COMMISSION OF THE EUROPEAN COMMUNITIES

THE THREE-DIMENSIONAL PLOTTING PROGRAM

TRICE

by

G. NASTRI and C. CERVINI

1970

Joint Nuclear Research Center
Geel Establishment - Belgium
Central Bureau for Nuclear Measurements - CBNM
LEGAL NOTICE

This document was prepared under the sponsorship of the Commission of the European Communities.

Neither the Commission of the European Communities, its contractors nor any person acting on their behalf:

make any warranty or representation, express or implied, with respect to the accuracy, completeness or usefulness of the information contained in this document, or that the use of any information, apparatus, method or process disclosed in this document may not infringe privately owned rights; or

assume any liability with respect to the use of, or for damages resulting from the use of any information, apparatus, method or process disclosed in this document.

This report is on sale at the addresses listed on cover page 4

<table>
<thead>
<tr>
<th>at the price of FF 6.65</th>
<th>FB 60.—</th>
<th>DM 4.40</th>
<th>Lit. 750</th>
<th>Fl. 4.30</th>
</tr>
</thead>
</table>

When ordering, please quote the EUR number and the title which are indicated on the cover of each report.


This document was reproduced on the basis of the best available copy.
The program TRICE (FORTRAN IV IBM 1800) projects a surface on a plane from a given observation point and plots the projected surface without hidden points.
THE THREE-DIMENSIONAL PLOTTING PROGRAM
TRICE

by

G. NASTRI and C. CERVINI

1970

Joint Nuclear Research Center
Geel Establishment - Belgium
Central Bureau for Nuclear Measurements - CBNM
ABSTRACT

The program TRICE (FORTRAN IV IBM 1800) projects a surface on a plane from a given observation point and plots the projected surface without hidden points.

KEYWORDS

FORTRAN
IBM
T-CODES
DIAGRAMS
RECORDING SYSTEMS
SURFACES
CONTENTS

Introduction 5

1. 1. The algorithm of the projection (Main Progr. TRICE and subroutine XRYR) 5
1. 2. Why the subroutine XRYR is called three times 6

2. Subroutine TRIFN for reading the coordinates of the surface \( \sigma \) to be plotted 7

3. 1. The visibility test performed by TRIVJ 7
3. 2. The first visibility test 8
3. 3. The second visibility test 9
3. 4. Interpolation formulas used in TRIVJ 10

4. How to use TRICE: time requirements and limitations 11

5. Input data 12
6. Block diagrams 14

7. Listing of the program TRICE 19

8. Example of plots done by TRICE 27

9. 1. Example of program writing the disk files TRIXY, TRIZ (Progr. USBES) 31
9. 2. Example of program writing the disk files TRIXY, TRIZ (Progr. TRI) 34
9. 3. Example of program writing a tape to be read by TRICE (Progr. ELLIN) 39

10. Examples of input data 45

Acknowledgements 46

References 46
Introduction *)

TRICE is a FORTRAN IV (IBM 1800) program performing the perspective transformation of the coordinates of a surface $\sigma$ viewed from a given observation point $C$ and projected into a plane $\pi$. No line connecting $C$ with a point of $\sigma$ can be parallel to $\pi$. The resulting projection is then plotted without hidden points. The set of routines under the name of TRICE includes a main program TRICE and three subroutines TRIFN, TRIVJ, XRYR. Additional subroutines used for the plot (MXMN, PLOT) are described in reference [1].

The main program TRICE reads as input data cards:

a) the coordinates of the observation point $C$;

b) the angles (in degrees) made by the line of sight (normal from $C$ to the projection plane $\pi$) with the three cartesian axes $x, y, z$;

c) the distance of the projection plane $\pi$ from the observation point $C$.

Then TRICE plots the projected points which are visible from the observation point $C$.

Two points of the projected surface $\sigma$ are connected by a continuous line only if they are both visible.

The perspective transformation of the coordinates of the points of $\sigma$ is executed by the subroutine XRYR.

The subroutine TRIFN reads as input data the coordinates of the points of the surface $\sigma$ (to be projected and plotted) from cards, tape or disk. The user must write the program to generate the surface $\sigma$ in a suitable form for TRIFN.

Examples follow in sections 9.1., 9.2., 9.3.

The subroutine TRIVJ determines whether a point is hidden or not and builds a matrix where hidden points are marked by zero and visible point by 1.

If requested, this matrix is printed out.

If ISEV = 0 no analysis of hidden points is done and all of the points are plotted.

1.1. The Algorithm of the Projection (Main Program TRICE and Subroutine XRYR)

In order to perform the projection of a point $P \equiv (x, y, z)$ on the plane $\pi$ we need the following data:

$x', y', z'$, the coordinates of the observation point $C$;

$\alpha, \beta, \gamma$, the angles made by the line of sight (normal to $\pi$ from $C$) with the axes $x, y, z$;

d, the distance of the projection plane $\pi$ from $C$.

*) Manuscript received on 13 March 1970
The formulas used are the following [2]:

\[ q_x = C_x + d \cos \alpha \]
\[ q_y = C_y + d \cos \beta \]
\[ q_z = C_z + d \cos \gamma \]

\[ K = \frac{d}{[(x-C_x)\cos \alpha + (y-C_y)\cos \beta + (z-C_z)\cos \gamma]} \]  

\[ \xi = C_x + K(x - C_x) \]
\[ \eta = C_y + K(y - C_y) \]
\[ \zeta = C_z + K(z - C_z) \]

\[ XR = \frac{[(\xi - q_x) \cos \beta - (\eta - q_y) \cos \alpha]}{\sin \gamma} \]  

\[ YR = \frac{(\zeta - q_z)}{\sin \gamma} \]

\[ XR, \ YR \] are the coordinates of the projection of \( P \) in the plane \( \pi \).

If \( \sin \gamma = 0 \) the perspective transformation is

\[ XR = \frac{[-(\xi - q_x) \cos \gamma + (\zeta - q_z) \cos \alpha]}{\sin \beta} \]
\[ YR = \frac{(\eta - q_y)}{\sin \beta} \]

The angles \( \alpha, \beta, \gamma \), must satisfy the condition

\[ \cos^2 \alpha + \cos^2 \beta + \cos^2 \gamma = 1. \]

1.2. Why the Subroutine XR, YR is Called Three Times

In order to save memory space, the coordinates of the projected points are calculated three times.

I. One for the search of the maximum and minimum values of \( XR, \ YR \). Such extreme values are required for the determination of the scale of the plot.

II. Then the values of \( XR, \ YR \) are again evaluated to plot the curves containing points with the same coordinate \( x \) (in the original cartesian reference). Each time a new curve (defined by the same \( x \)) must be drawn, the new values forming the vectors \( XR(l), \ YR(l) \) destroy the old ones.

III. We must evaluate a third time the whole set of vectors \( XR(l), \ YR(l) \) for tracing the curves defined by the same value of \( y \) (in the original cartesian reference). This way we may deal with a surface with up to \( 69 \times 69 = 4761 \) mesh points.
2. Subroutine TRIFN for Reading the Coordinates of the Surface $\sigma$ to be Plotted.

The surface $\sigma$ to be plotted is given by points. The coordinates $x, y$ of such points are defined by the mesh points of a grid (see fig; 1) and the coordinates $z$ correspond to them through the indices $I, J$. The coordinates $X(I), Y(J)$ of the grid as well as the corresponding $Z(I, J)$ of the surface $\sigma$ to be plotted may be taken from one of the following sources:

a) cards
b) tape (unit specified by input data NTAPE)
c) disk files TRIXY, TRIZ.

In the case a) it is even possible to read only the coordinates $z(I, J)$ from the cards and to calculate in the program the grid elements starting from the extreme values XMIN, XMAX; YMIN, YMAX.

In the cases b) and c) a previous program is supposed to have prepared the tape or disk.

Examples are given in sections 9.1., 9.2., 9.3.

The coordinates defining the grid and $\sigma$ can be optionally printed out.

3.1. The Visibility Tests Performed by TRIVJ

We develop our criterion of visibility by studying the line segment $CP$ which joins the observation point C with a variable point $P$ of the surface $\sigma$ to be plotted (cf. fig. 3).

If $CP$ has no intersections with the surface $\sigma$ the point $P$ is visible, otherwise it is hidden.

In order to determine the existence of such intersections we select suitable points $P_k$ of $CP$ ($k = 1, 2, \ldots$) identified by the coordinates $x_k, y_k$ and calculate the corresponding values of $z$ for $CP$ and for the surface $\sigma$; $z_k$ and $z^*$ are such values of the coordinate $z$ at $x_k, y_k$, respectively for $CP$ and for the surface $\sigma$.

If the sign of the difference

$$\delta_k = z_k - z^* \quad (k = 1, 2, \ldots) \quad (3.1.)$$

does not change over the whole range of index $k$, no intersection exists between $CP$ and $\sigma$: as a consequence the point $P$ is visible. The opposite conclusion is reached if $\delta_k$ changes its sign.

TRIVJ examines all of the points $P$ of $\sigma$ whose coordinates are defined by TRIFN. For each point $P$ TRIVJ creates one or two sets of points $P_k$ depending on the index ISEV given as input data.

There are one or two corresponding visibility tests.

a) The first set of points $P_k$ is defined by the abscisses

$$x_k = X(k) \quad (3.2.)$$

received from TRIFN: only the range is limited to the extension of the line segment $CP$.

The first visibility test assumes such set of points $P_k$. 
b) The second set $P_k$ has the ordinates

$$y_k = Y(K)$$

(3.3.)

as defined in TRIFN. While the visibility test associated with the first choice of $P_k$ is performed always for $ISEV > 0$ the second set $P_k$ as well as the associated test are considered only if $ISEV \leq 3,4$.

The index $ISEV$, called severity degree, specifies the requested severity in doing the visibility test. To the severity corresponds the time required. If $ISEV = 3,4$ the time of execution is practically doubled.

3.2. The First Visibility Test

For each point $P$ we assume at the beginning that it is visible by defining

$$JVIS(I, J) = 1$$

(3.4.)

If the sequence $\delta_k$ of eq. (3.1.) contains only one value of $\delta_k$ there is no possibility of modifying the condition (3.4.).

For making the best use of TRIVJ the coordinates $CX$ and $CY$ of the observation point $C$ and the coordinates of the grid should satisfy the conditions

$$CX < X(1) < X(2) < \ldots X(IX)$$

$$CY < Y(1) < Y(2) < \ldots Y(IY)$$

(3.5.)

as shown in fig. 1.)

---

Fig. 1

THE GRID
As already outlined the first test (Sect. 3.1.a)-based on evaluating $\delta_k$ at different values of the abscisses given by eq. (3.2.)-is always performed for $\text{ISEV} > 0$.

When the conditions (3.5.) hold, the curve belonging to the surface $\sigma$ and defined by

$$x = X(1)$$

is always visible, while the curve of $\sigma$ for

$$x = X(2)$$

gives rise to only one value $\delta_k'$, which is not sufficient for modifying the condition (3.4.). In this case, the only way for improving the visibility check is to perform the second test (Sect. 3.1.b).

In the case sketched by fig. 2 there is the possibility of creating

besides the value of $\delta_k'$ corresponding to the point $B$ another $\delta_k'$ for the point $A$ by extrapolating the surface $\sigma$ out of the definition area. TRIVJ performs this task if the option $\text{ISEV} = 2$ is expressed.

3.3. The Second Visibility Test

Assuming as points for evaluating $\delta_k'$ the set $P$ defined according to the eq. (3.3.), the second test (3.1.b) takes place if $\text{ISEV} > 2$. The extrapolation concerning only the points $P$ having abscissa $x = X(2)$ is performed under the option $\text{ISEV} = 4$.  

![Fig. 2](image-url)
3.4. Interpolation Formulas Used in TRIVJ

Let us consider the simple general formulas used for interpolation in the case of the first test (Sect. 3.1.a).

For the definition of the symbols refer to fig. 3.

At the beginning are given:

a) the coordinates of the points C, P;

b) the coordinates \( x_k, y_1, y_{1-1} \) belonging to the grid;

c) the coordinates \( z_1, z_{1-1} \) which define the surface \( \sigma \).

Then we determine

\[
\begin{align*}
    y_k &= \frac{y - C_y}{x - C_x} \cdot (x_k - C_x) + C_y \\
    z_k &= \frac{z - C_z}{x - C_x} (x_k - C_x) + C_z \\
    z &= \frac{z_1 - z_{1-1}}{y_1 - y_{1-1}} (y_k - y_{1-1}) + z_{1-1}
\end{align*}
\]
4. How to Use TRICE: Time Requirements and Limitations

The part of TRICE that consumes more time is the visibility test: for one test on 4761 points the IBM 1800 computer takes more than one hour and a half. Therefore it is more advisable to use ISEV > 2 only in special cases where no satisfaction was reached with ISEV ≤ 2. Nevertheless when the number of points is reduced the time requested decreases more than proportionally. In case of troubles the user can get better results by arranging the problem according to the scheme of eq. (3.5.) (scheme of fig. 1).

Peaks and edges defined by few points may still result "transparent" due to the adopted method of interpolation. The user will introduce more points in the definition of such features of the surface \( \sigma \).

If the denominator of the expression of \( K \) (eq. 1.1.) becomes zero, \( K \) itself loses its signification.

Reasons of such degeneration are:

I) the line \( CP \) connecting the observation point \( C \) with some point \( P \) of the surface \( \sigma \) is parallel to the plane \( \pi \), that is perpendicular to the line of sight;
II) the observation point \( C \) coincides with a point \( P \) of the surface \( \sigma \) where the surface itself is defined (coordinates \( x \) and \( y \) in a mesh point of the grid). Therefore the user must be careful to avoid such topics which will prevent the successful completion of the plot. The choice of the coordinates of \( C \) and of the direction of sight may affect the results negatively, but not the choice of the distance \( d \) between \( C \) and the projection plane \( \pi \); nevertheless \( d \) must not be zero.

A cutting value \( RANG \) is foreseen for keeping the plot within suitable limits allowing definition of the scales and execution of a plot although curtailed.

TRICE allows also projection on planes \( \pi \) parallel to the plane \( x, y \); in this case is \( \gamma = 0 \) and therefore instead of formulas (1.2.), (1.3.), formulas (1.4.), (1.5.) are used. The coordinate \( z \) of \( C \) must be definitively higher (or lower) than the coordinate \( z \) of the points of \( \sigma \).
5. **Input Data**

**First card**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>CX</td>
</tr>
<tr>
<td>11 - 20</td>
<td>CY</td>
</tr>
<tr>
<td>21 - 30</td>
<td>CZ</td>
</tr>
<tr>
<td>31 - 40</td>
<td>SIZX</td>
</tr>
<tr>
<td>41 - 50</td>
<td>SIZY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ICALL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TRIFN is called for reading a new surface σ</td>
</tr>
<tr>
<td>1</td>
<td>TRIFN is not called and the old points are treated again (different views of the same surface σ)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>D1</th>
<th>distance between the view point C and the projection plane π</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>assume the last visibility matrix which has been calculated</td>
</tr>
<tr>
<td>0.</td>
<td>complete plot including hidden points</td>
</tr>
<tr>
<td>1.</td>
<td>simple test without extrapolation</td>
</tr>
<tr>
<td>2.</td>
<td>simple test with extrapolation</td>
</tr>
<tr>
<td>3.</td>
<td>double test without extrapolation</td>
</tr>
<tr>
<td>4.</td>
<td>double test with extrapolation</td>
</tr>
</tbody>
</table>

-1, assume the last visibility matrix which has been calculated

**Second card**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>ALFA</td>
</tr>
<tr>
<td>11 - 20</td>
<td>BETA</td>
</tr>
<tr>
<td>21 - 30</td>
<td>GAMMA</td>
</tr>
<tr>
<td>31 - 40</td>
<td>RANG</td>
</tr>
</tbody>
</table>

N.B. Avoid view lines which could be perpendicular to some CP direction: danger of aborting the plot. The value of RANG is intended to cut down excessive values of XR(I), YR(J) arising from a wrong choice of the view point and view line. Avoid values of RANG which cut away useful parts of the plot or the whole plot itself. (See sect. 4).

If GAMMA = 0 or RANG ≤ 0 no control is done on RANG.

**Third card (optional)**

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 5</td>
<td>IREAD</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6 - 10</td>
<td>IX</td>
</tr>
<tr>
<td>11 - 15</td>
<td>IY</td>
</tr>
<tr>
<td>16 - 20</td>
<td>NTAPE</td>
</tr>
</tbody>
</table>
(-1, visibility matrix is printed; vectors $X(I)$, $Y(J)$ and matrix $Z(I, J)$ are printed with FORMAT (35F4.0)
(0, visibility matrix is printed; vectors $X(I)$, $Y(J)$ and matrix $Z(I, J)$ are printed with FORMAT (14E10.3)
(1, vectors $X(I)$, $Y(J)$ and matrix $Z(I, J)$ are printed with FORMAT (35F4.0); no visibility matrix printed.

N. B. Vectors $X(I)$, $Y(J)$ and matrix $Z(I, J)$ are printed only if TRIFN is called. The visibility matrix is printed only if TRIVJ is called.

Fourth card (optional) FORMAT (4F10.0), only if ICALL=0 and IREAD=0

1 - 10 XMIN minimum value of $X(I)$
11 - 20 XMAX maximum value of $X(I)$
21 - 30 YMIN minimum value of $Y(J)$
31 - 40 YMAX maximum value of $Y(J)$

If ICALL = 0 and IREAD = 1, enter instead of the fourth card, data $X(I)$, $Y(J)$ (separate sets of cards) with FORMAT (8E10.3).

If ICALL = 0 and IREAD = 0, 1, enter data $Z(I, J)$ (separate sets of cards for each value of $I$) with FORMAT (8E10.3).

Last card (to be introduced only for the first plot)
The last card indicates on which unit $I$ the plot is expected to be. Columns 1-3 contain * * $I$ ($I = 0, 1, 2, 3)$.

N. B. Repeat the complete scheme of input data (except the last card) for each plot requested. Add a blank card at the end of the whole input.
6. BLOCK DIAGRAMS

6.1 Main Program TRICE

START

999

Read CX, CY, CZ, SIZX, SIZY, ICALL, D1, ISEV

600

Read ALFA, BETA, GAMMA

Evaluate QX, QY, QZ

800

ICALL = 0

YES

CALL TRIFN

NO

900

ISEV ≤ 0

NO

CALL TRIVJ

YES

1000

Determination of Scale Factors

1100

a.

Plotting lines (with varying Y(J)) defined by a given X(I), I = 1, IX

b.

Plotting lines (with varying X(I)) defined by a given Y(J), J = 1, IY

c.

CALL FINIM (SIZX + 15., 0.)

700

End file plot tape

CALL EXIT
Program TRICE (Continuation 1)

Determination of Scale Factors

DO 52 I = 1, IX
CALL XRYR (I, J, O)
CALL MXMN (XR, J, 1, XMA1, XMI1)
CALL MXMN (YR, J, 1, YMA1, YMI1)
XMAX = XMA1
YMAX = YMA1
XMIN = XMI1
YMIN = YMI1

IF I <= 1 THEN
DO 52 I = 1, IX
END IF

XMAX < XMA1
YMAX < YMA1
XMIN < XMI1
YMIN < YMI1

DX = ABS ((XMAX - XMIN) / SIZX)
DY = ABS ((YMAX - YMIN) / SIZY)
Program TRICE (Continuation 2)

Plotting lines defined by a given $X(I)$

N.B. The same method is used to join with a full line the points with the same coordinate $Y(J)$. 
6.2 Subroutine TRIVJ

Entry

KSEV = ISEV / 2
LSEV = KSEV / 2

DO 10 I = 1, IX

X(I) ≤ CX

CONTINUE

INDX = IX + 1

Write CX, INDX

DO 110 J = 1, IX

Y(J) ≥ CY

CONTINUE

INDY = IY + 1

Write CY, INDY

Write: "Visibility Matrix"

48

Write JVIS(I, J) for J = 1, IY

DO 80 J = 1, IY

JVIS(I, J) = 1

DELTX = X(I) - CX

DELTY = Y(J) - CY

DELTZ = Z(I, J) - CZ

TEST calculated by interpolation at different values of X(K)

> 10^-3

TEST

64 ≤ 10^-3

ISEV

≤ 2

TEST calculated by interpolation at different values of Y(L)

65 > 2

TEST

63 > 10^-4

JVIS(I, J) = 0

CONTINUE

80

IPRI

90

CONTINUE

RETURN
Subroutine TRIVJ (Continuation)

TEST calculated by interpolation at different values of X(K)

```
DO 50 K = IX1, IX2
    DELTK = X(K) - CX
    YK = COEFY * DELTK + CX

DO 40 L = 2, 1Y
```

```
ZK = COEFZ * DELTM + CX
```

```
YK - Y(L)
```

```
ZAST = Z(K, L)
```

```
ZTEST(K) = ZAST - ZK
```

```
CONTINUE
```

```
52
```

```
SUMN = 0
SUMD = 0

DO 60 K = KMIN, KMAX
    SUMN = SUMN + ZTEST(K)
    SUMD = SUMD + ABS(ZTEST(K))

CONTINUE
```

```
TEST = ABS(1 - ABS(SUMN/SUMD))
```

N.B. The same method is used - if requested - to calculate TEST by interpolation at different values of Y(L). The only difference is that, contrary to KSVEV = 1, the case LSEV = 1 is not considered apart.
FOR TRICE

LIST SOURCE PROGRAM
ONE WORD INTEGERS
NONPROCESS PROGRAM

IOCS(CARD,1443 PRINTER)
DEFINE FILE 12(4800,2,U,IM)
DEFINE FILE 11(160,2,U,IM)
DIMENSION PLX(99),PLY(99)
COMMON XR(69),YR(69),ANG1,ANG2,ANG3,SANG1,QX,QY,QZ,D1,GAMMA,SANG2
COMMON Y(69),Z(69,69),JVIS(69,69),IX,IY,CX,CY,CZ,IPRI,ISEV
COMMON RANG

DEFINE FILE,11< 160,2,U,IM)
IM gi STION PLX(99),PLY(99)
COMMON XR(69),YR(69),ANG1,ANG2,ANG3,SANG1,QX,QY,QZ,D1,GAMMA,SANG2
COMMON Y(69),Z(69,69),JVIS(69,69),IX,IY,CX,CY,CZ,IPRI,ISEV

READ(1) CX,CY,CZ
READ(1) SIZX, SIZY, SIZZ, ICALL, DI, ISEV

1 FORMAT(5F10.0,10I0,5F10.0,10I0)

IF(SIZX)600,700,600

600 READ(5,1) ALFA,BETA,GAMMA,RANG
WRITE(6, 2)CX,CY,CZ,ALFA,BETA,GAMMA,D1
2 FORMAT( 1H0,8X,'CX·,8X,'CY·,8X,'CZ·, 6X,'ALFA',6X,'BETA',5X,'GAMMA'

1'8X,'D1',1H0,7F10.0///)

FACT=3.14159/180.
RADIA=ALFA*FACT
RADIB=BETA*FACT
RADIC=GAMMA*FACT
ANG1=COS(RADIA)
ANG2=COS(RADIB)
ANG3=COS(RADIC)
SANG1=SIN(RADIA)
SANG2=SIN(RADIB)
QX=CX+D1*ANG1
QY=CY+D1*ANG2
QZ=CZ+ANG3*DI

IF(ICALL)900,800,900

800 CALL TRIFN

900 DO 52 I=1,IX
CALL XRYR(I,J,0)
CALL MXMN(XR,IY,1,XMA1,XMI1)
CALL MXMN(YR,IY,1,YMA1,YMI1)
IF(I-1) 91,91,92
91 XMAX=XMA1
YMAX=YMA1
XMIN=XMI1
YMIN=YMI1
GO TO 52
92 IF(XMAX-XMA1)93,94,93
93 XMAX=XMA1
94 IF(XMIN-XMI1)95,95,96
95 XMIN=XMI1
96 IF(YMAX-YMA1)97,97,98
97 YMAX=YMA1
98 IF(YMIN-YMI1)52,52,100
100 YMIN=YMI1
52 CONTINUE

DX=ABS((XMAX-XMIN)/SIZX)
DY=ABS((YMAX-YMIN)/SIZY)
WRITE(6,6)XMIN,XMAX,YMIN,YMAX
6 FORMAT(1H0,4E12.5///)

IF(ISEV)1000,940,1100

940 DO 980 I=1,IX
DO 950 J=1,IY

980 CALL TRIFN

1100 STOP
JVIS(I,J)=1
950 CONTINUE
960 CONTINUE
go to 1000
1100 CALL TRIVJ
1000 DO 108 I=1,IX
    CALL XRYR(I,J,0)
    DO 105 J=1,IY
    PLX(J)=XR(J)/DX
    PLY(J)=YR(J)/DY
105 CONTINUE
    IF(JVIS(I,J))70,70,71
    71 CALL PLOT(PLX(J),PLY(J),2)
    GO to 106
    70 CALL PLOT(PLX(J),PLY(J),3)
106 CONTINUE
108 CONTINUE
    DO 60 J=1,IY
    CALL XRYR(I,J,1)
    DO 61 I=1,IX
    PLX(I)=XR(I)/DX
    PLY(I)=YR(I)/DY
61 CONTINUE
    CALL PLOT(PLX(I),PLY(I),3)
    IX=IX-1
    DO 107 I=1,IX
    81 IF(JVIS(I,J))80,80,81
    80 CALL PLOT(PLX(I),PLY(I),3)
    GO to 107
    81 CALL PLOT(PLX(I),PLY(I),2)
83 CALL PLOT(PLX(I),PLY(I),3)
80 CALL PLOT(PLX(I),PLY(I),3)
GO to 107
82 CALL PLOT(PLX(I),PLY(I),3)
81 I=I+1
107 CONTINUE
60 CONTINUE
    CALL FINIM(SIZX+15.,0.)
go to 999
700 CALL FINTR
    CALL EXIT
END

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CORE REQUIREMENTS FOR TRICE
COMMON 14868 INSKEI COMMON 0 VARIABLES 62 PROGRAM 820
*LIST SOURCE PROGRAM
*NONPROCESS PROGRAM
*ONE WORD INTEGERS

SUBROUTINE XRYR(I,J,K)
COMMON XR(69),YR(69),ANG1,ANG2,ANG3,SANG1,QX,QY,QZ,D1,GAMMA,SANG2
COMMON Y(69),X(69),Z(69,69),JVIS(69,69),IX,IY,CX,CY,CZ,IPRI,ISEV
COMMON RANG
IF(K)2,49,2
DO 53J=1,IY
FK=D1/((XI(I)-CX)*ANG1+(Y(J)-CY)*ANG2+(Z(I,J)-CZ)*ANG3)
PX= CX+FK *(XI(I)-CX)
PY= CY+FK *(Y(J)-CY)
PZ= CZ+FK *(Z(I,J)-CZ)
IF(GAMMA)299,300,299
XR(J)=((-QX+PX)*ANG3+(PZ-QZ)*ANG1)/SANG2
YR(J)=(PY-QY)/SANG2
GO TO 53
299 XR(J)=((PX -QX )*ANG2-(PY-QY)*ANG1)/SANG1
IF(RANG)54,54,51
51 AB=ABS(XR(J))
IF(AB=RANG)54,54,55
55 XR(J)=RANG*XR(J)/AB
54 YR(J)=(PZ-QZ)/SANG1
IF(RANG)53,53,57
57 AB=ABS(YR(J))
IF(AB=RANG)53,53,56
56 YR(J)=RANG*YR(J)/AB
53 CONTINUE
RETURN
END

FEATURES SUPPORTED
NONPROCESS ONE WORD INTEGERS

CORE REQUIREMENTS FOR XRYR
COMMON 14868 INSKEL COMMON 0 VARIABLES 18 PROGRAM 558

END OF COMPILATION
// FOR TRIFN
*LIST SOURCE PROGRAM
*NONPROCESS PROGRAM
*ONE WORD INTEGERS

SUBROUTINE TRIFN
COMMON XR(69), YR(69), ANG1, ANG2, ANG3, SANG1, OX, OY, OZ, D1, GAMMA, SANG2
COMMON Y(69), X(69), Z(69, 69), JV1(69, 69), IX, IY, CX, CY, CZ, IPRI, ISEV
READ(5,1) IREAD, IX, IY, NTAPIE, IPRI
1 FORMAT(5I5)
   IF(IREAD)20,20,24
20 GO TO(2,3,5), IREAD
   N=5
20 GO TO 4
3 N=NTAPIE
RE WIND NTAPIE
4 READ (N+10) (X(I), I = 1, IX)
     READ(N+10) (Y(J), J = 1, IY)
10 FORMAT(8F10.3)
   GO TO 25
20 N=5
   READ(5,21) XMIN, XMAX, YMIN, YMAX
21 FORMAT(8F10.0)
     XIX=IX-1
     XSTEP=(XMAX-XMIN)/XIX
     X(I)=XMIN
     DO 22 I=2, IX
22 X(I)=X(I-1)+XSTEP
     YIY=IY-1
     YSTEP=(YMAX-YMIN)/YIY
     Y(I)=YMIN
     DO 23 J=2, IY
23 Y(J)=Y(J-1)+YSTEP
25 DO 50 I=1, IX
50 READ(N,10) (Z(I,J) , J=1, IY)
   GO TO 6
5 IM=1
   READ (11*IMX(I), I = 1, IX), (Y(J), J = 1, IY)
   IM=1
   READ (12*IM)((Z(I,J), J = 1, IY), (I = 1, IX)
6 IF(IPR)82,99,82
99 WRITE(6,221) (X(I), I = 1, IX)
221 FORMAT(1HO,'TABLE OF X'//(4X,14E10.3))
   WRITE(6,222) (Y(J), J = 1, IY)
222 FORMAT(1HO,'TABLE OF Y'//(4X,35F4.0))
   WRITE(6,91)
91 FORMAT(1HO,'MATRIX Z'//IHO,'I ** I IS THE INDEX OF X(I) ** FOR E
   EACH I, VALUES OF Z(I,J) ARE GIVEN WITH J RUNNING FROM 1 TO IY ** ')
   DO 62 I=1, IX
51 L=I
   WRITE(6,122) L, (Z(I,J), J = 1, IY)
122 FORMAT(1HO,I2,1X ,14E10.3//(4X,14E10.3))
   CONTINUE
62 GO TO 84
84 WRITE(6,92) (X(I), I = 1, IX)
92 FORMAT(1HO,'TABLE OF X'//(4X,35F4.0))
   WRITE(6,92) (Y(J), J = 1, IY)
92 FORMAT(1HO,'TABLE OF Y'//(4X,35F4.0))
   WRITE(6,91)
1 FORMAT(35F4.0)
}
C TRIVJ DECIDES ABOUT THE VISIBILITY OF POINT P(X,Y,Z) FROM C(CX,CY,CZ)

DO 10 I=1,IX
   IF (CX-X(I))4,4,10

10 CONTINUE

INDX=IX+1
WRITE(6,12)CX,INDX
12 FORMAT(1HO,'TO CX =',F10.2,' CORRESPONDS INDX =',I6)
DO 110 J=1,IY
   IF (CY-Y(J))104,104,110

104 INDY=J
   GO TO 111

110 CONTINUE

INDY=IY+1
WRITE(6,112)CY,INDY
112 FORMAT(1HO,'TO CY =',F10.2,' CORRESPONDS INDY =',I6/)
WRITE(6,72)
72 FORMAT(1HO,'VISIBILITY MATRIX ')//4H I//

DO 90 I=1,IX

DO 80 J=1,IY
90 CONTINUE

C ** * THE INDICES I,J IDENTIFY THE POINT P(X,Y,Z) TO BE EXAMINATED

JVIST(I,J)=1
DELTX=X(I)-CX
DELTY=Y(J)-CY
DELTZ=Z(I,J)-CZ

IF (CX-X(I))32,64,33
32 IX1=INDX
IX2=1-1
GO TO 34
33 IX1=INDX-1
IX2=INDEX+1
34 KMAX=IX1
COEFY=DELTY/DELTX
COEFZ=DELTZ/DELTX
INDK=1
IF (IX1-IX2)31,64,64
31 CONTINUE

DO 50 K=IX1,IX2
   DELTK=X(K)-CX
   YK=COEFY*DELTK+CY
   IF (YK-Y(1))35,36,36
35 DELTM=DELTK
YM=YK
KMTN=K+1
GO TO (6,7), INDK
36 INDK = 2
37 KMIN = K
38 IF(K-IX1)7,7,8
39 IF(KSEV-1)7,9,5
40 IF(J-2)7,9,7
41 KMIN = K-1
42 M = KMIN
43 ZM = COEFZ*DELTM+CZ
44 ZAST = (YM-Y(2))*(Z(M,2)-Z(M,1))/(Y(2)-Y(1))+Z(M,2)
45 ZTEST(M) = ZAST-ZM
46 C **** C,P AND THE POINT OF COORDINATES (X(K),Y(K),Z(K)) ARE ON THE SAME STRAIGHT LINE
47 ZK = COEFZ*DELTK+CZ
48 DO 40 L = 2, IY
49 IF(YK-Y(L))37,38,40
50 ZAST = (YK-Y(L-1))*(Z(K,L)-Z(K,L-1))/(Y(L)-Y(L-1))+Z(K,L-1)
51 C * YK IS BETWEEN Y(L-1) AND Y(L) GO TO 43
52 ZAST = Z(K,L)
53 GO TO 43
54 CONTINUE
55 GO TO 50
56 ZTEST(K) = ZAST-ZK
57 KMAX = K
58 CONTINUE
59 IF(KMIN-KMAX) 52,64,64
60 SUMN = 0
61 SUMD = 0
62 DO 60 K = KMIN,KMAX
63 SUMN = SUMN + ZTEST(K)
64 SUMD = SUMD + ABS(ZTEST(K))
65 CONTINUE
66 IF(SUMD-0.000001)64,64,63
67 TEST = ABS(1.-ABS(SUMN/SUMD))
68 IF(TEST-0.001)64,64,63
69 IF(IUSEV-2)80,80,65
70 IF(CY-Y(J))132,80,133
71 IY1 = INDY
72 IY2 = J-1
73 GO TO 134
74 IY1 = J+1
75 IY2 = INDY-1
76 LMAX = IY1
77 COEFX = DELTX/DETY
78 COEFZ = DELTZ/DETY
79 INDL = 1
80 IF(IY1-IY2)131,80,80
81 CONTINUE
82 DO 150 L = IY1,IY2
83 DELTL = Y(L)-CY
84 XL = COEFX*DETL+CX
85 IF(XL-X(I))135,136,136
86 DELTM = DELTL
XM=XL
LMIN=L+1
GO TO 150
136 GO TO (106,107),INDL
106 INDL=2
LMIN=
108 IF(LSEV-1)107,105,105
105 IF(I-2)107,109,107
109 LMIN=L-1
M=LMIN
ZM=COEFZ*DELTM+CZ
ZAST=(XM-X(2))*(Z(2,M)-Z(1,M))/(X(2)-X(1))+Z(2,M)
ZTEST(M)=ZAST-ZM
C **** C P AND THE POINT OF COORDINATES (XL,Y(L),ZL) ARE ON THE SAME
C STRAIGHT LINE
107 ZL=COEFZ+DETL+o
DD=2. IX
137 ZAST=(XL-X(K-1))*(Z(K,L)-Z(K-1,L))/(X(K)-X(K-1))+Z(K-1,L)
C * XL IS BETWEEN X(K-1) AND X(K)
GO TO 143
138 ZAST=Z(K,L)
GO TO 143
140 CONTINUE
GO TO 150
143 ZTEST(L)=ZAST-ZL
LMAX=L
150 CONTINUE
IF(LMIN- LMAX)152,80,80
152 SUMN=0
SUMD=0
DO 160 K=LMIN,LMAX
SUMN=SUMN+ZTEST(K)
SUMD=SUMD+ABS(ZTEST(K))
160 CONTINUE
IF(SUMD-0.000001)80,80,61
61 TEST=ABS(1.-ABS(SUMN/SUMD))
62 IF(TEST-0.0001)80,80,63
63 JVIS(I,J)=0
80 CONTINUE
IF(IPRI) 48,48,90
48 CONTINUE
WRITE(6,81)1,JVIS(I,J),J=1,IY
81 FORMAT(35I4/(I4X,34I4))
90 CONTINUE
RETURN
END

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS

CORE REQUIREMENTS FOR TRIVJ
COMMON 14868 INSKEL COMMON  0 VARIABLES 268 PROGRAM 1180
8. Example of Plots Done by TRICE

Figs. 4, 5: The Bessel function J (cf. Sect. 9.1.)
Fig. 6: Surface calculated by TRI (cf. Sect. 9.2.)

Fig. 7: The sections of the galleries are parabolas.
Fig. 8: Parabolic galeries viewed inside.

Fig. 9: The elliptic-integral of the first kind (cf. Sect. 9.3.)
Fig. 10: Plotting data from cards.
9.1. Example of Program Writing the Disk Files TRIXY, TRIZ (Program USBES)

USBES (calling the standard IBM subroutine BESJ, [3]) evaluates as \( z(x, y) \) the \( J \) Bessel function for given values of \( x \) and orders \( y \). The tables of \( x, y, z \) are then optionally saved on disk for later use in TRICE.

**Input of USBES**

<table>
<thead>
<tr>
<th>One card only</th>
<th>FORMAT (7F10.0,15)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 - 10 STEPX</td>
<td>interval between two consecutive values of ( x )</td>
</tr>
<tr>
<td>11 - 20 STEPY</td>
<td>difference between the order ( y ) of two consecutive Bessel functions calculated (STEPY integer but introduced as floating)</td>
</tr>
<tr>
<td>21 - 30 YMX</td>
<td>maximum order ( y ) of Bessel functions</td>
</tr>
<tr>
<td>31 - 40 XMX</td>
<td>maximum value of ( x )</td>
</tr>
<tr>
<td>41 - 50 YMI</td>
<td>minimum order ( y ) of Bessel functions</td>
</tr>
<tr>
<td>51 - 60 XMI</td>
<td>minimum value of ( x )</td>
</tr>
<tr>
<td>61 - 70 ACCRY</td>
<td>the requested accuracy (e.g. 0.01)</td>
</tr>
<tr>
<td>71 - 75 IPRI =</td>
<td></td>
</tr>
<tr>
<td>0, save on disk ( x, y, z )</td>
<td></td>
</tr>
<tr>
<td>1, print and save on disk ( x, y, z )</td>
<td></td>
</tr>
</tbody>
</table>

**Limitations for the Choice of \( y \)**

- \( y \) integer
- \( y > 0 \), always
- \( y < 20 + 10 \cdot \frac{x-x^2}{3} \), for \( x \leq 15 \)
- \( y < 90 + \frac{x}{2} \), for \( x > 15 \).
// JOB XXX
// FOR USBES
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
*IOCS(CARD,1443 PRINTER,DISK)
*LIST SOURCE PROGRAM

DEFINE FILE 11(160,2,U,IM)
DEFINE FILE 12(4800,2,U,IM)
COMMON X(69),Y(69),1(69,69)
READ(5,1)STEPX,STEPY,XMX,XMI,YMX,YMI,ACCRY,IPRI

1 FORMAT(7F10.0,2I5)
IX=ABS(XMX-XMI)/STEPX
IY=ABS(YMX-YMI)/STEPY
N=YMI
DO 21 J=1,IY
XX=XMI
DO 20 I=1,IX
IER=0
CALL BESJ(XX,N,BJ,ACCRY,IER)
XX=XX+STEPX
Z(I,J)=BJ
20 CONTINUE
Y(J)=N
N=N+STEPY
21 CONTINUE
X(1)=XMI
DO 22 I=2,IX
X(I)=X(I-1)+STEPX
22 CONTINUE
IF(IPRI)24,23,24 N
WRITE(6,71)(X(I),I=1,IX)
71 FORMAT(1H0,'TABLE OF X'/(4X,28F5.1))
WRITE(6,72)(Y(J),J=1,IY)
72 FORMAT(1H0,'TABLE OF Y'/(4X,28F5.1))
WRITE(6,91)
91 FORMAT(1H0,'MATRIX Z'/1H0,'I * * I IS THE INDEX OF X(I) * * FOR E
ACH 1, VALUES OF Z(I,J) ARE GIVEN WITH J RUNNING FROM 1 TO IY * *')
DO 140 I=1,IX
WRITE(6,122) I,(Z(I,J),J=1,1Y)
122 FORMAT(1H0,14E10.3/(4X,14E10.3))
140 CONTINUE
IF(IPRI)25,26,26 IM=1
WRITE(11,'IM')(X(I),I=1,IX),(Y(J),J=1,1Y)
IM=1
WRITE(12,'IM')((Z(I,J),I=1,IX),J=1,1Y)
25 CALL EXIT
END

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CORE REQUIREMENTS FOR USBES
COMMON 9798 INSHEL COMMON 0 VARIABLES 44 PROGRAM 468

END OF COMPILATION
SUBROUTINE BESJ(X,N,BJ,D,IER)

BJ=.0
IF(N)10,20,20
10 IER=1
RETURN
20 IF(X) 30,30,31
30 IER=2
RETURN
31 IF(X-15.)32,32,34
32 NTEST=20.+10.*X-X**2/3
GO TO 36
34 NTEST =90.+X/2.
36 IF(N-NTEST) 40,38,38
38 IER=4
RETURN
40 IER=0
41 N1=N+1
BPREV=.0
IF(X-5.)50,60,60
50 MA=X+6.
GO TO 70
60 MA=1.4*X+60./X
70 MB=N+IFIX(X)/4+2
MZERO=MA
IF(MA-MB)80,90,90
80 MZERO=MB
90 MMAX=NTEST
DO 190 M=MZERO,MMAX,3
FM1=1.0E-28
FM=.0
ALPHA=.0
IF(M-(M/2)*2)120,110,120
110 JT=-1
GO TO 130
120 JT=1
130 M2=M-2
DO 160 K=1,M2
MK=M-K
BMK=2.*FLOAT(MK)*FM1/X-FM
FM=FM1
FM1=BMK
IF(MK-N-1)150, 140,150
140 BJ=BMK
150 JT=-JT
S=1+JT
160 ALPHA=ALPHA+BMK*S
BMK=2.*FM1/X-FM
IF(N)180,170,180
170 BJ=BMK
180 ALPHA=ALPHA+BMK
BJ=BJ/ALPHA
IF(ABS(BJ-BPREV)-ABS(D*BJ))200,200,190
190 BPREV=BJ
IER=3
200 RETURN
END
9.2. Example of Program Writing the Disk Files TRIXY, TRIZ
(Program TRI)

TRI (FORTRAN 4, IBM 1800) builds two sets of coordinates x and y
defining points on a grid (cf. fig. 1) having as symmetry axes the
axes x and y.
A function \( z(x, y) \) describing a surface \( \sigma \) is then calculated in such
points under the same conditions of symmetry (cf. fig. 1).
The surface \( \sigma \) consists of a dome obtained by rotation of a parabola
around the axis z: non cylindrical galeries enter into the dome.
Each gallery is symmetrical with respect to a different vertical
plane \( \tau \) containing the axis z. The sections of a single gallery nor-
mal to its symmetry plane \( \tau \) are hyperbolic cosinus. Let us call
axis of a gallery the intersection of the plane \( \tau \) with the plane x, y.

Input of TRI

<table>
<thead>
<tr>
<th>First card</th>
<th>FORMAT(5F10.0, 515)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>RMAX</td>
</tr>
<tr>
<td>11 - 20</td>
<td>HCUP</td>
</tr>
<tr>
<td>21 - 30</td>
<td>STEPX</td>
</tr>
<tr>
<td>31 - 40</td>
<td>STEPY</td>
</tr>
<tr>
<td>41 - 50</td>
<td>DENS</td>
</tr>
<tr>
<td>51 - 55</td>
<td>NX</td>
</tr>
<tr>
<td>56 - 60</td>
<td>NY</td>
</tr>
<tr>
<td>61 - 65</td>
<td>NB</td>
</tr>
<tr>
<td>66 - 70</td>
<td>IPRI</td>
</tr>
<tr>
<td>71 - 75</td>
<td>IESP</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Second card</th>
<th>FORMAT(3F10.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>ANG(L)</td>
</tr>
<tr>
<td>11 - 20</td>
<td>AMPL(L)</td>
</tr>
<tr>
<td>21 - 30</td>
<td>LUNG(L)</td>
</tr>
</tbody>
</table>

N. B. \( L \) runs from 1 to NB, i.e. concerns only the galeries
occupying (though partially) the first quadrant. Then
this card must be entered NB times.
// FOR TRI
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
*IOCS(CARD,1,443 PRINTER,DISK)
DEFINE FILE 11(.160,2,0,1,0)
DEFINE FILE 12(.4800,2,0,1,0)
REAL MARG,LUNG(6)
DIMENSION M(7),ANG(6),AMF(6),TANG(6),ZM(7),P(6),Q(6)
COMMON C1(69),COEF1(69),RCUP(69),RAXIS(69)
COMMON Y(69),X(69),Z(69),69)
EQUIVALENCE(TANG(1),AMPL(1))
READ(5,1)RMAX,HCP,STEPS,STEPS,DENS,NX,NY,NB,IPRI,IESP
1 FORMAT(5F10.3,161)
A=HCP/(RMAX*RMAX) AM=AM+A COEFF=DENS/(3.*AM)
EPSI=1./(10.**IESP)
WRITE(6,16)RMAX
16 FORMAT(IHO,'RMAX =',F8.2)
C OCCORRE UNA TAVOLA CHE DIA I DUE PARAMETRI COEF1 E C1 IN FUNZIONE DI RAXIS
NR=NX+NY
DELTR=RMAX/NR
RCUP(1)=0
DO 120 1=1,NR
RCUP(I+1)=RCUP(I)+DELTR
120 CONTINUE
NR=NR+1
DO 68 I=1,NR
R=RCUP(I)
SQRR=R*R
ZCUP=HCP-A*SQRR
RAM =R*AM
FACT=1.+RAM*RAM
ROOT=SQRT(FACT)
XX=ABS(RAM)-EPSI
IF(XX)54,54,55
54 CONTINUE
TMUZ=0
TMUR=-DENS/(AM+AM)
COIVA=1.618033988
SIVA=1.272019650
GO TO 56
55 TMUZ =COEFF/RAM*( FACT*ROOT- 1. )
TMUR =-TMUZ /RAM
65 DTMX=3*COEFF*ROOT+TMUR
DTMZ=DTMX/RMUR
TPAR=DTMZ*AM
GAMMA=1.-DENS*ROOT/DTMZ
GAMMA=-GAMMA
DELA=SQR(GAMMA*GAMMA+4.)
COEFY=GAMMA*SQRR/2.
ID=1
YY=-COEFY*(DETA+GAMMA)
IF(YY)63,64,64
63 YY=COEFY*(-GAMMA+DETA)
IF(XX)74,74,75
75 CONTINUE
COIWA = R / SQRT(SQRR - YY)

CONTINUE

PYTAN = -TPAR * ROOT * COIWA
PYVER = TNUZ * SIVA
ALPHA = ATAN(COIVA / SIVA)
B = PYVER / PYTAN
B = B + B
DISCR = SQRT(B * B + 4.)
IF(DISCR + B 66, 66.65
WRITE(6, 67) B, DISCR
67 FORMAT(2E12.2)
GO TO 68
CONTINUE
VAV = PYTAN / DEQ1
DENS1 = VAV * ALOG((B + DISCR) / 2.)
ARGE1 = DENS1 / VAV
E1 = EXP(ARGE1)
COEF1(I) = PYTAN / DENS1
C1(I) = ZCUP - (E1 + 1. / E1) / 2. * COEF1(I)
RAXIS(I) = R / COIVA
CONTINUE
READ(5, 2)(ANG(L), AMPL(L), LUNG(L), L=1, NB)
2 FORMAT(3F10.5)
DO 5 L=1, NB
ANGOL = ANG(L)
ANGOL = ANGOL * 3.1415927 / 180.
P(L) = SIN(ANGOL)
Q(L) = COS(ANGOL)
AMP = AMPL(L)
AMP = AMP * 3.1415927 / 180.
TANG(L) = SIN(AMP) / COS(AMP)
WRITE(6, 1) ANG(L), ANGOL, P(L), Q(L), TANG(L)
DO 5 L=1, NB
5 CONTINUE
IX1 = NX + 1
IX0 = NY + 2
IX2 = NX + NX + 1
IY1 = NY + 1
IYO = NY + 2
IY2 = NY + NY + 1
X(IX1) = 0
Y(IY1) = 0
DO 10 I = IX0, IX2
X(I) = X(I-1) + STEPX
CONTINUE
DO 20 J = IYO, IY2
Y(J) = Y(J-1) + STEPY
CONTINUE
M(NB+1) = 0
ZM(NB+1) = 0
Z1(I, J) = 0
SQRR = X(I) * X(I) + Y(J) * Y(J)
R = SQRT(SQRR)
\[ M(NB+1) = 1 \]

\[ ZCUP = HCUP - A * SQRR \]

\[ ZM(NB+1) = ZCUP \]

\[ DO 40 L = 1, NB \]

\[ M(L) = 0 \]

\[ ZM(L) = 0 \]

\[ T = \theta \Gamma_{1}^{2} + \chi_{1} + P(L) * Y(J) \]

\[ IF(T > LUNG(L)) \rightarrow 37, 37, 40 \]

\[ D = P(L) * X(I) - Q(L) * Y(J) \]

\[ D = ABS(D) \]

\[ MARG = T * \tan(L) \]

\[ IF(D > MARG - EPSI) \rightarrow 39, 40, 40 \]

\[ M(L) = 1 \]

C **RICERCA NELLA TAVOLA DI RAXIS E DEI CORRISPONDENTI PARAMETRI C1 E COEF1**

\[ DO 130 II = 1, NR \]

\[ IF(T > RAXIS(II)) \rightarrow 125, 124, 130 \]

124 \[ II = II \]

125 \[ II = II \]

130 \[ CONTINUE \]

131 \[ COEFF = COEF1(II) \]

132 \[ C = C1(II) \]

133 \[ CONTINUE \]

134 \[ DELTT = (T - RAXIS(II-1)) / (RAXIS(II) - RAXIS(II-1)) \]

135 \[ COEFF = COEF1(II-1) + (COEF1(II) - COEF1(II-1)) * DELTT \]

136 \[ E1 = \exp(D / COEFF) \]

\[ ZM(L) = COEFF * (E1 + 1 / E1) / 2 + C \]

\[ DO 100 J = IY1, IY2 \]

100 \[ CONTINUE \]

C **COSTRUZIONE DELLA PARTE SIMMETRICA RISPETTO ALL'ASSE Y**

\[ DO 110 I = 1, NX \]

110 \[ X(I) = -X(11) \]

\[ DO 100 J = IY1, IY2 \]

100 \[ CONTINUE \]

C **COSTRUZIONE DELLA PARTE SIMMETRICA RISPETTO ALL'ASSE X**

\[ DO 121 J = 1, NY \]

121 \[ Y(J) = -Y(JJ) \]

\[ DO 133 I = 1, IX2 \]

133 \[ CONTINUE \]

IX = IX2

IY = IY2
WRITE(6,21) (X(I), I=1,IX)
21 FORMAT(1HO,'TABLE OF X'/(4X,28F5.1))
WRITE(6,22) (Y(J), J=1,IY)
22 FORMAT(1HO,'TABLE OF Y'/(4X,28F5.1))
WRITE(6,91)
91 FORMAT(1HO,'MATRIX Z'/(1HO,'I '*'I IS THE INDEX OF X(I) '*' FOR EACH I VALUES OF Z(I,J) ARE GIVEN WITH J RUNNING FROM 1 TO IY '*' )
DO 140 I=1,IX
WRITE(6,122) I,(Z(I,J), J=1,IY)
122 FORMAT(1HO,I3,14E10.3/(4X,14E10.3))
140 CONTINUE
IM=1
WRITE(11,IM) (X(I), I=1,IX), (Y(J), J=1,IY)
IM=1
WRITE(12,IM) ((Z(I,J), I=1,IX), J=1,IY)
CALL EXIT
END

FEATURES SUPPORTED
NONPROCESS
ONE WORD INTEGERS
IOCS

CORE REQUIREMENTS FOR TRI
COMMON 10350 INSKEI COMMON 0 VARIABLES 212 PROGRAM 1674

END OF COMPILATION
9.3. Example of Program Writing a Tape to be Read by TRICE (Program ELLIN)

ELLIN (FORTRAN 4, IBM 1800) evaluates in the first quadrant of the complex plane defined by \( t = (x(I), y(J)) \) the normal elliptic integral of the first kind

\[
\int_0^t \frac{dt}{\sqrt{(1-t^2)(1-k^2 t^2)}}
\]

The real value of \( k \) built in is

\( k = 0.8 \).

Therefore both the singularities belong to the real axis in the points \( x=1, x=1.25 \). Only one branch of the function \( u(t) \) is considered, corresponding to the operation

\[
(1-t^2)(1-k^2 t^2) = \rho e^{i\varphi}
\]

\[
(1-t^2)(1-k^2 t^2) = \rho^{1/2} e^{i\varphi/2}
\]

The subroutines required for the complex operations are CXSUB, CXMPY, CXDIV, CXRPW, CXABS, CXPOL, CXCRT [4].

Input of ELLIN

Only one card FORMATT(610)

<table>
<thead>
<tr>
<th>Card</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - 10</td>
<td>IO = 1</td>
</tr>
<tr>
<td>11 - 20</td>
<td>JO = 1</td>
</tr>
<tr>
<td>21 - 30</td>
<td>IX = total number of points along the real axis (IX≤28)</td>
</tr>
<tr>
<td>31 - 40</td>
<td>IY = total number of points along the imaginary axis (IY≤28)</td>
</tr>
<tr>
<td>41 - 50</td>
<td>IND = 1, the step between two values of x and y is STEP=0.125 = 2, the step between two values of x and y is STEP=0.0625 = 3, the step between two values of x and y is STEP=0.03125</td>
</tr>
<tr>
<td>51 - 60</td>
<td>NTAPE = logic tape unit for the output (NTAPE = 7, 8, 9, 10)</td>
</tr>
</tbody>
</table>
JOB XXX
// FOR ELLIN
*IOCS(CARD.1443 PRINTER, MAGNETIC TAPE)
*LIST SOURCE PROGRAM
*ONE WORD INTEGERS
*NONPROCESS PROGRAM
DIMENSION Z2(2), ONE(2), FIRST(2), SECON(2), STO(2), F(28), G(28), S(2)
U(28,28), V(28,28), Z(55,55), REAL(55), AIMM(55), X(55), Y(55), AK(2),
A2(2), DEN(2)
READ(5,1) IO, JO, IX, IY, IND, NTAPE

1 FORMAT (8110)
IO = 1
JO = 1
M = IX - IO + 1
N = IY - JO + 1
AK(1) = 0.64
AK(2) = 0.
ONE(1) = 1.
ONE(2) = 0.
HALF = 0.5
GO TO (6, 7, 8), IND

6 STEP = 0.125
GO TO 9
7 STEP = 0.0625
GO TO 9
8 STEP = 0.03125
9 X(1) = STEP/2.
X(2) = STEP
DO 10 K = 3, M
10 X(K) = X(K-1) + STEP
Y(1) = STEP/2.
Y(2) = STEP
DO 11 L = 3, N
11 Y(L) = Y(L-1) + STEP
C EVALUATE THE FUNCTION TO BE INTEGRATED ALONG THE IMAGINARY AXIS

T(1) = 0.
T(2) = Y(L)
DO 12 L = 1, N
12 T(2) = Y(L)
CALL CXMPY(T, T, Z2)
CALL CXSUB(ONE, Z2, FIRST)
CALL CXMPY(AK, Z2, A2)
CALL CXSUB(ONE, A2, SECON)
CALL CXMPY(FIRST, SECON, S)
CALL CXRPW(S, DEN, HALF)
CALCXDIV(ONE, DEN, STO)
F(L) = STO(1)
G(L) = STO(2)
CONTINUE

C EVALUATION OF THE FUNCTION TO BE INTEGRATED OUT OF THE AXES
DO 14 K = 1, M
DO 15 L = 1, N
14 T(1) = X(K)
15 T(2) = Y(L)
CALL CXMPY(T, T, Z2)
CALL CXSUB(ONE, Z2, FIRST)
CALL CXMPY(AK, Z2, A2)
CALL CXSUB(ONE, A2, SECON)
CALL CXMPY(FIRST, SECON, S)
CALL CXRPW(S, DEN, HALF)
CALCXDIV(ONE, DEN, STO)
U(K, L) = -STO(1)
V(K, L) = -STO(2)
WRITE(6,19)T,Z2,FIRST,A2,SECON,S,DEN

CONTINUE

RXI=STEP/6.
DY=DIX
DX=DXI+DXX
D=DX

THE FIRST INTEGRATION TAKES PLACE ALONG THE IMAGINARY AXIS

REAL(J0)=0
AIIIMM(J0)=0

INTEGRATION FOR THE FIRST POINT OF IMAGINARY AXIS (J1=1) * HALF STEP
J1=J0+1
J=J1

REAL(J)=(0.+4*G(L)+G(L+1))*(-DY)
AIIIMM(J)=(1.+4*F(L)+F(L+1))*DY

INTEGRATION CONTINUES WITH NORMAL STEP * DEFINE AGAIN F(1),G(1),Y(1)

WRITE(6,19)F(1),G(1)

FORMAT(4X,14E10.3)

REAL(L)=0.
F(L)=1.
G(L)=0.

WRITE(6,21)(F(L),L=1,N)

FORMAT(1HO,'VECTOR F * REAL PART OF THE INTEGRAND * AXIS Y'/
1(4X,14E10.3))

WRITE(6,22)(G(L),L=1,N)

FORMAT(1HO,'VECTOR G * IMAGINARY PART OF THE INTEGRAND * AXIS Y'/
1(4X,14E10.3))

INTEGRATION CONTINUES ALONG THE REAL AXIS (J2=J1+1)

L=1
DO 40 J=J2,IY
REAL(J)=(REAL(J-1)+2.*REAL(J)+REAL(J+1))/4.
AIIIMM(J)=(AIIIMM(J-1)+2.*AIIIMM(J)+AIIIMM(J+1))/4.
40 CONTINUE

INTERPOLATION FOR SMOOTHING VALUES OF REAL, AIIIMM

IY1=IY-1
DO 50 J=J1,IY1
REAL(J)=(REAL(J-1)+2.*REAL(J)+REAL(J+1))/4.
AIIIMM(J)=(AIIIMM(J-1)+2.*AIIIMM(J)+AIIIMM(J+1))/4.
50 CONTINUE

Z(I0,J0)=SQRT(REAL(J)*REAL(J)+AIIIMM(J)*AIIIMM(J))

WRITE(6,41)(REAL(J),J=1,IY)

FORMAT(1HO,'VECTOR REAL * INTEGRAL ALONG AXIS Y'/(4X,14E10.3))

WRITE(6,42)(AIIIMM(J),J=1,IY)

FORMAT(1HO,'VECTOR AIIIMM * INTEGRAL ALONG AXIS Y'/(4X,14E10.3))

GO TO(51,52,53),IND

ISIN1=I0+8
ISIN2=I0+10
GO TO 54

ISIN1=I0+16
ISIN2=I0+20
GO TO 54

ISIN1=I0+32
ISIN2=I0+40
54 Z(I0,J0)=0

WRITE(6,56)
evaluate the elliptic integral in the first quadrant

\[ K = 1 \]
\[ L = J + 1 \]
\[ R = \text{REAL}(J) + (F(L) + 4 \cdot U(K, L) + U(K+1, L)) \times DX1 \]
\[ AIM = \text{AIM}(J) + (G(L) + 4 \cdot V(K, L) + V(K+1, L)) \times DX1 \]

no \( Z \) calculated here * it will be done when beginning loop 60

\[ \text{ROLD} = \text{REAL}(J) \]
\[ \text{AIOLD} = \text{AIM}(J) \]
\[ \text{RNEW} = R \]
\[ \text{AINEW} = AIM \]

define again \( U(1, L), V(1, L) \) to be used as function to be integrated

defined in the points of axis \( Y (I = 10) \)
\[ U(K, L) = F(J) \]
\[ V(K, L) = G(J) \]

if \((J-J1) \neq 99, 59, 68 \)
\[ L = 1 \]
\[ DO \, 67 \quad I = 11, 12 \]
\[ \text{FACT} = 1. \]
\[ IF(I-I1) \neq 99, 62, 61 \]

integration along the axis defined by \( J = J1 \)

\[ K = 1 \]
\[ R = \text{ROLD} \]
\[ AIM = \text{AIOLD} \]
\[ R = R + (U(K, L) + 4 \cdot U(K+1, L) + U(K+2, L)) \times DX \]
\[ AIM = AIM + (V(K, L) + 4 \cdot V(K+1, L) + V(K+2, L)) \times DX \]

interpolation for smoothing the integral along the axis defined by \( J = J1 \)
\[ RR = (\text{ROLD} + \text{RNEW} + \text{RNEW} + R) / 4. \]
\[ AA = (\text{AIOLD} + \text{AINEW} + \text{AINEW} + AIM) / 4. \]
\[ Z(I, J1) = \sqrt{(RR \times RR + AA \times AA)} \]
\[ \text{ROLD} = \text{RNEW} \]
\[ \text{AIOLD} = \text{AINEW} \]
\[ \text{RNEW} = R \]
\[ \text{AINEW} = AIM \]

\[ K = 1 \]
\[ IF(I-I1) \neq 66, 63, 64 \]
\[ Z(I, J0) = 2. \]
\[ U(K, L-1) = 0 \]
\[ V(K, L-1) = 0 \]
\[ \text{GO TO 67} \]

if \((I-I1) \neq 66, 65, 655 \)
\[ Z(I, J0) = 2.66 \]
\[ U(K, L-1) = 0 \]
\[ V(K, L-1) = 0 \]
\[ \text{GO TO 67} \]

integration for the values of \( X \) axis

\[ \text{FACT} = -1. \]
\[ I(1) = X(K) \]
\[ I(1) = X(K) \]
\[ I(2) = Y(I, L-1) \]
CALL CXSUB(ONE,Z2,FIRST)
CALL CXMPY(AK,Z2,A2)
CALL CXSUB(ONE,A2,SECON)
CALL CXMPY(FIRST,SECON,S)
CALL CXRPW(S,DEN,HALF)

VERM=-ST0(2)
XR=R+(V(K,L)+4.*V(K,L-1)+VERM)*DY1
XAIM=AIM-(U(K,L)+4.*U(K,L-1)+TERM)*DY1

V(K,L-1)=VERM
U(K,L-1)=TERM

CONTINUE

70 CONTINUE

WRITE(6,73)

73 FORMAT(1H0,20X,'*'* ARRAYS U AND V CONTAIN THE REAL AND THE IMMA-
INARY PART OF THE FUNCTION TO BE INTEGRATED *'*')

WRITE NTAP

K=M
II=IX
DO 75  I=1,IX
X(II)= X(K)
K=K-1
II=II-1
L=N
JJ=IY
DO 76  J=1,IY
Y(JJ)=Y(L)
L=L-1
JJ=JJ-1
WRITE(6,77)
77 FORMAT(1HO, 'MATRIX U')
    DO 78 K=1,M
    WRITE(6,93) K, (U(K,L), L=1,N)
78 CONTINUE
    WRITE(6,79)
79 FORMAT(1HO, 'MATRIX V')
    DO 80 K=1,M
    WRITE(6,93) K, (V(K,L), L=1,N)
80 CONTINUE
    WRITE(NTAPE,81) (X(I), I=1,IX)
    WRITE(NTAPE,81) (Y(J), J=1,IY)
81 FORMAT(8E10.3)
82 FORMAT(1HO, 'VECTOR X'/(4X,14E10.3))
    WRITE(6,83) (X(I), I=1,IX)
83 FORMAT(1HO, 'VECTOR Y'/(4X,14E10.3))
    WRITE(6,91)
91 FORMAT(1HO, 'MATRIX Z')
    DO 95 I=1,IX
    WRITE(NTAPE,81) (Z(I,J), J=1,IY)
    WRITE(6,93) (Z(I,J), J=1,IY)
93 FORMAT(14/(4X,14E10.3))
95 CONTINUE
    END FILE NTAPE
    REWIND NTAPE
99 CALL EXIT
END

FEATURES SUPPORTED
NONPROCESS *
ONE WORD INTEGERS ' IOCS
CORE REQUIREMENTS FOR ELLIN COMMON 0 INSKEL COMMON 0 VARIABLES 9850 PROGRAM 2552

END OF COMPI LATION
// JOB x x x
*STOREDAD 2 FX2 TRIZ 30 9600
*STOREDAD 2 FX2 TRIXY 1 320
*DUMPET
// JOB x x x
// XEQ USBES
*FILES(12,TRIZ,2),(11,TRIXY,2)
*CCEND
  0.5  1.  20.  24.  0.  0.5  0.01
// JOB x x x
// XEQ TRICE
*FILES(12,TRIZ,2),(11,TRIXY,2)
*LOCALTRIFN,TRIVJ,XRYR
*CCEND
  33.  33.  5.  15.  10.  80.  4
  60.  3  47  20  90.  100.  
**0
  24.  33.  5.  15.  10.  1  80.  2
  90.  3  47  20  90.  1
// JOB x x x
// XEQ TRI
*FILES(12,TRIZ,2),(11,TRIXY,2)
*CCEND
  10.  7.5  .345 .345  1.  34  34  3  1  4
  120.  .0.  52.  8.7  60.  52.  8.7
// JOB x x x
// XEQ TRICE
*FILES(12,TRIZ,2),(11,TRIXY,2)
*LOCALTRIFN,TRIVJ,XRYR
*CCEND
  -20. -20.  7.5  15.  10.  20.  4
  35.  55.  90.  **1
// JOB x x x
// XEQ ELLIN
*CCEND
// JOB x x x
// XEQ TRICE
*FILES(12,TRIZ,2),(11,TRIXY,2)
*LOCALTRIFN,TRIVJ,XRYR
*CCEND
  25.  35.  4.  90.  10.  100.  4
  45.  45.  2.  **3
  2  28  28  9  -1
Acknowledgements

The authors thank Mr. Horstmann for his interesting discussions which made it possible to perform this job.

References


NOTICE TO THE READER

All scientific and technical reports are announced, as and when they are issued, in the monthly periodical "euro abstracts", edited by the Centre for Information and Documentation (CID). For subscription (1 year : US$ 16.40, £ 6.17, BF 820) or free specimen copies please write to:

Handelsblatt GmbH
"euro abstracts"
Postfach 112
D 4 Düsseldorf 1 (Deutschland)

or

Office de vente des publications officielles des Communautés européennes
37, rue Glesener
Luxembourg

To disseminate knowledge is to disseminate prosperity — I mean general prosperity and not individual riches — and with prosperity disappears the greater part of the evil which is our heritage from darker times.

Alfred Nobel
SALES OFFICES

All reports published by the Commission of the European Communities are on sale at the offices listed below, at the prices given on the back of the front cover. When ordering, specify clearly the EUR number and the title of the report which are shown on the front cover.

SALES OFFICE FOR OFFICIAL PUBLICATIONS OF THE EUROPEAN COMMUNITIES
37, rue Giesener, Luxembourg (Compte chèque postal N° 191-90)

BELGIQUE — BELGIE
MONITEUR BELGE
Rue de Louvain, 40-42 - 1000 Bruxelles
BELGISCH STAATSBLAD
Leuvenseweg 40-42 - 1000 Brussel

DEUTSCHLAND
BUNDESANZEIGER
Postfach - 5000 Köln 1

FRANCE
SERVICE DE VENTE EN FRANCE DES PUBLICATIONS DES COMMUNAUTES EUROPEENNES
26, rue Desaix - 75 Paris 15e

ITALIA
LIBRERIA DELLO STATO
Piazza G. Verdi, 10 - 00198 Roma

LUXEMBOURG
OFFICE DE VENTE DES PUBLICATIONS OFFICIELLES DES COMMUNAUTES EUROPEENNES
37, rue Giesener - Luxembourg

NEDERLAND
STAATSDRUKKERIJ
Christoffel Plantijnstraat - Den Haag

UNITED KINGDOM
H. M. STATIONERY OFFICE
P.O. Box 569 - London S.E.1

Commission of the European Communities
D.G. XIII - C.I.D.
29, rue Aldringer
L u x e m b o u r g