of the orthogonal mesh, except that the mesh spacing has been varied in the regions forward of the body, on the body, and to the rear of the body. This spacing is controlled by the input.

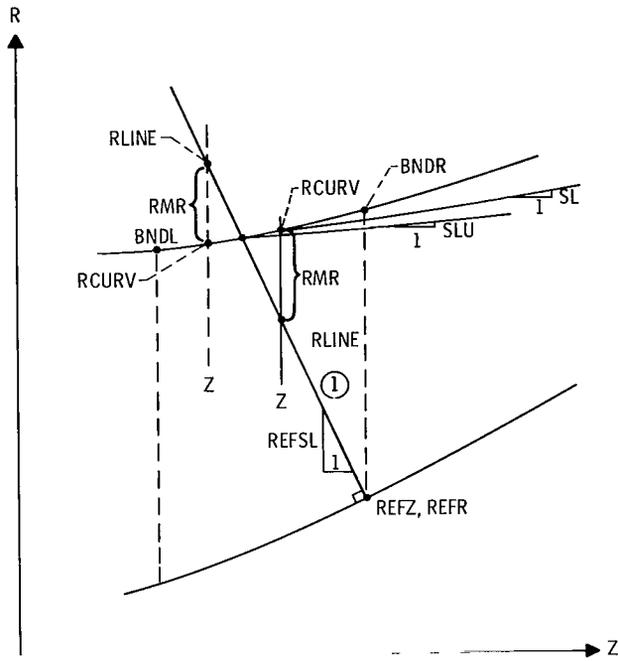


Figure 8. - Procedure to compute RMR by CROSS subroutine.

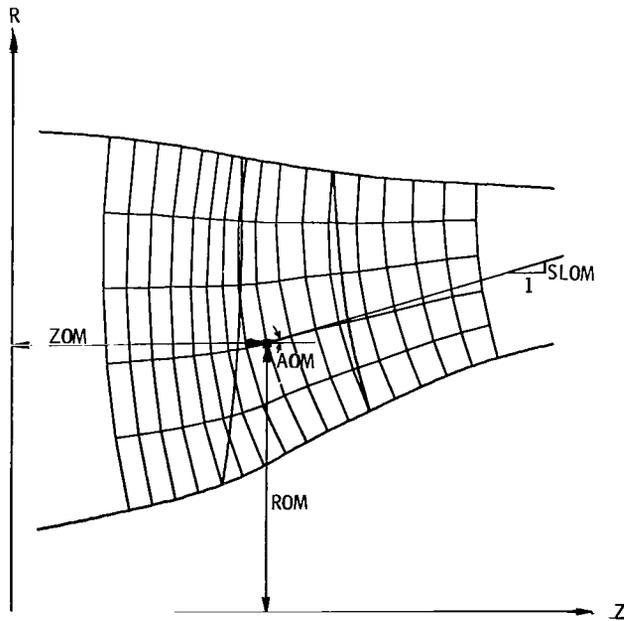


Figure 9. - Completed orthogonal mesh.

## USE OF PROGRAM

The orthogonal mesh program is very easy to use. Input consists of a small number of coordinate arrays describing boundary geometry. These arrays are very easy to assemble. Following sections describe the input and output thoroughly. Numerical examples are given to illustrate what the program does and to give an overview of the input.

### Numerical Examples

Six examples are given which illustrate the range of geometries for which meshes

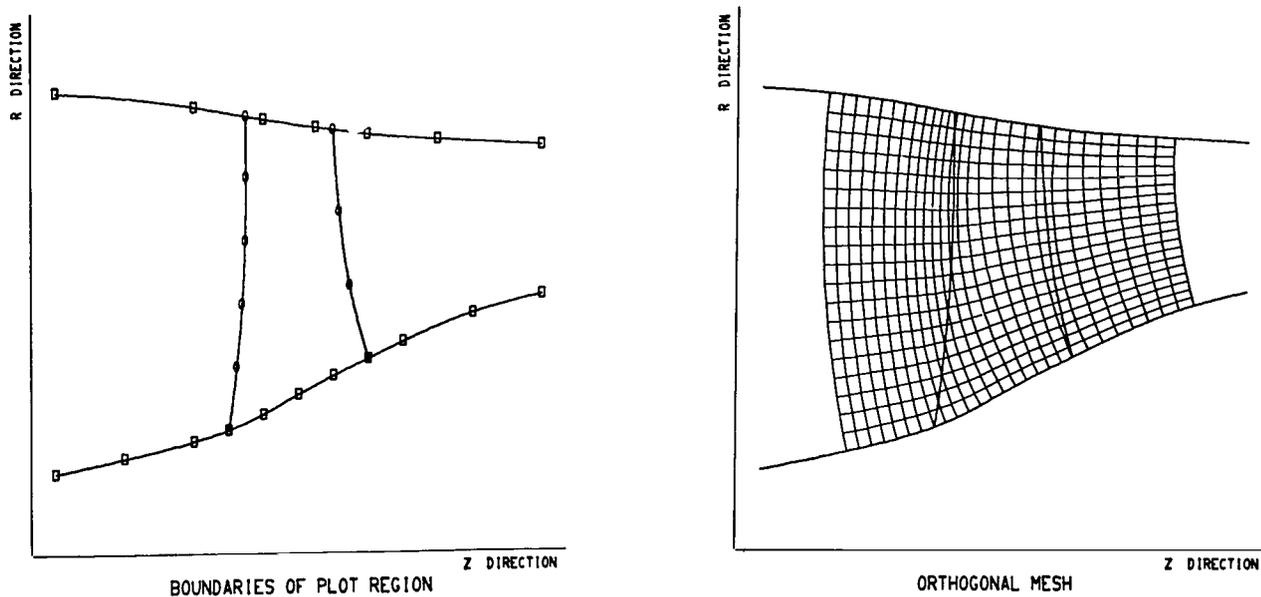


Figure 10. - Converging channel with fine mesh.

can be generated. The first of these examples is the converging channel which has been illustrated in previous figures in the section DESCRIPTION OF METHOD. This channel was shown in figure 9 with a coarse mesh. It is illustrated in figure 10 with a more commonly desired fine mesh suitable for finite difference calculations. The complete input for this example is listed in table I, and the output is shown and discussed in the section Output.

TABLE I. - INPUT FOR CONVERGING-CHANNEL EXAMPLE

ORTHOGONAL MESH TEST CASE- FINE MESH

MESH INPUT DATA AND INPUT ARRAY BOUNDS

| ML        | MT | MZ       | MR | NBOT | NTOP | NLE | NTE |
|-----------|----|----------|----|------|------|-----|-----|
| 8         | 20 | 28       | 20 | 11   | 7    | 6   | 4   |
| ZBEGIN    |    | ZEND     |    |      |      |     |     |
| -0.250000 |    | 0.750000 |    |      |      |     |     |

BOUNDARY INPUT DATA

|            |           |           |          |          |          |          |          |  |
|------------|-----------|-----------|----------|----------|----------|----------|----------|--|
| ZBOT ARRAY |           |           |          |          |          |          |          |  |
| -0.500000  | -0.300000 | -0.100000 | 0        | 0.100000 | 0.200000 | 0.300000 | 0.400000 |  |
| 0.500000   | 0.700000  | 0.900000  |          |          |          |          |          |  |
| RBOT ARRAY |           |           |          |          |          |          |          |  |
| 0.878000   | 0.920000  | 0.968000  | 1.000000 | 1.044000 | 1.100000 | 1.152000 | 1.200000 |  |
| 1.248000   | 1.330000  | 1.383000  |          |          |          |          |          |  |
| ZTOP ARRAY |           |           |          |          |          |          |          |  |
| -0.500000  | -0.100000 | 0.100000  | 0.250000 | 0.400000 | 0.600000 | 0.900000 |          |  |
| RTOP ARRAY |           |           |          |          |          |          |          |  |
| 1.961000   | 1.918000  | 1.882000  | 1.860000 | 1.840000 | 1.825000 | 1.808000 |          |  |

BODY INPUT DATA

|           |              |              |              |              |              |  |
|-----------|--------------|--------------|--------------|--------------|--------------|--|
| ZLE ARRAY |              |              |              |              |              |  |
| 0         | 0.230000E-01 | 0.390000E-01 | 0.470000E-01 | 0.500000E-01 | 0.500000E-01 |  |
| RLE ARRAY |              |              |              |              |              |  |
| 1.000000  | 1.180000     | 1.360000     | 1.540000     | 1.720000     | 1.892000     |  |
| ZTE ARRAY |              |              |              |              |              |  |
| 0.400000  | 0.346000     | 0.317000     | 0.300000     |              |              |  |
| RTE ARRAY |              |              |              |              |              |  |
| 1.200000  | 1.410000     | 1.620000     | 1.853000     |              |              |  |

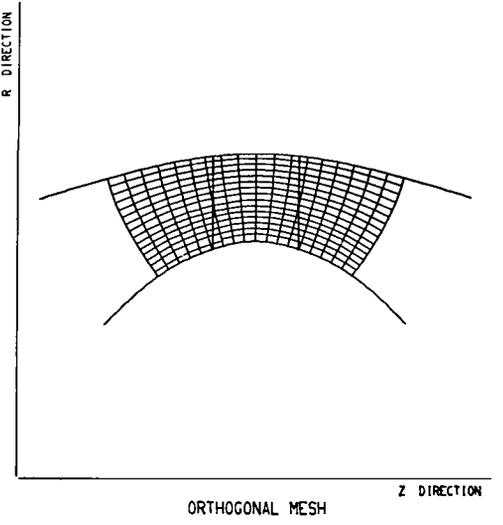
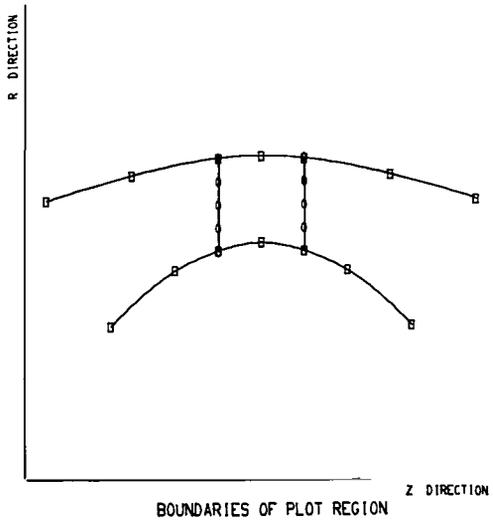
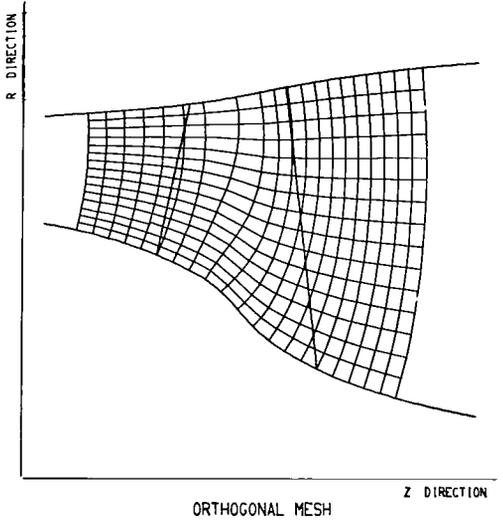
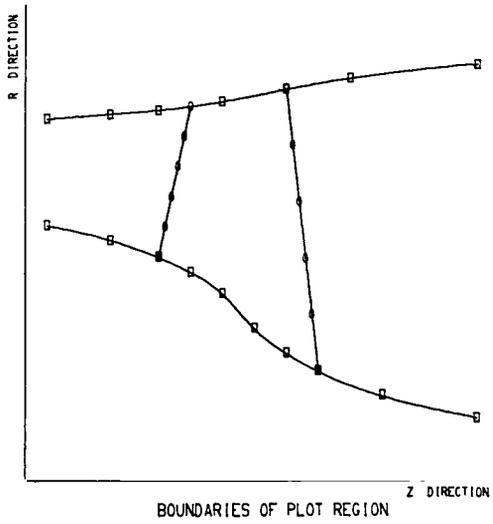
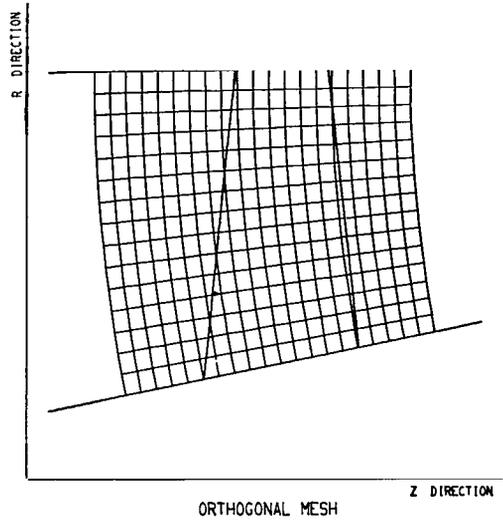
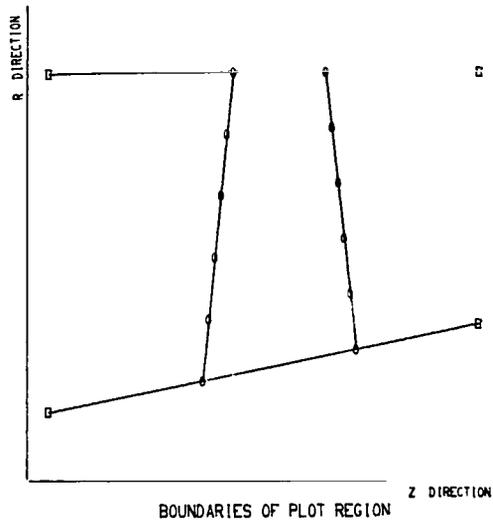


Figure 11. - Input and output plots of typical geometries and meshes.

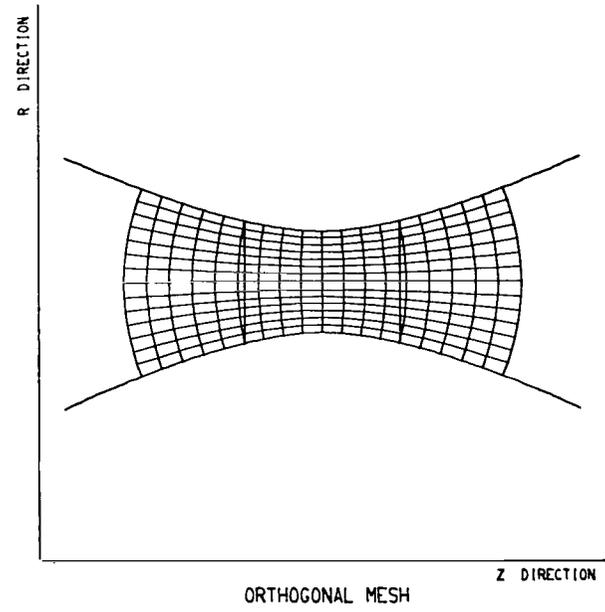
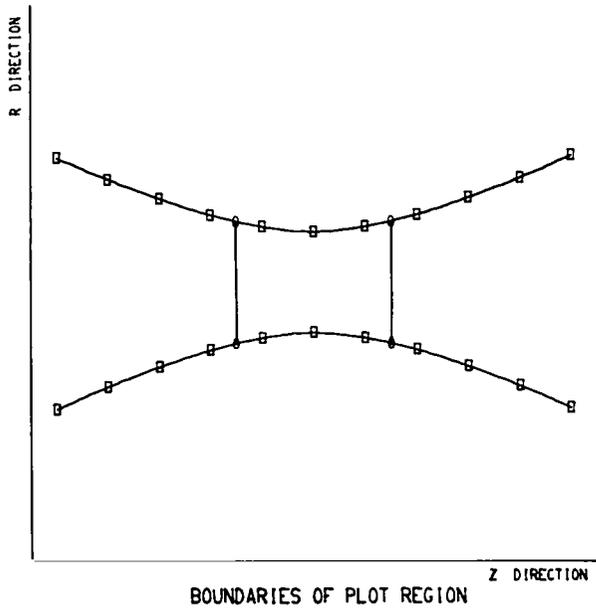
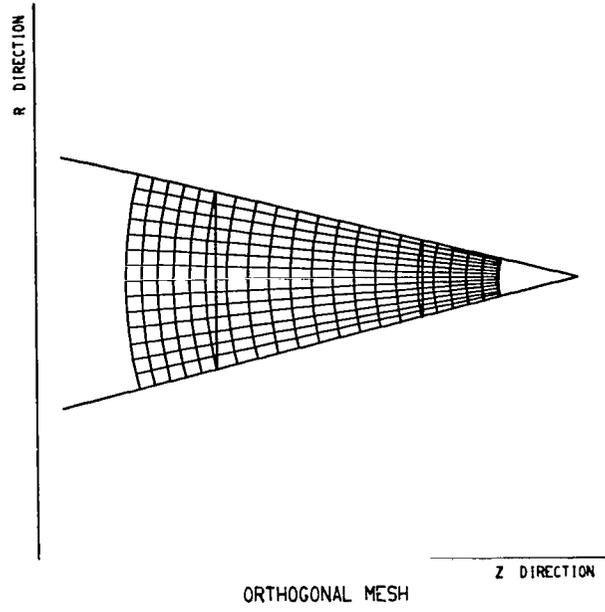
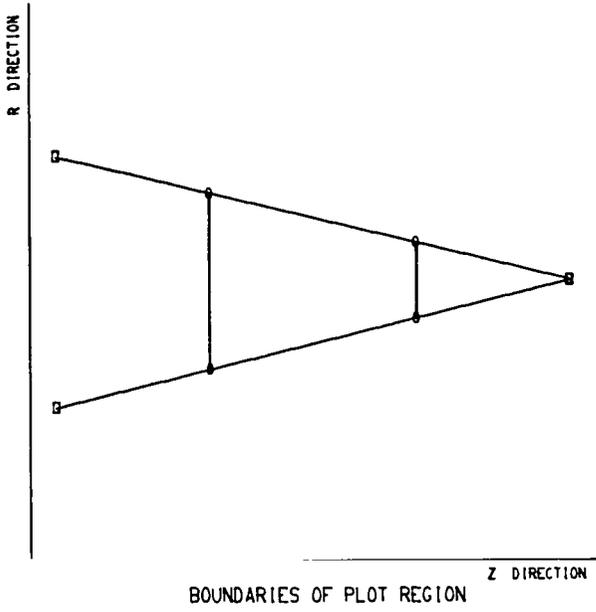


Figure 11. - Concluded.



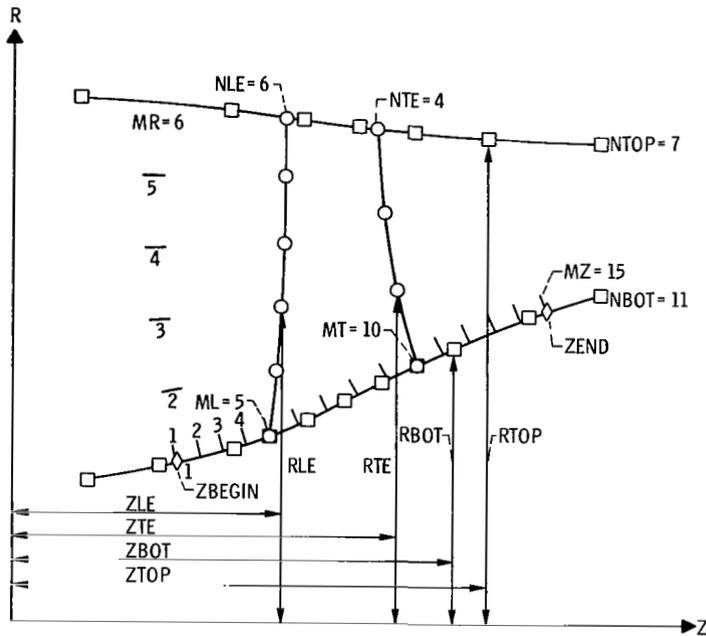


Figure 13. - Complete representation of input variables.

**Dictionary of input variables.** - The following variables are given as input to the program and are illustrated in figure 13.

|      |                                                                                                                     |
|------|---------------------------------------------------------------------------------------------------------------------|
| ML   | number of mesh lines between left boundary of orthogonal mesh (ZBEGIN) and front, or leading, edge of body (ZLE(1)) |
| MR   | number of mesh lines between bottom and top boundaries of orthogonal mesh                                           |
| MT   | number of mesh lines between left boundary of orthogonal mesh (ZBEGIN) and rear, or trailing, edge of body (ZTE(1)) |
| MZ   | number of mesh lines between left and right boundaries of orthogonal mesh (ZBEGIN to ZEND)                          |
| NBOT | number of points in ZBOT and RBOT arrays defining bottom boundary of region                                         |
| NLE  | number of points in ZLE and RLE arrays defining front, or leading, edge of body                                     |
| NTE  | number of points in ZTE and RTE arrays defining rear, or trailing, edge of body                                     |
| NTOP | number of points in ZTOP and RTOP arrays defining top boundary of region                                            |

|        |                                                                                         |
|--------|-----------------------------------------------------------------------------------------|
| RBOT   | array of R-coordinates of input points defining bottom boundary of region               |
| RLE    | array of R-coordinates of input points defining front, or leading, edge of body         |
| RTE    | array of R-coordinates of input points defining rear, or trailing, edge of body         |
| RTOP   | array of R-coordinates of input points defining top boundary of region                  |
| ZBEGIN | Z-coordinate of intersection of first vertical mesh line with bottom boundary of region |
| ZBOT   | array of Z-coordinates of input points defining bottom boundary of region               |
| ZEND   | Z-coordinate of intersection of final vertical mesh line with bottom boundary of region |
| ZLE    | array of Z-coordinates of input points defining front, or leading, edge of body         |
| ZTE    | array of Z-coordinates of input points defining rear, or trailing, edge of body         |
| ZTOP   | array of Z-coordinates of input points defining top boundary of region                  |

Special instructions for preparing input. - The following instructions and restrictions should be observed when preparing input:

- (1) The number of mesh lines in either direction must not exceed 50:

$$MZ \leq 50$$

$$MR \leq 50$$

- (2) The number of input coordinates on any surface must not exceed 50:

$$NBOT \leq 50$$

$$NTOP \leq 50$$

$$NLE \leq 50$$

$$NTE \leq 50$$

(3) The origin of coordinates ( $Z = 0, R = 0$ ) can be placed in any location convenient to the user.

(4) The arrays of coordinates describing body geometry should begin and end with points on the channel walls. If this condition is not met, the program will execute; but mesh spacing will be altered depending upon where these arrays of coordinates begin and end.

(5) The horizontal orthogonals are generated by means of spline curves through points in the RRAD array. The RRAD array, in turn, is a function of the number of points given in the ZBOT array of input. If either channel wall has curvature to it, enough points should be given in ZBOT (5 to 10 at least) to ensure an RRAD array with enough points to guarantee accurate horizontal orthogonals.

(6) No points on the generated orthogonal mesh should be outside the extremes of the input ZBOT and ZTOP arrays. ZBOT and ZTOP should contain values sufficiently outside the limits of ZBEGIN and ZEND to prevent this condition from occurring (see fig. 14). If the orthogonal mesh does pass beyond the bounds of ZBOT or ZTOP, the SPLINE routines will be used for extrapolation; and the ROOT routine may fail to converge, stopping the program.

(7) The array ZTOP should contain Z-coordinates which at least cover the same range as those in the ZBOT array. Otherwise, the vertical lines extended upward from the bottom channel wall (see fig. 3) will have to intersect an extrapolated portion of the upper channel wall. The program will execute under these conditions, but the portion of

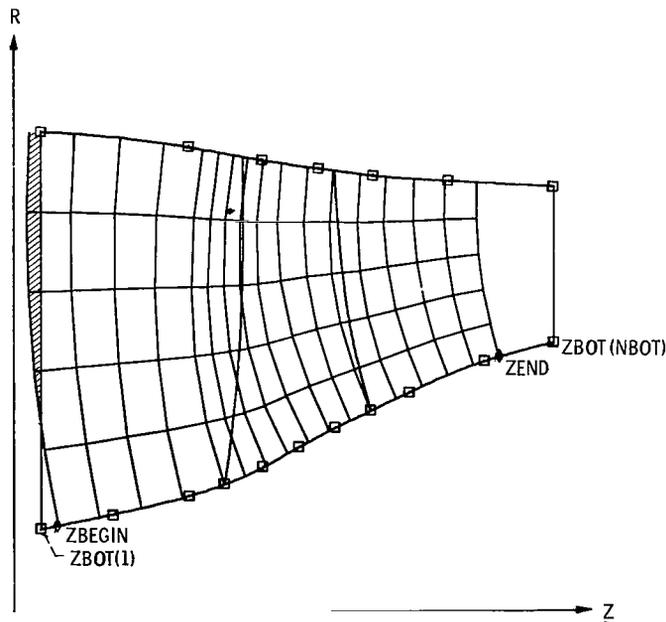


Figure 14. - Example of orthogonal mesh beyond limits of input.

the upper wall generated by spline extrapolation may not conform to what the user would input in its place.

## Output

Output from the program consists of two parts: a computer listing with printed tables of orthogonal mesh coordinates and angles, and microfilm plots of both input geometry and generated orthogonal mesh.

A partial listing of the output from the converging channel of the first example is given in table II. The microfilm plots from this example are given in figure 10.

TABLE II. - PARTIAL OUTPUT FOR CONVERGING-CHANNEL EXAMPLE

```

*****
***                                     ***
*** Z COORDINATES OF                 ***
*** ORTHOGONAL MESH                 ***
***                                     ***
*****
** BOTTOM ROW NU. 1 **
-0.250000      -0.214286      -0.178571      -0.142857      -0.107143      -0.714286E-01      -0.357143E-01      -0.186265E-08
0.333333E-01  0.666667E-01  0.100000E+00  0.133333      0.166667      0.200000      0.233333      0.266667
0.300000      0.333333      0.366667      0.400000      0.443750      0.487500      0.531250      0.575000
0.618750      0.662500      0.706250      0.750000

** ROW NU. 2 **
-0.260833      -0.225334      -0.189948      -0.154663      -0.119461      -0.843637E-01      -0.494692E-01      -0.147291E-01
0.176419E-01  0.506561E-01  0.825723E-01  0.115429      0.148905      0.182907      0.217214      0.251453
0.285571      0.319556      0.353388      0.387076      0.431257      0.475628      0.520157      0.564733
0.609346      0.654000      0.698698      0.743360

** ROW NU. 3 **
-0.270797      -0.235447      -0.200325      -0.165403      -0.130650      -0.960822E-01      -0.619012E-01      -0.280608E-01
0.336329E-02  0.348136E-01  0.664123E-01  0.944879E-01  0.131704      0.166089      0.201299      0.236525
0.271495      0.306189      0.340566      0.374634      0.419175      0.464085      0.509350      0.554738
0.600201      0.645746      0.691380      0.736963

** ROW NU. 4 **
-0.279901      -0.244640      -0.209716      -0.175092      -0.140724      -0.106613      -0.730324E-01      -0.399967E-01
-0.946350E-02  0.210250E-01  0.516590E-01  0.828075E-01  0.115367      0.149780      0.185677      0.221940
0.257829      0.293290      0.328260      0.362737      0.407596      0.452948      0.498884      0.545062
0.591356      0.637774      0.684328      0.730831

** ROW NU. 5 **
-0.288153      -0.252922      -0.218135      -0.183745      -0.149696      -0.115979      -0.828917E-01      -0.505551E-01
-0.208329E-01  0.873441E-02  0.384014E-01  0.685735E-01  0.100214      0.134248      0.170515      0.207763
0.244641      0.280919      0.316533      0.351449      0.396615      0.442296      0.488822      0.535766
0.582854      0.630122      0.677573      0.724988

** ROW NU. 6 **
-0.295559      -0.260303      -0.225595      -0.191376      -0.157583      -0.124195      -0.915096E-01      -0.597631E-01
-0.307617E-01  0.204624E-02  0.266855E-01  0.556769E-01  0.865000E-01  0.119772      0.156050      0.194079
0.232004      0.269149      0.305447      0.340836      0.386316      0.432214      0.479231      0.526873
0.574742      0.622828      0.671148      0.719459

** ROW NU. 7 **
-0.302118      -0.266787      -0.232104      -0.197999      -0.164399      -0.131275      -0.989149E-01      -0.676506E-01
-0.392746E-01  0.113282E-01  0.165253E-01  0.447651E-01  0.743638E-01  0.106621      0.142526      0.181034
0.220003      0.258057      0.295071      0.330962      0.376768      0.422789      0.470187      0.518469
0.567066      0.615934      0.665087      0.714268

** ROW NU. 8 **
-0.307831      -0.272379      -0.237668      -0.203621      -0.170155      -0.137232      -0.105131      -0.742468E-01
-0.464033E-01  0.191368E-01  0.279180E-02  0.352528E-01  0.638549E-01  0.950118E-01  0.130180      0.168830
0.208734      0.247728      0.285478      0.321896      0.368036      0.414106      0.461763      0.510603
0.559878      0.609481      0.659424      0.709441

** ROW NU. 9 **
-0.312691      -0.277080      -0.242293      -0.208250      -0.174860      -0.142078      -0.110172      -0.795773E-01

```

TABLE II. - Concluded, PARTIAL OUTPUT FOR CONVERGING-CHANNEL EXAMPLE

```

*****
***                                     ***
*** R COORDINATES OF ***
*** ORTHOGONAL MESH ***
***                                     ***
*****
** BOTTOM ROW NO. 1 **
0.930968      0.939051      0.947521      0.956470      0.966011      0.976274      0.987504      1.000000
1.013058      1.027663      1.044000      1.062059      1.081053      1.100000      1.118113      1.135385
1.152080      1.168154      1.184075      1.200000      1.221124      1.242108      1.262418      1.281749
1.299921      1.316761      1.332091      1.345825

** ROW NO. 2 **
0.989140      0.996724      1.004676      1.013097      1.022084      1.031783      1.042462
1.053565      1.065995      1.079952      1.095724      1.113166      1.131356      1.149213      1.166127
1.182247      1.197764      1.212897      1.227897      1.247805      1.267857      1.287481      1.306235
1.323919      1.340348      1.355339      1.368737

** ROW NO. 3 **
1.033250      1.039846      1.046616      1.053658      1.061064      1.068917      1.077285      1.086387
1.095760      1.106279      1.118092      1.131538      1.147024      1.164000      1.181366      1.197909
1.213514      1.228397      1.244747      1.256827      1.275424      1.294404      1.313221      1.331306
1.348413      1.364350      1.376924      1.391932

** ROW NO. 4 **
1.085183      1.091098      1.097113      1.103317      1.109796      1.116623      1.123824      1.131554
1.139461      1.148263      1.158171      1.169484      1.182785      1.198147      1.214631      1.230742
1.245815      1.260038      1.273601      1.286765      1.304000      1.321772      1.339654      1.356976
1.373415      1.388774      1.402851      1.415409

** ROW NO. 5 **
1.137561      1.142831      1.148137      1.153564      1.159187      1.165077      1.171239      1.177767
1.184379      1.191699      1.199933      1.209351      1.220501      1.233945      1.249150      1.264641
1.279135      1.292645      1.305639      1.317676      1.333553      1.349975      1.366792      1.383251
1.398932      1.413626      1.427124      1.439166

** ROW NO. 6 **
1.190334      1.194985      1.199621      1.204316      1.209145      1.214167      1.219389      1.224852
1.230330      1.236358      1.243124      1.250875      1.260082      1.271461      1.285061      1.299622
1.313476      1.326268      1.338210      1.349522      1.364001      1.379014      1.394641      1.410135
1.424965      1.438906      1.451742      1.463198

** ROW NO. 7 **
1.243455      1.247507      1.251499      1.255501      1.259581      1.263795      1.268150      1.272657
1.277131      1.282026      1.287509      1.293799      1.301296      1.310662      1.322435      1.335736
1.348830      1.360822      1.371895      1.382258      1.395353      1.408875      1.423200      1.437623
1.451510      1.464610      1.476702      1.487500

** ROW NO. 8 **
1.296877      1.300343      1.303712      1.307049      1.310418      1.313880      1.317414      1.321048
1.324677      1.328515      1.332671      1.337881      1.343880      1.351423      1.361266      1.373046
1.385181      1.396298      1.406644      1.415831      1.427531      1.439928      1.452455      1.465703
1.478557      1.490731      1.501996      1.512063

** ROW NO. 9 **
1.350559      1.353445      1.356204      1.358897      1.361581      1.364304      1.367081      1.369907

*****
***                                     ***
*** ANGLES WITH RESPECT ***
*** TO THE Z DIRECTION ***
*** OF THE ORTHOGONAL MESH ***
***                                     ***
*****
** BOTTOM ROW NO. 1 **
17.563      13.034      13.676      14.484      15.456      16.674      18.303      20.324
22.491      24.854      27.381      29.279      29.857      29.157      27.917      26.902
26.177      25.638      25.478      25.650      25.797      25.351      24.405      23.233
21.841      20.220      18.361

** ROW NO. 2 **
11.484      11.849      12.373      13.051      13.882      14.885      16.247      17.986
19.418      22.070      24.413      26.774      28.046      28.027      26.870      25.748
24.871      24.263      23.992      24.067      24.386      24.151      23.341      22.256
20.949      19.410      17.627

** ROW NO. 3 **
10.446      10.715      11.132      11.692      12.392      13.229      14.345      15.822
17.509      19.442      21.580      23.927      25.842      26.489      25.786      24.561
23.589      22.878      22.496      22.475      22.869      22.846      22.203      21.209
19.991      18.539      16.840

** ROW NO. 4 **
9.442      9.625      9.944      10.396      10.977      11.682      12.580      13.817
15.265      16.976      18.900      21.046      23.309      24.580      24.519      23.331
22.271      21.477      20.987      20.870      21.251      21.441      20.992      20.093
18.908      17.607      15.998

** ROW NO. 5 **
5.465      8.569      8.800      9.153      9.625      10.210      10.931      11.950
13.180      14.670      16.382      18.316      20.536      22.354      22.903      22.046
20.907      20.040      19.458      19.251      19.547      19.940      19.694      18.908
17.881      16.615      15.100

** ROW NO. 6 **
7.509      7.542      7.692      7.955      8.326      8.891      9.378      10.202
11.235      12.515      14.021      15.746      17.752      19.885      20.984      20.671
19.486      18.559      17.907      17.613      17.813      18.352      18.307      17.656
16.730      15.564      14.148

** ROW NO. 7 **
6.509      6.536      6.612      6.792      7.071      7.444      7.903      8.553
9.410      10.498      11.809      13.333      15.125      17.263      18.821      19.046
17.999      17.024      16.326      15.952      16.065      16.686      16.836      16.338
15.517      14.456      13.143

** ROW NO. 8 **
5.636      5.547      5.555      5.658      5.852      6.131      6.487      6.984
7.685      8.601      9.731      11.067      12.654      14.595      16.483      17.178
14.435      15.429      14.701      14.267      14.303      14.953      15.290      14.957
14.245      13.292      12.087      10.636

** ROW NO. 9 **

```

## DESCRIPTION OF PROGRAM

In this section the general characteristics of the program and each of its subroutines are given, as well as a dictionary of all the program variables and a complete program listing.

The coding of the program is done entirely in FORTRAN IV. The Lewis library routines which are used in the OUPLOT subroutine are described in the appendix.

Several labeled commons are used throughout the program:

/INPU/ contains the variables given as input by the user

/CALC/ contains more calculated constants and the output coordinates and angles of the orthogonal mesh

/CROS/ contains the variables transferred between the MESH and CROSS subroutines

Several conventions are used in the naming of variables. In any variable name, a Z or R usually refers to the Z- and R-coordinates. OM refers to a coordinate or variable on the orthogonal mesh. BOT and TOP refer to the bottom and top channel walls; and LE and TE refer to the leading, or upstream, edge and trailing, or downstream, edge of the body, respectively. Slopes have the letters SL in their variable names, and SD refers to second derivatives. The letters TEM refer to a temporary storage variable for some quantity. In the OUPLOT routine a variable containing the letters PLT is always an array to be plotted. In the general routines, ROOT, SPLINE, and SPLINT, X's and Y's will always be replaced by Z- and R-coordinates from the calling routine.

The program is arranged presently to handle a mesh of up to 50 points by 50 points. It is being run at Lewis on the IBM 7094/7044 direct coupled system with a 32 768-word core (77777<sub>(8)</sub>). The total program storage requirement is 47074<sub>(8)</sub> of which 30000<sub>(8)</sub> is used in storage of variables. The special Lewis plot routines use about 3000<sub>(8)</sub> words of storage.

### Description of Subroutines

The following sections describe the subroutines which comprise the program. Further information on MESH, CROSS, and ROOT can be found in the section DESCRIPTION OF METHOD.

Subroutine INPUT. - INPUT reads in the input data describing the mesh, the channel walls, and the body geometry. It then calculates several constants describing the mesh.

Subroutine MESH. - MESH is the principal routine in the program. It calculates the coordinates of the orthogonal mesh while making use of four other routines - ROOT, CROSS, SPLINE, and SPLINT.

MESH first estimates the horizontal, or left-to-right, orthogonal by dividing the space between channel walls into streamtubes of approximately equal height (see fig. 4). These tubes are then intersected by vertical, or bottom-to-top, orthogonal links by means of an improved Euler method technique. Each of these links is the average between a line which is normal to the bottom orthogonal (of an adjacent orthogonal pair) and another which is approximately normal to the top orthogonal (see fig. 5).

Axial distance between vertical orthogonals is determined by the number of mesh lines requested in the following three regions: upstream (to the left) of the body (ML mesh lines from ZBEGIN to ZLE(1)), on the body (MT - ML mesh lines from ZLE(1) to ZTE(1)), and downstream of the body (MZ - MT mesh lines from ZTE(1) to ZEND) (see fig. 13). MR controls the number of horizontal orthogonals, which is the same in all three regions. Note that a vertical orthogonal line which begins within the confines of the body may pass across the front or rear edge of the body as the line is generated in the radial direction.

Further details on the operation of MESH are given in the section DESCRIPTION OF METHOD.

Subroutine OUTPUT. - OUTPUT prints the coordinates (ZOM, ROM) of the orthogonal mesh and the angles (AOM) of the horizontal orthogonals with the true horizontal plane. These values are printed at each mesh point.

Subroutine OUPLOT. - OUPLOT makes two different plots on microfilm for each set of input data. The first plot shows the input channel and body geometry. The second plot shows the generated orthogonal mesh within the channel region (see fig. 10).

OUPLOT makes use of six routines from the Lewis in-house library. These six - LRMRGN, LRANGE, LRGRID, LRCHSZ, LRLEGN, and LRCURV - are described in the appendix.

OUPLOT first interpolates on the four boundaries to be plotted - bottom and top channel walls and front and rear edges of body - to obtain enough points so that these boundaries will plot as smooth curves. It then examines the range of values on these curves to determine the limits of the two plots.

OUPLOT then calls the six library routines to accomplish the microfilm plotting. A margin is set on the microfilm (LRMRGN), the range of plotted points is given (LRANGE), and grid lines and axis labeling are specified (LRGRID). Titles are written on the plots by LRLEGN, and finally the desired curves are plotted by calls on LRCURV.

Subroutine ROOT. - ROOT is a general routine which locates the point within an interval where a function is equal to a given value. In this program the function value is RMR (see figs. 7 and 8), and the given value ROOT searches for is  $RMR = 0$ .

ROOT calls another routine (FUNCT) to compute the function of the independent variable. In this program CROSS is always called as the function routine, and it calculates and returns RMR to ROOT.

ROOT begins with a given interval (BNDL to BNDR) in which the intersection (causing RMR to equal zero) is known to occur. It then divides the interval in half and checks which half contains the intersection. (The check is performed by using the sign of RMR.) The resulting half interval is again divided and checked at the midpoint. By repeating this process 20 times the intersection point is located to within  $1/(2^{20})$  of the original interval.

ROOT contains an error message - ROOT HAS FAILED TO CONVERGE IN THE GIVEN INTERVAL, which is given if RMR does not approach zero to within the given tolerance, TOLER, in 20 iterations. TOLER is set in the MESH routine to 1/2 percent of the distance between two adjacent horizontal orthogonals.

More information on the ROOT routine is given in the section DESCRIPTION OF METHOD.

Subroutine CROSS. - CROSS is called by ROOT and, in a special case, by MESH. When called by MESH, CROSS computes only RCURV (see fig. 8) in the situation where the vertical orthogonal link is truly vertical or radial. When called by ROOT, CROSS calculates RMR for a given value of the Z-coordinate between BNDL and BNDR. It also returns the slope, SL, through ROOT to the MESH routine.

CROSS uses equations of a cubic spline curve to compute RCURV and SL on the upper horizontal orthogonal. The spline curve information is transmitted to CROSS from MESH in the /CROS/ common.

More information on CROSS is given in the section DESCRIPTION OF METHOD.

Subroutine SPLINE. - SPLINE fits a cubic spline curve to the X- and Y-coordinates supplied to it. It solves the tridiagonal matrix equation of reference 2 to obtain coefficients for the piecewise cubic polynomial function of the spline curve. Slopes and second derivatives at the spline points are returned to the calling subroutine. SPLINE uses the end condition that the second derivative at either end point is one-half that at the adjacent spline point.

Subroutine SPLINT. - SPLINT interpolates on a cubic spline curve. It uses the same algorithm as SPLINE to fit a curve to the X and Y input points. It then uses the interpolation equations of reference 2 to obtain coordinates and derivatives at given intermediate points.

SPLINT can also be used to extrapolate, although the results are questionable. They become more inaccurate the farther the extrapolated point is from the end point and the higher the curvature of the spline curve at the end point. When extrapolation is done by SPLINT, the message SPLINT USED FOR EXTRAPOLATION is printed, as well as coordinates of the spline curve points and extrapolated points.

## Dictionary of Variables

The following variables are used throughout the program. The input variables are defined in the section Dictionary of input variables.

|        |                                                                                                          |
|--------|----------------------------------------------------------------------------------------------------------|
| A      | temporary storage variable in SPLINE and SPLINT                                                          |
| AOM    | array of angles made by horizontal orthogonal mesh lines with Z-axis at each point of mesh (fig. 9), deg |
| B      | temporary storage variable in SPLINE and SPLINT                                                          |
| BNDL   | left bound on Z- or X-coordinate in MESH and ROOT (figs. 7 and 8)                                        |
| BNDR   | right bound on Z- or X-coordinate in MESH and ROOT (figs. 7 and 8)                                       |
| C      | temporary storage variable in SPLINE and SPLINT                                                          |
| D      | temporary storage variable in SPLINE and SPLINT                                                          |
| DEL R  | incremental distance in R-direction                                                                      |
| DEL Z  | incremental distance in Z-direction                                                                      |
| DFX    | gradient of R with respect to Z along a horizontal orthogonal in ROOT                                    |
| DMDZ   | expansion distance to size of plot in OUPLOT                                                             |
| DYDX   | gradient of Y with respect to X in SPLINE                                                                |
| DYDXNT | interpolated gradient of Y with respect to X in SPLINT                                                   |
| D2YDX2 | second derivative of Y with respect to X in SPLINE and SPLINT                                            |
| EOP    | end of plot indicator in OUTPLOT                                                                         |
| FUNCT  | interval subroutine called in ROOT                                                                       |
| FX     | function calculated in ROOT                                                                              |
| FX1    | function retained in ROOT                                                                                |
| G      | temporary array in SPLINE and SPLINT                                                                     |
| H      | temporary array in SPLINE and SPLINT                                                                     |
| I      | iteration index                                                                                          |
| IPLT   | marker indicating which plots have been completed in OUPLOT                                              |
| J      | iteration index                                                                                          |
| K      | iteration index                                                                                          |
| MAXN   | number of spline points or number of interpolated points in SPLINT                                       |

|              |                                                                                                                                     |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------|
| <b>MEXT</b>  | marker in <b>SPLINT</b> indicating that extrapolation has occurred                                                                  |
| <b>MINT</b>  | marker in <b>SPLINT</b> indicating that interpolated point coincides with given point                                               |
| <b>ML</b>    | see Input                                                                                                                           |
| <b>MLM1</b>  | $ML - 1$                                                                                                                            |
| <b>MLP1</b>  | $ML + 1$                                                                                                                            |
| <b>MR</b>    | see Input                                                                                                                           |
| <b>MRM1</b>  | $MR - 1$                                                                                                                            |
| <b>MT</b>    | see Input                                                                                                                           |
| <b>MTM1</b>  | $MT - 1$                                                                                                                            |
| <b>MTP1</b>  | $MT + 1$                                                                                                                            |
| <b>MZ</b>    | see Input                                                                                                                           |
| <b>MZM1</b>  | $MZ - 1$                                                                                                                            |
| <b>N</b>     | number of <b>X</b> and <b>Y</b> spline points in <b>SPLINE</b> and <b>SPLINT</b>                                                    |
| <b>N1</b>    | $N - 1$ in <b>SPLINE</b> and <b>SPLINT</b>                                                                                          |
| <b>NBOT</b>  | see Input                                                                                                                           |
| <b>NINT</b>  | number of interpolated <b>X</b> and <b>Y</b> points in <b>SPLINT</b>                                                                |
| <b>NLE</b>   | see Input                                                                                                                           |
| <b>NTE</b>   | see Input                                                                                                                           |
| <b>NTOP</b>  | see Input                                                                                                                           |
| <b>RBOT</b>  | see Input                                                                                                                           |
| <b>RBPLT</b> | array of <b>R</b> -coordinates of points defining bottom boundary of region plotted in <b>OUPLOT</b>                                |
| <b>RBRNG</b> | <b>R</b> -coordinate of bottom boundary of points plotted in <b>OUPLOT</b>                                                          |
| <b>RCURV</b> | <b>R</b> -coordinate of curved horizontal orthogonal line for a given value of <b>Z</b> in <b>CROSS</b> (fig. 8)                    |
| <b>REFR</b>  | reference <b>R</b> -coordinate of orthogonal mesh point on previously computed horizontal orthogonal in <b>MESH</b> (figs. 5 and 8) |
| <b>REFSL</b> | reference slope of normal to horizontal orthogonal in <b>MESH</b> (fig. 8)                                                          |
| <b>REFZ</b>  | reference <b>Z</b> -coordinate of orthogonal mesh point on previously computed horizontal orthogonal in <b>MESH</b> (figs. 5 and 8) |



|                              |                                                                                                                       |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| SLU                          | slope of horizontal orthogonal curve where it is intersected by a normal from lower orthogonal curve in MESH (fig. 8) |
| SXK,SXK1                     | temporary storage in SPLINT                                                                                           |
| SYM,SYN                      | data variables in OUPLOT containing identification of symbols to be used for plotting                                 |
| TEMP                         | temporary storage array in MESH                                                                                       |
| TITL1, TITL2<br>TITL3, TITL4 | data arrays containing titles to be plotted in OUPLOT                                                                 |
| TOLER                        | tolerance governing calculation of intersection of straight line with curve in MESH and ROOT                          |
| X                            | array of X-coordinates of spline points in SPLINE and SPLINT                                                          |
| XFIND                        | X-coordinate of intersection of straight line with curve in ROOT (fig. 7)                                             |
| XINT                         | array of X-coordinates of interpolation points in SPLINT                                                              |
| XL,XR                        | left and right X-coordinates of bounds on region in which intersection occurs in ROOT                                 |
| XMID                         | X-coordinate of midpoint between XL and XR in ROOT                                                                    |
| Y                            | array of Y-coordinates of spline points in SPLINE and SPLINT                                                          |
| YGIVN                        | given value of function, FUNCT, to be matched by root-finding process in ROOT                                         |
| YINT                         | array of Y-coordinates of interpolation points in SPLINT                                                              |
| Z                            | given value of Z-coordinate at which RCURV and RLINE are calculated in CROSS (fig. 8)                                 |
| ZBEGIN                       | see Input                                                                                                             |
| ZBOT                         | see Input                                                                                                             |
| ZBOTZ                        | temporary variable in CROSS                                                                                           |
| ZBPLT                        | array of Z-coordinates of points defining bottom boundary of region plotted in OUPLOT                                 |
| ZEND                         | see Input                                                                                                             |
| ZFPLT                        | array of Z-coordinates of points defining front, or leading, edge of body plotted in OUPLOT                           |
| ZLE                          | see Input                                                                                                             |

|       |                                                                                                                                                                |
|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ZLRNG | Z-coordinate of left boundary of points plotted in OUPLOT                                                                                                      |
| ZNL   | Z-coordinate of intersection of line normal to previously calculated horizontal orthogonal (at REFZ, REFR) with present horizontal orthogonal in MESH (fig. 5) |
| ZNU   | Z-coordinate of intersection of line through previously calculated horizontal orthogonal (at REFZ, REFR) with and normal to SLU line in MESH (fig. 5)          |
| ZOM   | array of Z-coordinates of mesh points on orthogonal mesh (fig. 9)                                                                                              |
| ZRPLT | array of Z-coordinates of points defining rear, or trailing, edge of body plotted in OUPLOT                                                                    |
| ZRRNG | Z-coordinate of right boundary of points plotted in OUPLOT                                                                                                     |
| ZTE   | see Input                                                                                                                                                      |
| ZTEM  | temporary storage for Z-coordinate arrays                                                                                                                      |
| ZTOP  | see Input                                                                                                                                                      |
| ZTPLT | array of Z-coordinates of points defining top boundary of region plotted in OUPLOT                                                                             |
| ZZBOT | temporary variable in CROSS                                                                                                                                    |

## Program Listing

A complete listing of the FORTRAN program is given here. If this mesh generation program were to be incorporated into a larger program of the reader, the routines MESH, ROOT, CROSS, SPLINE, and SPLINT would be used essentially as given here. The INPUT and OUTPUT routines would probably be incorporated into input and output routines in the reader's program. Finally, the OUPLOT routine would have to be partially recoded to suit the plotting equipment at the reader's facility.

```
$IBFTC MAIN
```

```
10 CALL INPUT
   CALL MESH
   CALL OUTPUT
   CALL OUPLOT
   GO TO 10
   END
```

SIBFTC INPUT

SUBROUTINE INPUT

C  
 C--INPUT READS IN INPUT DATA AND CALCULATES SEVERAL CONSTANTS  
 C--DESCRIBING THE MESH

C  
 COMMON /INPUT/ ML,MT,MZ,MR,ZBEGIN,ZEND,  
 INBOT,NTOP,ZBOT(50),RBOT(50),ZTOP(50),RTOP(50),  
 2NLE,NTE,7LE(50),RLE(50),ZTE(50),RTE(50)  
 COMMON /CALC/ MLM1,MLP1,MTM1,MTP1,MZM1,MRM1,  
 17OM(50,50),ROM(50,50),ADM(50,50)

C  
 C--READ INPUT DATA  
 READ (5,1050)  
 WRITE(6,1000)  
 WRITE(6,1050)  
 WRITE(6,1100)  
 READ (5,1010) ML,MT,MZ,MR,NBOT,NTOP,NLE,NTE  
 WRITE(6,1020) ML,MT,MZ,MR,NBOT,NTOP,NLE,NTE  
 WRITE(6,1110)  
 READ (5,1030) ZBEGIN,ZEND  
 WRITE(6,1040) ZBEGIN,ZEND  
 WRITE(6,1120)  
 READ (5,1030) (ZBOT(I),I=1,NBOT)  
 WRITE(6,1040) (ZBOT(I),I=1,NBOT)  
 WRITE(6,1130)  
 READ (5,1030) (RBOT(I),I=1,NBOT)  
 WRITE(6,1040) (RBOT(I),I=1,NBOT)  
 WRITE(6,1140)  
 READ (5,1030) (ZTOP(I),I=1,NTOP)  
 WRITE(6,1040) (ZTOP(I),I=1,NTOP)  
 WRITE(6,1150)  
 READ (5,1030) (RTOP(I),I=1,NTOP)  
 WRITE(6,1040) (RTOP(I),I=1,NTOP)  
 WRITE(6,1160)  
 READ (5,1030) (ZLE(I),I=1,NLE)  
 WRITE(6,1040) (ZLE(I),I=1,NLE)  
 WRITE(6,1170)  
 READ (5,1030) (RLE(I),I=1,NLE)  
 WRITE(6,1040) (RLE(I),I=1,NLE)  
 WRITE(6,1180)  
 READ (5,1030) (ZTE(I),I=1,NTE)  
 WRITE(6,1040) (ZTE(I),I=1,NTE)  
 WRITE(6,1190)  
 READ (5,1030) (RTE(I),I=1,NTE)  
 WRITE(6,1040) (RTE(I),I=1,NTE)

C  
 C--CALCULATE MISCELLANEOUS CONSTANTS  
 MLM1 = ML-1  
 MLP1 = ML+1  
 MTM1 = MT-1  
 MTP1 = MT+1  
 MZM1 = MZ-1  
 MRM1 = MR-1  
 RETURN

```

C
C--FORMAT STATEMENTS
1000 FORMAT (1H1)
1010 FORMAT (16I5)
1020 FORMAT (4(1X,I6),5X,2(1X,I6),5X,2(1X,I6))
1030 FORMAT (8F10.5)
1040 FORMAT (8(1X,G15.6))
1050 FORMAT (80H
      1
      )
1100 FORMAT (//1X,38FMESH INPUT DATA AND INPUT ARRAY BOUNDS/5X,2HML,5X,
      12HMT,5X,2HMZ,5X,2HMR,9X,4HNBOT,3X,4HNTP,9X,3HNLE,4X,3HNTE)
1110 FORMAT (5X,6HZBEGIN,11X,4HZEND)
1120 FORMAT (//1X,19HBOUNDARY INPUT DATA/5X,1CHRBTOT ARRAY)
1130 FORMAT (5X,1CHRBTOT ARRAY)
1140 FORMAT (5X,1CHZTOP ARRAY)
1150 FORMAT (5X,1CHRTOP ARRAY)
1160 FORMAT (//1X,15HBODY INPUT DATA/5X,9HZLE ARRAY)
1170 FORMAT (5X,9HRLE ARRAY)
1180 FORMAT (5X,9HZTE ARRAY)
1190 FORMAT (5X,9HRTE ARRAY)
      FND

```

\*IBFTC MESH

SUBROUTINE MESH

```

C
C--MESH GENERATES AND CALCULATES THE COORDINATES OF AN ORTHOGONAL
C--MESH BETWEEN THE INPUT BOUNDARIES

```

```

C
      COMMON /INPUT/ ML,MT,MZ,MR,ZBEGIN,ZEND,
      INBT,NTOP,ZBOT(50),RBT(50),ZTOP(50),RTOP(50),
      2NLF,NTE,ZLE(50),RLE(50),ZTE(50),RTE(50)
      COMMON /CALC/ MLM1,MLP1,MTM1,MTP1,MZM1,MRM1,
      1ZOM(50,50),ROM(50,50),ADM(50,50)
      COMMON /CROSS/ RTEM(50),SUTEM(50),REFZ,REFR,REFSL
      DIMENSION RRAD(50,50),SLOM(50),TEMP(50)
      EXTERNAL CROSS

```

```

C
C--STORE RRAD ON BOTTOM BOUNDARY

```

```

      DO 10 I=1,NBOT
      10 RRAD(I,1) = RBT(I)

```

```

C
C--CALCULATE RRAD ON TOP BOUNDARY

```

```

      CALL SPLINT(ZTOP,RTOP,NTOP,ZBOT,NBOT,RRAD(1,MR),TEMP)

```

```

C
C--CALCULATE RRAD ON RADIAL LINES FROM BOTTOM TO TOP BOUNDARY

```

```

      DO 20 I=1,NBOT
      DELR = (RRAD(I,MR)-RRAD(I,1))/FLOAT(MRM1)
      DO 20 J=2,MRM1
      20 RRAD(I,J) = RRAD(I,J-1)+DELR

```

```

C
C--CALCULATE ZOM ON BOTTOM BOUNDARY

```

```

      DELZ = (ZLF(1)-ZBEGIN)/FLOAT(MLM1)
      ZOM(1,1) = ZBEGIN
      DO 30 I=2,ML

```

```

30 ZOM(I,1) = ZOM(I-1,1)+DELZ
   DELZ = (ZTE(1)-ZLE(1))/FLOAT(MT-ML)
   DO 40 I=MLP1,MT
40 ZOM(I,1) = ZOM(I-1,1)+DELZ
   DELZ = (ZEND-ZTE(1))/FLOAT(MZ-MT)
   DO 50 I=MTP1,MZ
50 ZOM(I,1) = ZOM(I-1,1)+DELZ
C
C--CALCULATE ENTRIES IN ZOM AND ROM ROW BY ROW
C
   DO 150 J=2,MR
C
C--CALCULATE R-COORDINATES, SLOPES, AND ANGLES OF PREVIOUS ROW
   CALL SPLINT(ZBOT,RRAD(1,J-1),NBOT,ZOM(1,J-1),MZ,ROM(1,J-1),SLOM)
   DO 60 I=1,MZ
60 AGM(I,J-1) = ATAN(SLOM(I))*57.295780
C
C--CALCULATE RTEM, AND SECOND DERIVATIVES, SDTEM, CF RRAD
C--ALONG PRESENT ROW
   DO 70 I=1,NBOT
70 RTEM(I) = RRAD(I,J)
   CALL SPLINE(ZBOT,RTEM,NBOT,TEMP,SDTEM)
C
C--MOVE ALONG PRESENT ROW ONE POINT AT A TIME, LOCATING ZOM COORDINATES
C--OF ORTHOGONAL MESH POINTS ALONG THE ROW
   DO 140 I=1,M7
   REFZ = ZOM(I,J-1)
   REFR = ROM(I,J-1)
   DO 80 K=1,NBOT
   IF (ZBOT(K).LT.ZOM(I,J-1)) GO TO 80
   DELR = RRAD(K-1,J)-RRAD(K-1,J-1)-(ZOM(I,J-1)-ZBOT(K-1))/
1(ZBOT(K)-ZBOT(K-1))*(RRAD(K-1,J)-RRAD(K-1,J-1)-RRAD(K,J)+
2RRAD(K,J-1))
   GO TO 90
80 CONTINUE
90 IF (ABS(SLOM(I)).LE.0.0001) GO TO 120
C
C--LOCATE INTERSECTION OF LINE NORMAL TO PREVIOUS ROW WITH PRESENT ROW
   REFFSL = -1./SLOM(I)
   DEL7 = DELR/REFFSL
   IF (REFFSL.LT.0.) GO TO 100
   BNDL = ZOM(I,J-1)
   BNDR = BNDL+2.0*DELZ
   GO TO 110
100 BNDR = ZOM(I,J-1)
   BNDL = BNDR+2.0*DELZ
110 TOLER = DELR/200.
   CALL ROOT(BNDL,BNDR,0.,CROSS,TOLER,ZNL,SLU)
   RNL = REFR+REFFSL*(ZNL-REFZ)
   GO TO 130
C
C--LOCATE INTERSECTION OF LINE NORMAL TO PREVIOUS ROW WITH
C--PRESENT ROW IF NORMAL LINE IS RADIAL
120 REFFSL = 0.
   ZNL = ZOM(I,J-1)
   CALL CROSS(ZOM(I,J-1),RNL,SLU)
C
C--CALCULATE INTERSECTION OF LINE NORMAL TO PRESENT ROW

```

```

C--FROM POINT ON PREVIOUS ROW
  130 ZNU = (ZUM(I,J-1)+(ROM(I,J-1)-RNL)*SLU+ZNL*SLU**2)/(1.+SLJ**2)
      RNU = RNL+SLU*(ZNU-ZNL)
C
C--CALCULATE FINAL ESTIMATE FOR NEW ZOM POINT BY AVERAGING
  140 ZOM(I,J) = (ZNL+ZNU)/2.
  150 CONTINUE
C
C--CALCULATE R-COORDINATES, SLOPES, AND ANGLES OF FINAL ROW
  CALL SPLINT(ZBOT,RRAD(1,MR),NBOT,ZUM(1,MR),MZ,ROM(1,MR),SLOM)
  DO 160 I=1,MZ
  160 ADM(I,MR) = ATAN(SLOM(I))*57.295780
      RETURN
      FND

```

5IBFTC OUTPUT

SUBROUTINE OUTPUT

```

C
C--OUTPUT PRINTS THE COORDINATES OF THE ORTHOGONAL MESH
C
  COMMON /INPU/ ML,MT,MZ,MR,ZBEGIN,ZEND,
  INBOT,NTOP,ZBOT(50),RBOT(50),ZTUP(50),RTOP(50),
  ZNLE,NTF,ZLE(50),RLE(50),ZTE(50),RTE(50)
  COMMON /CALC/ MLM1,MLP1,MTM1,MTP1,MZM1,MRM1,
  170M(50,50),RCM(50,50),ADM(50,50)
C
C--WRITE ZOM
  WRITE(6,1000)
  WRITE(6,1010)
  DO 10 J=1,MR
  IF (J.EQ.1) WRITE(6,1040) J
  IF (J.EQ.MR) WRITE(6,1050) J
  IF (J.NE.1.AND.J.NE.MR) WRITE(6,1060) J
  10 WRITE(6,1070) (ZOM(I,J),I=1,MZ)
C
C--WRITE ROM
  WRITE(6,1000)
  WRITE(6,1020)
  DO 20 J=1,MR
  IF (J.EQ.1) WRITE(6,1040) J
  IF (J.EQ.MR) WRITE(6,1050) J
  IF (J.NE.1.AND.J.NE.MR) WRITE(6,1060) J
  20 WRITE(6,1070) (ROM(I,J),I=1,MZ)
C
C--WRITE ADM
  WRITE(6,1000)
  WRITE(6,1030)
  DO 30 J=1,MR
  IF (J.EQ.1) WRITE(6,1040) J
  IF (J.EQ.MR) WRITE(6,1050) J
  IF (J.NE.1.AND.J.NE.MR) WRITE(6,1060) J
  30 WRITE(6,1080) (ADM(I,J),I=1,MZ)
      RETURN

```

```

C
C--FORMAT STATEMENTS
1000 FORMAT (1H1)
1010 FORMAT (/52X,27H***** /52X,3H***,21X,3H***/
1 52X,27H*** 7 COORDINATES OF ***/52X,27H***  ORTHOGONAL MESH
2 ***/52X,3H***,21X,3H***/52X,27H***** )
1020 FORMAT (/52X,27H***** /52X,3H***,21X,3H***/
1 52X,27H*** R COORDINATES OF ***/52X,27H***  ORTHOGONAL MESH
2 ***/52X,3H***,21X,3H***/52X,27H***** )
1030 FORMAT (/50X,32H***** /50X,3H***,26X,
13H***/ 50X,32H***  ANGLES WITH RESPECT ***/50X,32H***  TO THE
2 7 DIRECTION ***/50X,32H***  OF THE ORTHOGONAL MESH ***/50X,3
3H***,26X,3H***/50X,32H***** )
1040 FORMAT (/1X,17H** BOTTOM ROW NO.,I3,3H **)
1050 FORMAT (/1X,14H** TOP ROW NO.,I3,3H **)
1060 FORMAT (/1X,10H** ROW NO.,I3,3H **)
1070 FORMAT (1X,8G16.6)
1080 FORMAT (1X,8F16.3)
      END

```

\$TBFC DUPL0T

SUBROUTINE DUPL0T

```

C
C--DUPL0T PLOTS THE BOUNDARIES OF THE SOLUTION REGION,
C--AS WELL AS THE GENERATED ORTHOGONAL MESH
C
      COMMON /INPU/ ML,MT,MZ,MR,ZBEGIN,ZEND,
      INBOT,NTOP,ZBOT(50),RBOT(50),ZTOP(50),RTOP(50),
      2NLF,NTF,ZLE(50),RLE(50),ZTE(50),RTE(50)
      COMMON /CALC/ MLM1,MLP1,MTM1,MTP1,MZM1,MRM1,
      17DM(50,50),ROM(50,50),AOM(50,50)
      DIMENSION ZBPLT(100),RBPLT(100),ZTPLT(100),RTPLT(100),
      17FPLT(100),RFPLT(100),ZRPLT(100),RRPLT(100),ZTEM(100),RTEM(100),
      2SDTEM(100),TITL1(5),TITL2(3),TITL3(2),TITL4(2)
      DATA TITL1/30H BOUNDARIES OF PLOT REGION /
      DATA TITL2/18H ORTHOGONAL MESH /
      DATA TITL3/12HZ DIRECTION/
      DATA TITL4/12HR DIRECTION/
      DATA SYM/1HX/
      DATA SYN/1HO/

```

```

C
C--OBTAIN PLOT POINTS ON LOWER BOUNDARY
      DEL7 = (7BOT(NBOT)-ZBOT(1))/99.
      ZBPLT(1) = ZBOT(1)
      ZBPLT(100) = 7BOT(NBOT)
      DO 10 I=2,99
10 ZBPLT(I) = ZBPLT(I-1)+DELZ
      CALL SPLINT(ZBOT,RBOT,NBOT,ZBPLT,100,RBPLT,SDTEM)

```

```

C
C--OBTAIN PLOT POINTS ON UPPER BOUNDARY
      DELZ = (7TOP(NTOP)-ZTOP(1))/99.
      ZTPLT(1) = ZTOP(1)
      ZTPLT(100) = ZTOP(NTOP)
      DO 20 I=2,99

```

```

20 ZTPLT(I) = ZTPLT(I-1)+DELZ
   CALL SPLINT(ZTOP,RTOP,NTOP,ZTPLT,100,RTPLT,SDTEM)
C
C--OBTAIN PLOT POINTS UP LEADING EDGE OF BODY
DELK = (RLE(NLE)-RLE(1))/99.
RFPLT(1) = RLE(1)
RFPLT(100) = RLE(NLE)
DO 30 J=2,99
30 RFPLT(J) = RFPLT(J-1)+DELK
   CALL SPLINT(RLE,ZLE,NLE,RFPLT,100,ZFPLT,SDTEM)
C
C--OBTAIN PLOT POINTS UP TRAILING EDGE OF BODY
DELK = (RTE(NTE)-RTE(1))/99.
RRPLT(1) = RTE(1)
RRPLT(100) = RTE(NTE)
DO 40 J=2,99
40 RRPLT(J) = RRPLT(J-1)+DELK
   CALL SPLINT(RTE,ZTE,NTE,RRPLT,100,ZRPLT,SDTEM)
C
C--CALCULATE THE RANGE OF THE PLOT
ZLRNG = AMIN1(ZBOT(1),ZTOP(1))
ZRRNG = AMAX1(ZBOT(NBOT),ZTOP(NTOP))
DELZ = ZRRNG-ZLRNG
ZLRNG = ZLRNG-0.05*DELZ
ZRRNG = ZRRNG+0.05*DELZ
RBRNG = RBOT(1)
DO 50 I=2,NBOT
50 RBRNG = AMIN1(RBRNG,RBOT(I))
KTRNG = RTOP(1)
DO 60 I=2,NTOP
60 KTRNG = AMAX1(KTRNG,RTOP(I))
DELK = KTRNG-RBRNG
KBRNG = RBRNG-0.05*DELK
KTRNG = KTRNG+0.05*DELK
C
C--CHOOSE MAXIMUM RANGE, AND EXPAND RANGE IN OTHER DIRECTION
DMD2 = 1.1*ABS(DELZ-DELK)/2.
IF (DELK.GT.DELZ) GO TO 70
KTRNG = KTRNG+DMD2
KBRNG = KBRNG-DMD2
GO TO 80
70 ZKRNG = ZRRNG+DMD2
ZLRNG = ZLRNG-DMD2
C
C--PLOT BOUNDARIES OF REGION AND PLOT ORTHOGONAL MESH
80 CALL ICMRGN(1.0,1.0,1.0,1.0)
   CALL LRANGE(ZLRNG,ZRRNG,RBRNG,KTRNG)
   CALL IGRID(-1,-1,1.0,1.0)
   IPI = 1
   CALL LRLEGN(TITL1,28,0.3,0.5,0.0)
90 IF (IPI.EQ.2) CALL LRLEGN(TITL2,16,0.3,0.5,0.0)
   CALL ICHS7(2)
   CALL LRLEGN(TITL3,12,0.7,0.8,0.0)
   CALL LRLEGN(TITL4,12,1.0,0.8,0.0)
   CALL ICHS7(3)
   CALL IRCURV(ZBPLT,RBPLT,100,2,SYM,C,C)
   CALL IRCURV(ZTPLT,RTPLT,100,2,SYM,C,C)
   CALL IRCURV(ZFPLT,RFPLT,100,2,SYM,C,C)
   CALL IRCURV(ZRPLT,RRPLT,100,2,SYM,C,C)

```

```

IF (IPLT.EQ.2) GO TO 100
CALL LRCURV(ZBOT, RBOT, NBOT, 4, SYM, 0.0)
CALL LRCURV(ZTOP, RTOP, NTOP, 4, SYM, C.0)
CALL LRCURV(ZLE, RLE, NLE, 3, SYM, C.0)
CALL LRCURV(ZTE, RTE, NTE, 3, SYM, 1.0)
IPLT = 2
GO TO 90

```

```

C
C--PLOT VERTICAL MESH LINES
100 DO 120 I=1,M7
    ZTEM(1) = ZOM(I,1)
    RTEM(1) = ROM(I,1)
    DO 110 J=2,MR
        ZTEM(J) = ZOM(I,J)
    110 RTEM(J) = ROM(I,J)
120 CALL LRCURV(ZTEM, RTEM, MR, 2, SYM, C.0)

```

```

C
C--PLOT HORIZONTAL MESH LINES
EOP = 0.0
DO 130 J=2,MRM1
    IF (J.EQ.MRM1) EOP=1.0
130 CALL LRCURV(ZOM(1,J), ROM(1,J), MZ, 2, SYM, EOP)
RETURN
END

```

5IBFTC ROOT

```

SUBROUTINE ROOT(BNDL, BNDR, YGIVN, FUNCT, TOLER, XFIND, DFX)

```

```

C
C--ROOT FINDS A ROOT FOR (FUNCT-YGIVN) IN THE INTERVAL (BNDL, BNDR)
C

```

```

    XL = BNDL
    XR = BNDR
    CALL FUNCT(BNDL, FX1, DFX)
    DO 20 I=1,20
        XMID = (XL+XR)/2.
        CALL FUNCT(XMID, FX, DFX)
        IF ((FX1-YGIVN)*(FX-YGIVN).GT.C.) GO TO 10
        XR = XMID
        GO TO 20
    10 XL = XMID
        FX1 = FX
    20 CONTINUE
        XFIND = XMID
        IF (ABS(YGIVN-FX).LT.TOLER) RETURN
        *WRITE(6,1000) BNDL, BNDR, YGIVN
        STOP

```

```

1000 FORMAT(///4X,49HROOT HAS FAILED TO CONVERGE IN THE GIVEN INTERVAL
1/4X,6HBNDL =,G14.6,10X,6HBNDR =,G14.6,10X,7HYGIVN =,G14.6)
END

```

§IBFTC CROSS

```

SUBROUTINE CROSS(Z,RMR,SL)
C
C--CROSS CALCULATES R AS A FUNCTION OF Z ALONG A CURVE AND A
C--STRAIGHT LINE WHICH INTERSECTS IT, AND THEN FOR A GIVEN VALUE OF Z
C--CALCULATES THE DIFFERENCE BETWEEN THE VALUES OF R ON THE
C--STRAIGHT LINE AND THE CURVE
C
COMMON /INPU/ ML,MT,MZ,MR,ZBEGIN,ZEND,
INBOT,NTOP,ZBOT(50),RBOT(50),ZTOP(50),RTOP(50),
ZNLE,NTE,ZLE(50),RLE(50),ZTE(50),RTE(50)
COMMON /CROS/ RTEM(50),SDTEM(50),REFZ,KEFR,REFSL
C
C--LOCATE POSITION OF Z IN RRAD (OR RTEM) ARRAY
DO 10 I=2,NBOT
IF (Z.LE.ZBOT(I)) GO TO 20
10 CONTINUE
C
C--COMPUTE R-COORDINATE (RCURV) AND SLOPE (SL) ON THE CURVE
C--FOR THE GIVEN VALUE OF Z
20 DELZ = ZBOT(I)-ZBOT(I-1)
ZBOTZ = ZBOT(I)-Z
ZZBOT = Z-ZBOT(I-1)
SDI = SDTEM(I)
SDI1 = SDTEM(I-1)
RTI = RTEM(I)/DELZ
RTI1 = RTEM(I-1)/DELZ
RCURV = SDI1*ZBOTZ**3/6./DELZ + SDI*ZZBOT**3/6./DELZ +
1(RTI-SDI*DELZ/6.)*ZZBOT + (RTI1-SDI1*DELZ/6.)*ZBOTZ
SL = -SDI1*ZBOTZ**2/2./DELZ + SDI*ZZBOT**2/2./DELZ +
1RTI-RTI1 - (SDI-SDI1)*DELZ/6.
IF (REFSL.EQ.0.) GO TO 30
C
C--COMPUTE R-COORDINATE (RLINE) ON STRAIGHT LINE
C--FOR GIVEN VALUE OF Z
RLINE = KEFR+REFSL*(Z-REFZ)
C
C--COMPUTE DIFFERENCE IN R COORDINATES
RMR = RCURV-RLINE
RETURN
C
C--SPECIAL CASE FOR RADIAL STRAIGHT LINE
30 RMR = RCURV
RETURN
END

```

§IBFTC SPLINE

```

SUBROUTINE SPLINE(X,Y,N,DYDX,D2YDX2)
C
C--SPLINE FITS A SPLINE CURVE TO X AND Y
C--AND CALCULATES FIRST AND SECOND DERIVATIVES AT THE SPLINE POINTS
C--END POINT SECOND DERIVATIVES ONE HALF THOSE AT ADJACENT POINTS

```

C

```

DIMENSION X(N),Y(N),DYDX(N),D2YDX2(N)
DIMENSION G(100),H(100)
G(1) = -0.5
H(1) = 0.
N1 = N-1
IF (N1.LT.2) G(1)=1.0
IF (N1.LT.2) GO TO 20
DO 10 I=2,N1
A = (X(I)-X(I-1))/6.
B = (X(I+1)-X(I))/6.
C = 2.*(A+B)-A*G(I-1)
D = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
G(I) = B/C
10 H(I) = (D-A*H(I-1))/C
20 D2YDX2(N) = H(N1)/(2.0+G(N1))
DO 30 I=2,N
K = N+1-I
30 D2YDX2(K) = H(K)-G(K)*D2YDX2(K+1)
DYDX(1) = (X(1)-X(2))/6.*(2.*D2YDX2(1)+D2YDX2(2))+(Y(2)-Y(1))/(X(2)
1-X(1))
DO 40 I=2,N
40 DYDX(I) = (X(I)-X(I-1))/6.*(2.*D2YDX2(I)+D2YDX2(I-1))+(Y(I)-Y(I-1))
1/(X(I)-X(I-1))
RETURN
END

```

### SIBFTC SPLINT

SUBROUTINE SPLINT(X,Y,N,XINT,NINT,YINT,DYDXNT)

C

C--SPLINT FITS A SPLINE CURVE TO X AND Y, AND THEN CALCULATES  
C--INTERPOLATED Y VALUES AND DERIVATIVES AT GIVEN X POINTS  
C--END POINT SECOND DERIVATIVES ONE HALF THOSE AT ADJACENT POINTS

C

```

DIMENSION X(N),Y(N),XINT(NINT),YINT(NINT),DYDXNT(NINT)
DIMENSION G(100),H(100),D2YDX2(100)
IF (NINT.LE.0) RETURN

```

C

C COMPUTE SPLINE CURVE

```

G(1) = -0.5
H(1) = 0.
N1 = N-1
IF (N1.LT.2) G(1)=1.0
IF (N1.LT.2) GO TO 20
DO 10 I=2,N1
A = (X(I)-X(I-1))/6.
B = (X(I+1)-X(I))/6.
C = 2.*(A+B)-A*G(I-1)
D = (Y(I+1)-Y(I))/(X(I+1)-X(I))-(Y(I)-Y(I-1))/(X(I)-X(I-1))
G(I) = B/C
10 H(I) = (D-A*H(I-1))/C
20 D2YDX2(N) = H(N1)/(2.0+G(N1))
DO 30 I=2,N
K = N+1-I

```

```

30 D2YDX2(K)= H(K)-G(K)*D2YDX2(K+1)
C
C INTERPOLATE ON SPLINE CURVE TO GET YINT
  MINT = 0
  MFXT = 0
  DO 120 I=1,NINT
  K= 2
  IF (XINT(I)-X(1)) 50,40,60
40 YINT(I)= Y(1)
  MINT = 1
  GO TO 100
50 MFXT = 2
  GO TO 100
60 IF (XINT(I)-X(K)) 100,70,80
70 YINT(I)= Y(K)
  MINT = 1
  GO TO 100
80 K= K+1
  IF (K-N) 60,60,90
90 K= N
  MFXT = 2
100 SLK= X(K)-X(K-1)
  SXX= X(K)-XINT(I)
  SXX1= XINT(I)-X(K-1)
  IF (MINT.EQ.1) GO TO 110
  YINT(I)= D2YDX2(K-1)*SXX**3/6./SLK+D2YDX2(K)*SXX1**3/6./SLK
  1+(Y(K)/SLK-D2YDX2(K)*SLK/6.)*SXX1+(Y(K-1)/SLK-D2YDX2(K-1)*SLK/6.)
  2*SXX
110 MINT = 0
  DYDXNT(I)= -D2YDX2(K-1)*SXX**2/2./SLK+D2YDX2(K)*SXX1**2/2./SLK
  1+(Y(K)-Y(K-1))/SLK-(D2YDX2(K)-D2YDX2(K-1))*SLK/6.
  IF (MFXT.EQ.2) WRITE(6,1000) XINT(I),YINT(I)
  IF (MFXT.EQ.2) MEXT=1
120 CONTINUE
  MAXN = MAX0(N,NINT)
  IF (MFXT.GT.0) WRITE(6,1010) N,NINT,(X(I),Y(I),XINT(I),YINT(I),
  1DYDXNT(I),I=1,MAXN)
  RETURN
1000 FORMAT(56H SPLINE USED FOR EXTRAPOLATION.  EXTRAPOLATED X AND Y =
  1,2G14,6)
1010 FORMAT(2X,24HNUMBER OF POINTS GIVEN =,I3,4X,31HNUMBER OF POINTS IN
  1TERPOLATED =,I3/1)X,1HX,19X,14Y,14X,14HX INTERPOLATED,6X,14HY INTE
  2RPOLATED,5X,17HDYDX INTERPOLATED/(5E20,8))
  END

```

## CONCLUDING REMARKS

A FORTRAN IV program has been presented which computes and plots coordinates for a two-dimensional orthogonal mesh between the walls of a channel. All routines in the program are in easily transferable FORTRAN code. A plot routine, however, uses in-house subroutines and equipment and would have to be recoded elsewhere.

This report has presented a description of the method used, a description of the program itself, and a complete listing.

Lewis Research Center,

National Aeronautics and Space Administration,

Cleveland, Ohio, January 28, 1972,

764-74.

## APPENDIX - LEWIS LIBRARY SUBROUTINES

The following descriptions of the NASA Library subroutines - LRMRGN, LRANGE, LRGRID, LRLEGN, LRCHSZ, and LRCURV - are copied from the NASA Lewis Programmers Manual. These routines are part of a microfilm plotting package called CINEMATIC.

### Subroutine LRMRGN

Purpose. - LRMRGN is used to change the width of plot margins.

Usage. - CALL LRMRGN (XLEFT, XRIGHT, YBOTM, YTOP) where XLEFT (floating point) is the left margin width in absolute positioning units, XRIGHT (floating point) is the right margin width in absolute positioning units, YBOTM (floating point) is the lower margin width in absolute positioning units, and YTOP (floating point) is the upper margin width in absolute positioning units.

Method. - CINEMATIC sets margins around the plotting area as follows: LEFT and BOTTOM, 1.0 absolute positioning units; RIGHT and TOP, 0.4 absolute positioning units. A call to LRMRGN before LRCURV will change the width of the margins. A frame of film contains 10 absolute positioning units in the horizontal and in the vertical directions. Thus, a margin of 1.0 absolute positioning units is 1/10 of a frame wide.

Restrictions. - LRMRGN must be called before the plotting routine it applies to. The settings of LRMRGN remain in effect until changed by another call to LRMRGN.

### Subroutine LRANGE

Purpose. - LRANGE is used to set the range of (X, Y) curve points.

Usage. - CALL LRANGE (XLEFT, XRIGHT, YBOTM, YTOP), where XLEFT is the left end point of a plot in the user's units, XRIGHT is the right end point of a plot in the user's units, YBOTM is the lower end point of a plot in the user's units, and YTOP is the upper end point of a plot in the user's units.

Method. - The curve-plotting subroutine LRCURV searches the (X, Y) coordinates for maximums and minimums and scales the rest of the user's points to fit between them. A call to LRANGE before LRCURV suppresses the search. The settings of LRANGE remain in effect for all successive plots until changed by another call to LRANGE. XLEFT = XRIGHT = 0.0 can be used to remove the LRANGE X-settings without providing new ones. YBOTM = YTOP = 0.0 does the same for Y-settings.

Restrictions. - LRANGE must be called before the curve plotting routine it applies to. The settings of LRANGE remain in effect until changed by the user.

## Subroutine LRGRID

Purpose. - LRGRID is used to specify grid line changes.

Usage. - CALL LRGRID (IXCODE, IYCODE, DX, DY). IXCODE (fixed point) is a switch used as follows: It applies to vertical grid lines.

IXCODE = 0 means return to using CINEMATIC's built-in grid format (11 grid lines).

IXCODE =  $\pm 1$  means DX specifies how many grid lines; IXCODE = -1 suppresses grid labels.

IXCODE =  $\pm 2$  means DX specifies grid intervals; IXCODE = -2 suppresses grid labels.

IXCODE =  $\pm 3$  means DX specifies how many "tick marks" instead of grid lines; IXCODE = -3 suppresses grid labels.

IXCODE =  $\pm 4$  means DX specifies interval between "tick marks"; IXCODE = -4 suppresses grid labels.

DX (floating point) specifies grid line or "tick mark" frequency or intervals, depending on how IXCODE is set. IYCODE (fixed point) is the same as IXCODE, but it applies to horizontal grid lines. DY (floating point) is the same as DX but for horizontal grid lines.

Method. - CINEMATIC puts 11 horizontal and 11 vertical grid lines on every plot, unless LRGRID is called. When a grid line frequency is specified, CINEMATIC sets the interval between the specified number of grid lines to be equal to  $Z \times 10^n$ , where  $Z = 1.0, 2.0, 2.5, \text{ or } 5.0$  and  $n$  depends on the magnitude of the user's data. To get these intervals, CINEMATIC will adjust the end points of the plot, if necessary. This adjustment occurs only when a grid line frequency is specified. To avoid any adjustments, specify grid intervals.

Restriction. - LRGRID must be called before the plotting routine it applies to. The settings of LRGRID remain in effect until changed by another call to LRGRID.

## Subroutine LRLEGN

Purpose. - LRLEGN is used to print a legend anywhere on a plot.

Usage. - CALL LRLEGN (CHARS, N, IORIEN, X, Y, EOP). CHARS is an array of characters to be printed. CHARS must be dimensioned large enough for the number of characters desired (four characters per word on the 360; six characters per word

on the 7094/7044 direct couple). N (fixed point) is the number of characters to be printed. IORIEN (fixed point) is a switch:

IORIEN = 0 causes horizontal printing.

IORIEN = 1 causes vertical printing.

X (floating point) is the X-coordinate of the starting point in absolute positioning units.

Y (floating point) is the Y-coordinate of the starting point in absolute positioning units.

EOP (floating point) is a switch:

EOP = 0 indicates the current plot is not yet complete.

EOP = 1 indicates the current plot is complete. No more calls to plotting or printing subroutines for this plot will occur.

Method. - The user expresses the (X, Y) starting point of a line of printing in absolute positioning units. Absolute positioning units range from 0.0 to 10.0 in both the X- and Y-directions for one frame of film. Absolute positioning units give the user a coordinate system to specify (X, Y) points independent of points on a curve.

LRLEGN prints medium-size characters. The user may also get other character sizes, italics, lower case, and special symbols.

## Subroutine LRCHSZ

Purpose. - LRCHSZ is used to change the size of printed characters.

Usage. - CALL LRCHSZ (ISIZE) where ISIZE (fixed point) gives the size:

ISIZE = 0 means let CINEMATIC resume selecting the size.

ISIZE = 1 means miniature characters.

ISIZE = 2 means small characters.

ISIZE = 3 means medium characters.

ISIZE = 4 means large characters.

Method. - LRCHSZ changes the character size for all character printing that follows. The specified size remains in effect until changed by another call to LRCHSZ.

Large: 43 characters per line, 22 lines per frame.

Medium: 64 characters per line, 32 lines per frame.

Small: 86 characters per line, 43 lines per frame.

Miniature: 128 characters per line, 64 lines per frame.

Restriction. - LRCHSZ must be called before the printing subroutine it applies to.

## Subroutine LRCURV

Purpose. - LRCURV is used to plot one curve of a multiple-curve plot.

Usage. - CALL LRCURV (X, Y, N, ITYPE, SYMBOL, EOP). X (floating point) is an array of X-coordinates for the curve. Y (floating point) is an array of Y-coordinates for the curve. N (fixed point) is the number of (X, Y) points to be plotted. ITYPE is a switch that indicates the type of plot desired:

ITYPE = 1 specifies a dot plot. Each (X, Y) point is represented by a dot.

ITYPE = 2 specifies a vector plot. Successive (X, Y) points are joined by straight lines.

ITYPE = 3 specifies a symbol plot. Each (X, Y) point is represented by a symbol. The FORTRAN character in SYMBOL specifies the symbol used.

ITYPE = 4 specifies a special symbol plot. Each (X, Y) point is represented by a special symbol taken from the SPECIAL CHARACTER TABLE. The special symbol used is the one corresponding to the FORTRAN character in SYMBOL.

ITYPE = 5 is the same as ITYPE = 3, except that a smaller size symbol is used.

ITYPE = 6 is the same as ITYPE = 4, except that a smaller size symbol is used.

SYMBOL specifies the plotting symbol when ITYPE = 3 or 4. When ITYPE = 1 or 2, SYMBOL must appear in the call list but is not used by LRCURV. EOP is a switch that indicates when the last subroutine call for a given plot is being made:

EOP = 0.0 means the current plot is not yet complete. More subroutine calls for this plot will follow.

EOP = 1.0 means the current plot is complete. No more printing or plotting subroutines will be called for this plot.

Method. - LRCURV provides greater flexibility in drawing curves. LRCURV is useful for the plotting situation in which all (X, Y) points for a plot are not in the computer memory at the same time. Several calls to LRCURV may be made for the same plot.

The X and Y arrays are in whatever units the user is working with. LRCURV scales his data range to fit the size of the plot on film. The user should call LRANGE before LRCURV to supply the range of his data points to CINEMATIC. If the user does not call LRANGE, LRCURV will take the user's data range from the first call to LRCURV for any given plot.

LRCURV does not destroy the contents of X, Y, N, ITYPE, SYMBOL, or EOP during plotting.

## REFERENCES

1. McCracken, Daniel D.; and Dorn, William S.: Numerical Methods and FORTRAN Programming. John Wiley & Sons, Inc., 1964.
2. Walsh, J. L.; Ahlberg, J. H.; and Nilson, E. N.: Best Approximation Properties of the Spline Fit. J. Math. Mech., vol. 11, no. 2, 1962, pp. 225-234.



015 001 C1 U 01 720324 S00903DS  
DEPT OF THE AIR FORCE  
AF WEAPONS LAB (AFSC)  
TECH LIBRARY/WLOL/  
ATTN: E LOU BOWMAN, CHIEF  
KIRTLAND AFB NM 87117

POSTMASTER: If Undeliverable (Section 158  
Postal Manual) Do Not Return

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

## NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

**TECHNICAL REPORTS:** Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

**TECHNICAL NOTES:** Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

**TECHNICAL MEMORANDUMS:** Information receiving limited distribution because of preliminary data, security classification, or other reasons.

**CONTRACTOR REPORTS:** Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

**TECHNICAL TRANSLATIONS:** Information published in a foreign language considered to merit NASA distribution in English.

**SPECIAL PUBLICATIONS:** Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

**TECHNOLOGY UTILIZATION PUBLICATIONS:** Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Technology Surveys.

*Details on the availability of these publications may be obtained from:*

**SCIENTIFIC AND TECHNICAL INFORMATION OFFICE**

**NATIONAL AERONAUTICS AND SPACE ADMINISTRATION**

**Washington, D.C. 20546**