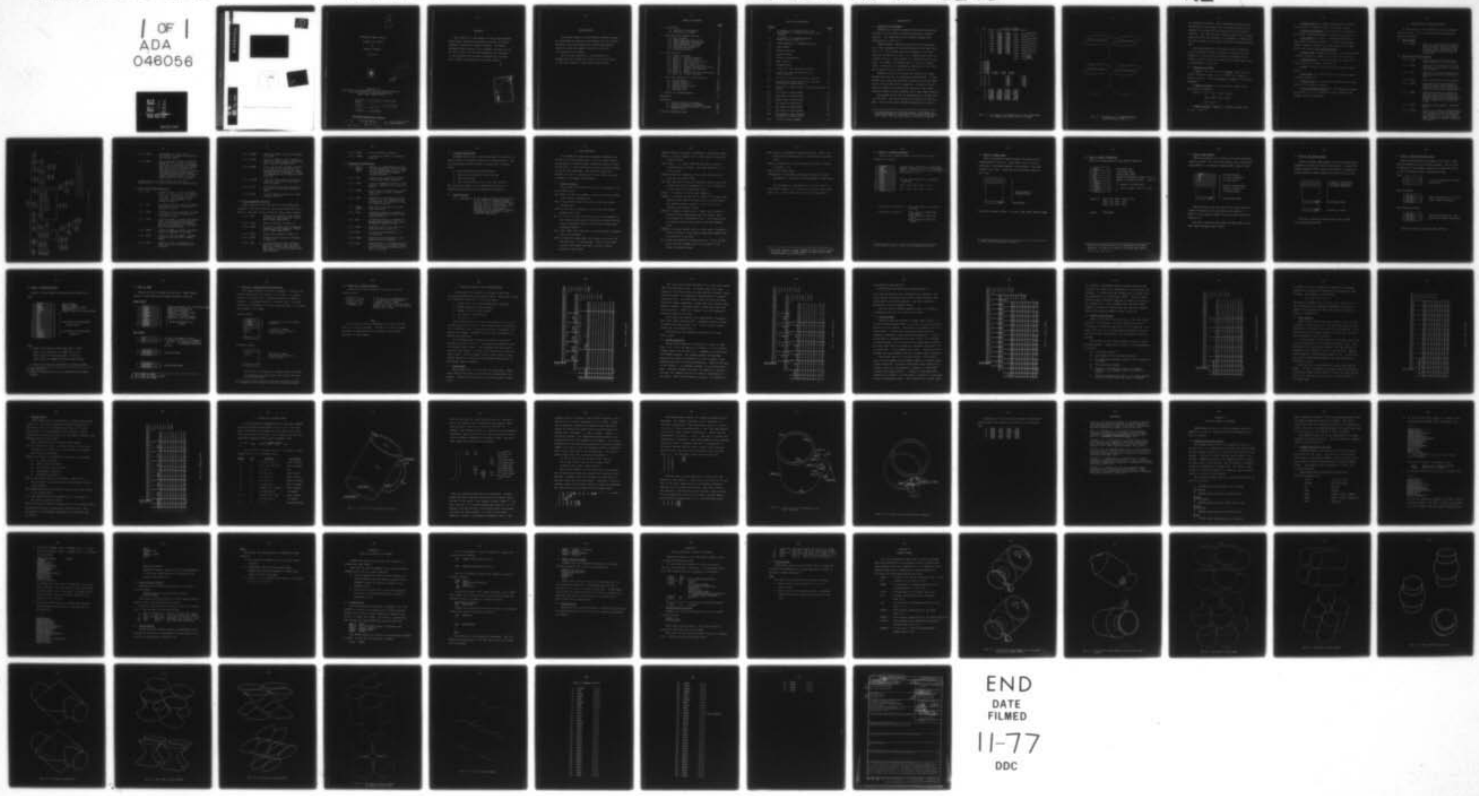
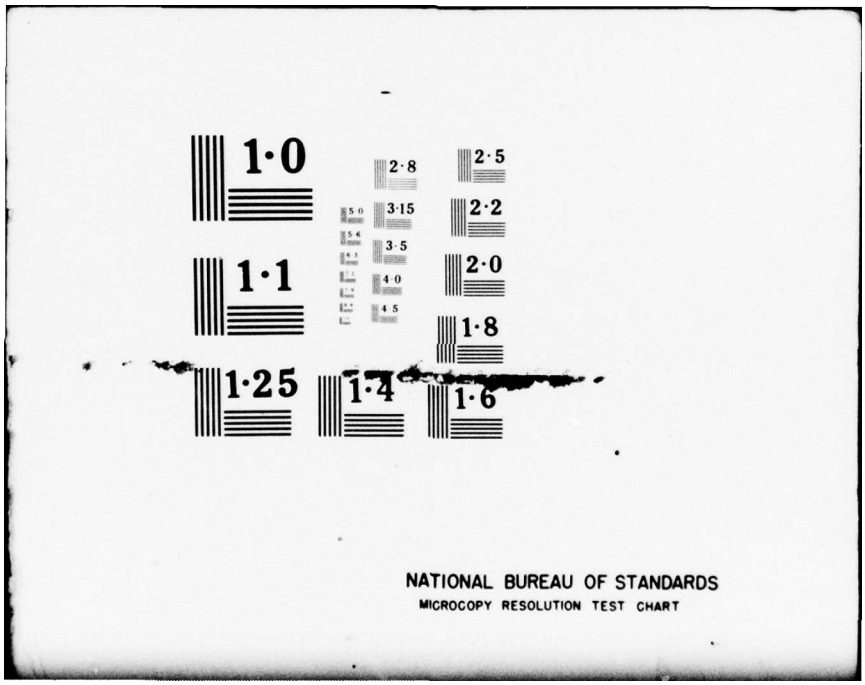
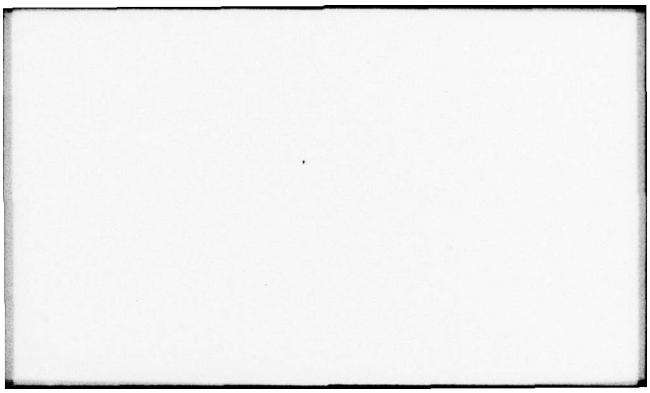
AD-A046 056 RENSSELAER POLYTECHNIC INST TROY N Y COMPUTER RESEAR--ETC F/G 9/2
QUADRAW USER MANUAL,(U)
JUN 77 M POTMESIL
UNCLASSIFIED CRL-53 AFOSR-76-2937 AFOSR-TR-77-1243 NL

| OF |
ADA
046056



END
DATE
FILMED
11-77
DDC


1·0
1·1
1·25
1·4
1·6
1·8
2·0
2·2
2·5
2·8
3·15
3·5
4·0
4·5
5·0
5·6
NATIONAL BUREAU OF STANDARDS
MICROCOPY RESOLUTION TEST CHART



(4)

Technical Report CRL-53

QUADRAW User Manual

by

Michael Potmesil

June 1977

Prepared for
Directorate of Mathematical and Information Sciences
Air Force Office of Scientific Research
under Grant AFOSR 76-2937

Rensselaer Polytechnic Institute

TROY, NEW YORK 12181

(See 1473)

ABSTRACT

This report is a user manual for the program QUADRAW which draws visible-line-projection pictures of objects bounded by sections of quadric surfaces. The report describes three versions of the program, one each for the CDC 6600, Univac 1110 and IBM System/360-370 computers. It also provides the necessary documentation to permit one to modify the program for any other computer.

ACKNOWLEDGMENT

The author wishes to thank Professor Herbert Freeman for providing the direction and guidance for this project. The author also acknowledges the contribution made by Joshua Zev Levin who added perspective projection and documented the program.

The work was supported by the Directorate of Mathematical and Information Sciences, Air Force Office of Scientific Research, under grant AFOSR 76-2937.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

1. INTRODUCTION

1.1 Outline of the Program

QUADRAW is a computer program which draws visible-line pictures of solid objects bounded by sections of quadric surfaces. The algorithm and the original program were developed by P. Woon [1,2].

QUADRAW reads a numerical description of a scene and one or more vantage-point specifications from an input file [Fig. 1.1]. The program produces a drawing of the scene from each vantage point [Fig. 1.2]. Drawings are output on a digital plotter. Each drawing of a scene may show (1) all lines, (2) hidden lines dashed, and (3) only the visible lines. Perspective or orthographic projection of the scene may be selected at the option of the user.

QUADRAW is written in Fortran and consists of a main program and more than 70 subroutines and functions. At the present time, there are available versions for the CDC 6600, Univac 1110 and IBM 360/370 computers. It should be quite simple to adapt one of these three versions to any other computer having a Fortran compiler and a large memory space.*

The QUADRAW source text is maintained as a library file on a CDC 6600. This file contains all versions of the program. The CDC text editor UPDATE can generate the text of

---

* Including system and plotting support, the program requires 150k words on a CDC 6600, 18500/38000 (I/D) words on a Univac 1110 and 250k bytes on an IBM 360 computer.

```
1      20.0                                  STACK OF 8 CUT SPHERES
                    1.0E-03    1.0E-04    1.0E-05    1.0E-04    1.0E-02
2
     1     1        1.000      1.000      1.000                            AT(+4,+4,+4)
                                -8.000     -8.000     -8.000     32.000
     2     1        1.000      1.000      1.000                            AT(+4,+4,-4)
                                -8.000     -8.000      8.000     32.000
     3     1        1.000      1.000      1.000                            AT(+4,-4,+4)
                                -8.000      8.000     -8.000     32.000
     4     1        1.000      1.000      1.000                            AT(+4,-4,-4)
                                -8.000      8.000      8.000     32.000
     5     1        1.000      1.000      1.000                            AT(-4,+4,+4)
                                 8.000     -8.000     -8.000     32.000
     6     1        1.000      1.000      1.000                            AT(-4,+4,-4)
                                 8.000     -8.000      8.000     32.000
     7     1        1.000      1.000      1.000                            AT(-4,-4,+4)
                                 8.000      8.000     -8.000     32.000
     8     1        1.000      1.000      1.000                            AT(-4,-4,-4)
                                 8.000      8.000      8.000     32.000
     9     1                                                               PLANE
                                                       1.000     -6.000    AT Z=6
    10     1                                                               PLANE
                                                       1.000      2.000    AT Z=-2
3
     1    -9 9999
     2   -10 9999
     3    -9 9999
     4   -10 9999
     5    -9 9999
     6   -10 9999
     7    -9 9999
     8   -10 9999
     9    -1 9998    -3 9998    -5 9998    -7 9999
    10    -2 9998    -4 9998    -6 9998    -8 9999
4
     1     9     2                          1.000                -4.000
     2    10     2                          1.000                -4.000
     3     9     2                          1.000                 4.000
     4    10     2                          1.000                 4.000
     5     9     2                          1.000                -4.000
     6    10     2                          1.000                -4.000
     7     9     2                          1.000                 4.000
     8    10     2                          1.000                 4.000
5
6
          3     20.000      0.000     90.000      0.000
          3     20.000      0.000     70.000     30.000
          3     20.000    -55.000     25.000     75.000
          3     20.000     10.000    -75.000    -55.000
          3     20.000     10.000     10.000      0.000
          3     20.000    -60.500    -10.000      5.000
          3     20.000     40.000    -50.000     22.500
          3     20.000     48.000    -22.500    -90.000
9
```

Fig. 1.1 An example of a QUADRAW input file (Scene made of 8 spheres which are cut by 2 planes)

Fig. 1.2 An example of a QUADRAW drawing
(Scene listed in Fig. 1.1)

any implemented version. This arrangement simplifies the maintenance of the program since any modifications of the program will automatically be applied to all implemented versions. Note that the three versions differ only in small parts which are very unlikely to be changed. A version for any other computer can be easily added to this master source text.

Although QUADRAW can also handle objects bounded entirely by planar surfaces, a different algorithm [3] and program [4], specifically developed for planar-surfaced objects, would be faster and require less memory space, and is recommended in that case.

1.2 Definition of Terms

A scene is set of one or more bodies. Each body is a closed, connected space in 3D, delimited by a number of quadric surfaces. The terms object and body are used interchangeably.

A quadric surface is the locus of all points in 3D satisfying the equation:

$$Q(x,y,z) = a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5xz + a_6yz + a_7x + a_8y + a_9z + a_{10} = 0$$

A planar surface or plane is a quadric surface with $a_1, a_2, \ldots a_6 = 0$.

A surface bound is a quadric surface used to delimit the sections of a quadric surface that bound a body.

A surface intersection is the locus of all points of two quadric surfaces satisfying $Q_1(x,y,z) = Q_2(x,y,z) = 0$.

A real vertex is a point in 3-space where three or more quadric surfaces intersect. If these three or more quadric surfaces intersect in several points, then these points are called a set of real vertices.

A virtual edge or limb is the locus of all points where a quadric surface is tangent to the lines of projection.

A virtual vertex is the intersection in 3-space of a virtual edge and a surface intersection.

An edge is either a virtual edge or a surface intersection.

A polar plane is a plane in which all virtual edges of a quadric surface lie.

A cutting plane is a plane which intersects the intersection of two quadric surfaces.

An object coordinate system is a 3D cartesian coordinate system, x" y" z", in which all object equations and vantage points are specified.

2. DESCRIPTIONS OF QUADRAW ROUTINES

In this section a brief description of each QUADRAW routine is given. Fig. 2.1 shows the interconnections among the routines.

2.1 Main Program

2.1.1 QDRAW - controls the execution of the program. It calls the main functional subroutines, opens and closes plotting output and measures execution time of the main steps of the program.

2.2 Main Functional Subroutines

2.2.1 INPOB - reads scene specifications and creates Tables 1-5 for the scene.

2.2.2 RVERT - computes the locations of the real vertices of the objects and stores them in Table 4.

2.2.3 SINT - computes all surface intersections, point by point and stores them in Table 6.

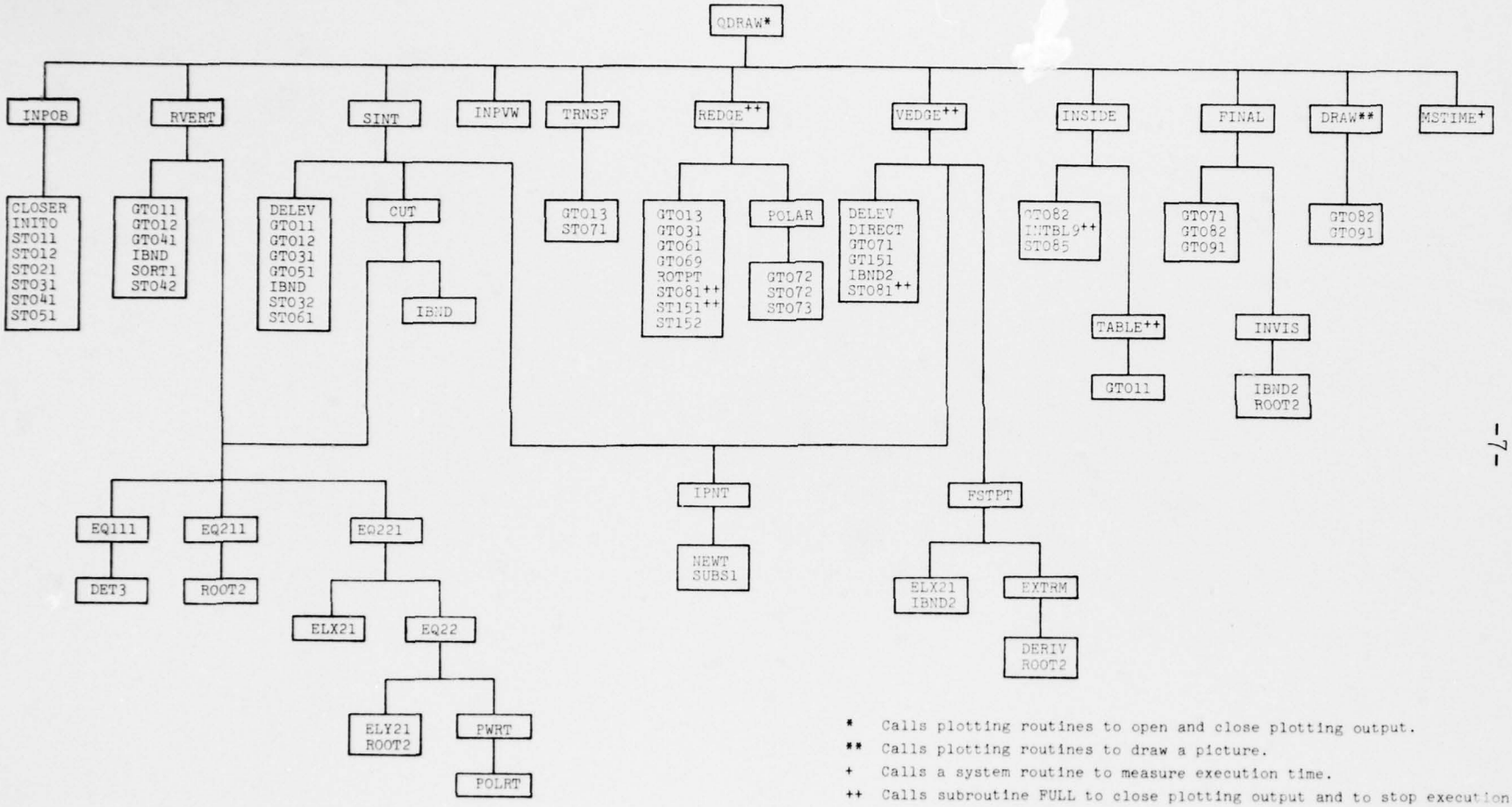
2.2.4 INPVW - reads vantage point specifications.

2.2.5 TRNSF - transforms the surface equations from the object specification coordinate system to the coordinate system of the current vantage point.

2.2.6 REDGE - transforms surface intersections from the object specification coordinate system to the coordinate system of the current vantage point. It also computes all virtual vertices.

2.2.7 VEDGE - computes virtual edges - limb segments.

2.2.8 INSIDE - sets up the surface minimum-maximum table (Table 9) which is later used for envelope tests in determining edge visibility. It also inserts edge minimum and maximum coordinates in Table 8.



* Calls plotting routines to open and close plotting output.

** Calls plotting routines to draw a picture.

\+ Calls a system routine to measure execution time.

++ Calls subroutine FULL to close plotting output and to stop execution.

Diagnostic calls to WDUMP are not shown.

Fig. 2.1 Interconnections of QUADRAW routines

2.2.9 FINAL - determines the visibility of all points of surface intersections and virtual edges.

2.2.10 DRAW - calls plotting routine to draw the projection of the scene; the scale factor of the drawing is determined from Table 9 and the point coordinates are obtained from Table 8. This subroutine may draw (1) visible intersections and virtual edges, (2) hidden intersections and virtual edges as dashed lines, or (3) all intersections and virtual edges.

Subroutines 2.2.1-3 are called once per scene.
Subroutines 2.2.4-10 are called once per vantage point.

2.3 Other Functional Subroutines

2.3.1 CUT - called by SINT to find a 'cutting point' if there are no real vertices on a real intersection. The equation of a 'cutting plane' which intersects the intersection is stored in Table 3.

2.3.2 IBND - an integer function called to determine whether a point is within a surface's bounds.

2.3.3 IBND2 - performs the same operations as IBND on surfaces in the vantage-point coordinate system.

2.3.4 ROTPT - called by REDGE to rotate (and translate, for perspective projections) each surface intersection point to get its coordinates in the vantage-point coordinate system.

2.3.5 DIRECT - called by VEDGE to adjust the direction of propagation of an edge.

2.3.6 DELEV - called by SINT and VEDGE to delete a pair of vertices from a vertex table.

2.3.7 FSTPT - called by VEDGE to determine the first point on a limb that has no vertices.

2.3.8 EXTRM - called by FSTPT to find the extrema of a limb.

2.3.9 INVIS - called by FINAL to check whether a face of a surface hides a particular point from the vantage point.

2.3.10 WDUMP - lists the current contents of Tables 1 - 9 and 15. It is called from many subroutines whenever an error is detected by the program in either the description of an object or computations.

2.3.11 CLOSER - lists all input cards that were not read after an error in input was found by INPOB.

2.3.12 INITO - initializes all global variables to zero before a new scene is read by INPOB.

2.3.13 SORT1 - sorts an array of data into a descending sequence.

2.4 Table Management Routines

These routines access the tables of the QUADRAW data structure. They are called by the routines described in 2.2 and 2.3. The small letters stand for single digit numbers.

2.4.1 GTmn - is the n'th routine for getting the contents of table number m. There are 12 GTmn subroutines.

2.4.2 GT151 - gets the contents of Table 15.

2.4.3 STOmn - is the n'th routine for storing the contents of table number m. There are 14 STOmn subroutines.

2.4.4 ST151 - stores the contents of Table 15.

2.4.5 ST152 - removes the last entry in the list in Table 15.

2.4.6 FULL - is called when the data structure runs out of space, that is, when Table 8 or Table 9 hits Table 15 or when Table 1 - 9 hits the end of the data structure.

2.4.7 TABLE - stores elements of Table 9.

2.4.8 INTBL9 - is called by INSIDE to initialize Table 9.

2.5 Mathematical Subroutines

2.5.1 EQ111, EQ211 & EQ221 - compute the intersection point(s) of three equations, with '2' standing for second-order equation and '1' standing for first-order (linear) equation.

2.5.2 EQ22 - computes the intersection points of two second-order equations.

2.5.3 ROOT2 - solves a quadratic equation of one variable, indicating the number of roots.

2.5.4 SUBS1 - substitutes a variable in an equation to make it into a simpler equation.

2.5.5 NEWT - computes the intersection of two curves (i.e., the intersection of two surfaces in a plane) by using the Newton's method.

2.5.6 PWRT & POLRT - help EQ22 in solving two second-order equations.

2.5.7 ELX21 - eliminates variable x between a second-order equation and a first-order equation.

2.5.8 ELY2 - eliminates variable y between two second-order equations of y and z. Sylvester's method is used.

2.5.9 DERIV - computes $\partial^2 y/\partial z^2$ or $\partial^2 z/\partial y^2$ of a conic section at point y,z.

2.5.10 DET3 - evaluates the determinant of a 3 x 3 matrix.

2.5.11 POLAR - computes the coefficients of the polar plane of a surface.

2.5.12 IPNT - computes points on intersections of two surfaces. It is used to trace points on surface intersections as well as on limbs.

2.6 Plotting Subroutines

QUADRAW assumes that a plotting package is available to draw pictures on a digital plotter or a similar device. The plotting package should be able to perform the following tasks:

a) open and close the plotting output

b) position the plotter to a new picture

c) draw solid lines

d) move the pen while it is in the "up" position

The plotting packages for the implemented versions of QUADRAW are described in their respective appendices.

2.7 Timing Routine

2.7.1 MSTIME(MS)- is an integer function which returns as its value the elapsed program execution time in milliseconds since time MS (also in milliseconds). This function calls a system routine to measure the time. The system routines for the implemented versions of QUADRAW are described in their respective appendices.

3. DATA STRUCTURE

All QUADRAW data structure and global variables are stored in the blank COMMON block. The ten tables used by the program are stored in array IPOOL(32000) which is also EQUIVALENCEd to POOL(32000) and therefore contains integer as well as real variables. The tables in IPOOL are numbered 1-9 and 15. Tables 10-14 do not exist but could be added if the program were extended.

3.1 Global Variables

The following variables and arrays are contained in the blank COMMON block:

PHI, THETA, PSI are the azimuth, elevation and twist angles of the current vantage point.

PSCL is the picture scale of the scene from the current vantage point.

IPTYP is the type of drawing to be made from the current vantage point view.

R1, R2, R3, R4, R5, R6, R7, R8, R9 are the nine elements in the rotation matrix which transforms an object from the object coordinate system to the current vantage point coordinate system.

EPS1, EPS2, EPS3, EPS4, EPS5 are the computational tolerances used in the program.

IHEAD contains an integer digit from column one of the last data card read by the program. Digits in the first column of data cards are used to delimit various sections of data input.

NSURFS contains the number of surfaces in the current scene.

TAPE is a logical variable set to .TRUE. when the plotting output is open.

IPB1, IPB2, ..., IPB14, IPB15 are pointers to the 15 possible tables in IPOOL. Note that Tables 10 - 14 do not exist.

MAXDM contains the dimension of the array IPOOL which is 32000 at the present time.

YMN, YMX, ZMN, ZMX are used by subroutine DRAW to store the maximum and minimum coordinates in y and z directions of the scene in the projection plane.

SPARE(7) is an array of seven spare variables.

LISTER is a logical variable set to .TRUE. if the input data specifies that tables in IPOOL are to be listed after the current scene is entered or after the current vantage point drawing is made.

ABORT is a logical variable set to .TRUE. when an error is detected during computations and the current object or vantage point computations fail. The program tries to continue with the next scene or the next vantage point.

DUMPER is a logical variable set to .TRUE. when a listing of tables in IPOOL is to be made by subroutine WDUMP after the detection of an error.

DE is used for perspective projections only. DE is the distance from the vantage point to the origin of the object coordinate system.

DEE is used for perspective projections only. DEE is the distance from vantage point to the picture projection plane.

IPTNA is a pointer to the last location in IPOOL occupied by Tables 1 - 9.

OBSCL is the scene scale.

IPOOL(32000) is the array containing the tables of scene description. This array is EQUIVALENCEd to POOL(32000).

The following is a description of all ten tables in IPOOL-POOL.* Tables 1 to 6 are set up once per scene, while tables 7 to 9 and 15 are set up once per vantage point.

---

* The array contains integer numbers as well as real numbers (floating point). We mark integer entries with I's and floating point entries with R's.

3.2 Table 1: Surface Equations

There is one 14-word element in this table for each surface of an object.

1	ISTYPE	I	surface type: 1=component, 2=auxiliary
2	IDEG	I	degree of equation: 1=linear, 2=quadric
3	LINK	I	pointer to the surface's bounds in Table 2*
4	FACTOR	R	scale factor = 1.0
5	A(1)	R	
6	A(2)	R	
7	A(3)	R	
8	A(4)	R	the ten coefficients of the surface
9	A(5)	R	equation:
10	A(6)	R	
11	A(7)	R	$a_1x^2 + a_2y^2 + a_3z^2 + a_4xy + a_5xz +$
12	A(8)	R	
13	A(9)	R	$a_6yz + a_7x + a_8y + a_9z + a_{10} = 0$
14	A(10)	R	

This table is written by: STO11 and STO12, both called by INPOB.

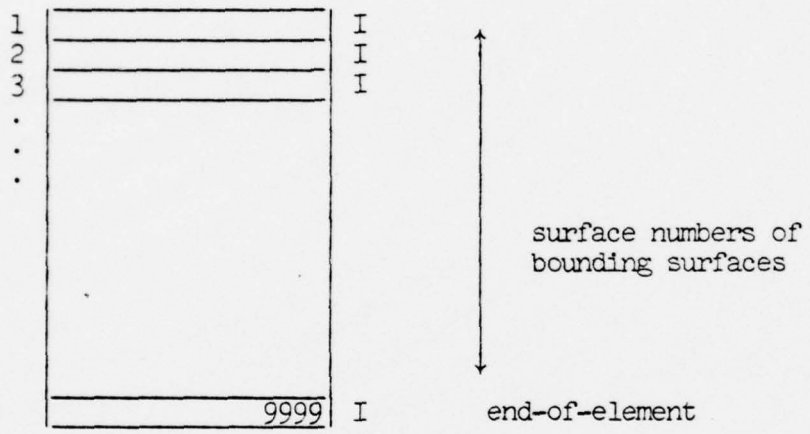
This table is read by: GTO11 called by RVERT, SINT and TABLE,
GTO12 called by RVERT and SINT,
GTO13 called by TRNSF and REDGE.

---

* The pointer is set to zero for self-bounding surfaces.

3.3 Table 2: Surface Bounds

There is one element of variable length in this table for each bounded surface* in an object. Pointer LINK in table 1 points to this element. Self-bounded surfaces - such as spheres - do not have elements in this table. Pointer LINK for a self-bounded surface is set to zero.



This table is written by INPOB. It is read by IBND, IBND2, TRNSF and REDGE.

---

* Surface bounds are explained in section 4.2.

3.4 Table 3: Surface Intersections

There is one 9-word element for every surface intersection.

1	INTYP	I	intersection type
2	IS1	I	first surface number
3	IS2	I	second surface number
4	KV	I	pointer to real vertex pointers - Table 5
5	KIP	I	pointer to intersection points - Table 6
6	CUTP(7)	R	equation of cutting plane*:
7	CUTP(8)	R	$cutp_7x + cutp_8y + cutp_9z + cutp_{10}=0$
8	CUTP(9)	R	
9	CUTP(10)	R	

Written by: STO31 from INPOB: words 1-3, 6-9
STO32 from SINT : word 5
STO51 from INPOB: word 4

Read by: SINT, REDGE

---

* Indices of coefficients are set up for compatibility with quadric equations. In section 4.4 we refer to the cutting plane equation as $c_1x + c_2y + c_3z + c_4 = 0$.

3.5 Table 4: Real Vertices

There is an element of up to 19 words in this table for each real vertex in an object. For a set of real vertices, that is vertex multiplicity MULTIP > 1, the element is repeated MULTIP number of times.

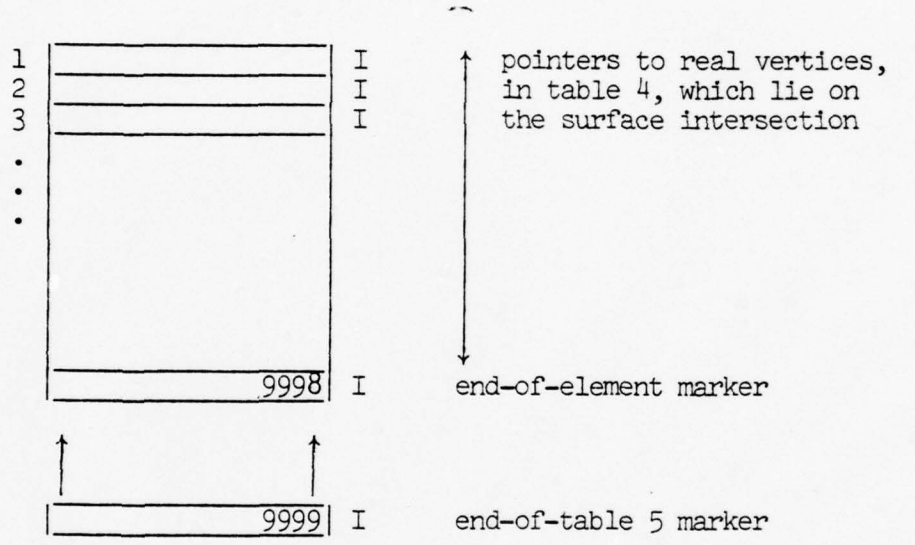
1	MULTIP	I	vertex multiplicity
2	VX	R	x, y and z coordinates
3	VY	R	of a real vertex
4	VZ	R	
5		I	
6		I	
7		I	numbers of surfaces which form this vertex, up to 15 surfaces allowed
	9999	I	end-of-element marker

Vertex multiplicity gives the number of real vertices which are formed by intersections of the same three or more surfaces. If MULTIP > 1, then subsequent elements in the set of real vertices have MULTIP set to 0.

This table is written by STO41 called from INPOB (words 2-4) and STO42 called from RVERT (words 1,5-last).

3.6 Table 5: Real Vertex Pointers

There is a variable length element in this table for each surface intersection which has one or more real vertices on it. Pointer KV in surface intersection Table 2 points to this element. If there is not any real vertex on a surface intersection pointer KV is set to zero.



This table is written by subroutine STO51 called by INPOB. It is read by subroutine SINT.

3.7 Table 6: Surface Intersection Points

There are three types of 3-word elements in this table: points, end-of-line markers and end-of-intersection markers. The first point of intersection of two surfaces is pointed to by pointer KIP in Table 3. This table usually occupies a large amount of space.

Point:

1	X	R
2	Y	R
3	Z	R

x, y and z coordinates of an intersection point.

End-of-line marker:

1	999 998.0	R
2	999 998.0	R
3	999 998.0	R

marker indicating the end of a line within a surface intersection.

End-of-intersection marker:

1	999 999.0	R
2	999 999.0	R
3	999 999.0	R

marker indicating the end of the last line of a surface intersection.

This table is written by subroutine ST061 from SINT.

3.8 Table 7: Transformed Surface

There is a 19-word element in this table for each surface of an object.

1	ISTYP*	I		type of surface
2	IDEG*	I		degree of surface
3	LINK2*	I		pointer to Table 2 (bounds)
4	FACTOR*	R		scale factor = 1.0
5	LINK15	I		pointer to Table 15 (virtual vertices)
6	B(1)	R	↑	
7	B(2)	R	|	
8	B(3)	R	|	
9	B(4)	R	|	
10	B(5)	R	|	
11	B(6)	R	|	coefficients of the transformed
12	B(7)	R	|	surface equation
13	B(8)	R	|	
14	B(9)	R	|	
15	B(10)	R	↓	
16	POLRA(7)	R	↑	coefficients of the polar plane
17	POLRA(8)	R	|	equation **
18	POLRA(9)	R	|	
19	POLRA(10)	R	↓	

Notes:

a) Words 1-15 are written by TRNSF except word 5 - LINK15. LINK15 is set initially to 0 by TRNSF, set to -1 by POLAR if there is not any polar plane. It is set to point to Table 15 by REDGE if there are virtual vertices.

b) Words 16-19 (polar plane coefficients) are written by POLAR.

---

* Copied from Table 1.

** Indices of coefficients are set up for compatibility with quadric equations.

3.9 Table 8: Edges

There are two types of elements into this table. Header elements which are 11 words long and point elements which are 3 words long.

Header Element:

1	ITYP**	I	surface intersection type, 0 for virtual edges
2	ICLAS**	I	edge class: H_1,H_2,H_3 or H_4
3	ISNA**	I	number of first surface
4	IORNTA**	I	orientation: +1=front, -1=back
5	ISNB**	I	number of second surface
6	IORNTB**	I	orientation: +1=front, -1=back
7	LINK8*	I	pointer to the next edge header
8	YMAX***	R	maximum and minimum y and z envelope
9	YMIN***	R	
10	ZMAX***	R	
11	ZMIN***	R	

Point Element:

1	X*	R	x, y and z coordinates of a point If the point is invisible X is changed to: 999 997.0 for orthogonal projection -X for perspective projection by FINAL.
2	Y*	R	
3	Z*	R	

1	999 998.0*	R	end-of-line marker
2	999 998.0*	R	
3	999 998.0*	R	

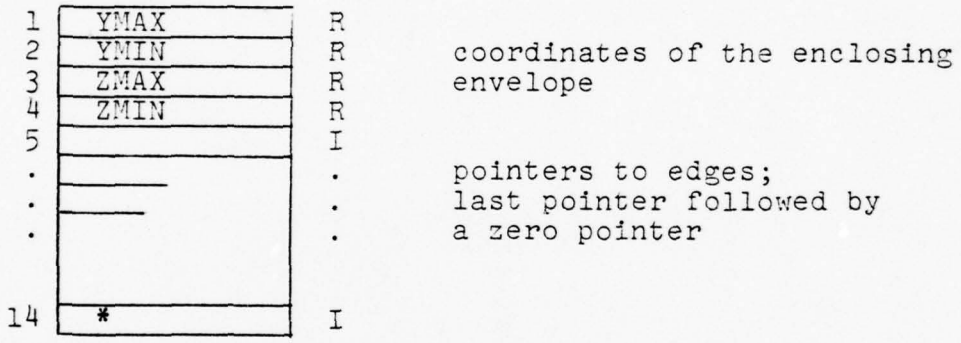
1	999 999.0*	R	end-of-surface marker
2	999 999.0*	R	
3	999 999.0*	R	

* Set by REDGE and VEDGE.
** Set by STO81 from REDGE and VEDGE.
*** Set by STO85 from INSIDE.

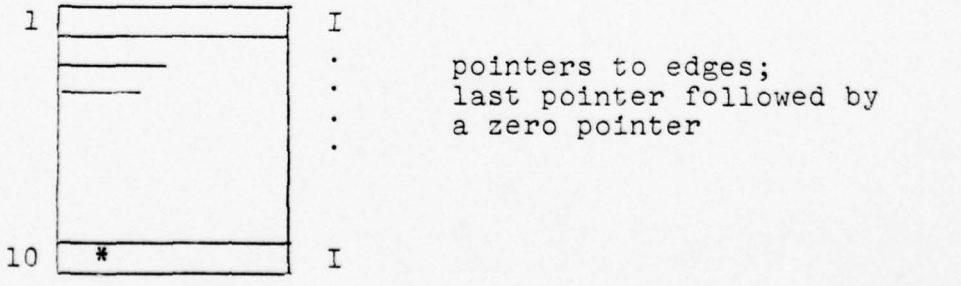
3.10 Table 9: Surface-Enclosing Envelopes

This table contains two types of elements. There is one primary element for each surface of the object. A primary element is 14 words long. Secondary elements are added as required. A secondary element is 10 words long. The table contains the coordinates of the enclosing envelope of a surface and pointers to its edges.

Primary element:



Secondary element:



This table is initialized by INTBL9 called by INSIDE.

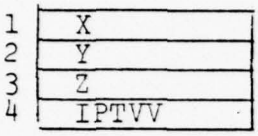
It is written by TABLE called from INSIDE, and read by FINAL and DRAW.

---

* A negative pointer points to the next secondary element.

3.11 Table 15: Virtual Vertices

There is one 4-word element in this table for each virtual vertex.



↕ The x,y and z coordinates of a virtual vertex written by REDGE; read by VEDGE.

Pointer to the next virtual vertex element in a list. Last pointer in list is set to zero.

Note

This table is written backwords at the same time that Table 8 is written forwards. If Table 8 or Table 9 should hit Table 15 subroutine FULL is called to stop further execution of the program.

4. SCENE AND VANTAGE POINT SPECIFICATIONS

In this section we describe the format of data that specifies QUADRAW scenes and vantage points. Each scene entered into QUADRAW is specified by these five parts:

1. Scene scale, title and computational tolerances.
2. Quadric equation of each surface.
3. Bounds of each bounded surface.
4. All surface intersections.
5. All real vertices.

Scene specifications are usually followed by one or more vantage point specifications. Each vantage point is specified by its position, type of projection and type of drawing to be made by the program. QUADRAW can use either orthographic or perspective projection.

The following pages describe the format of scene and vantage point data in detail. 80-column punch cards are used to illustrate data formats. They represent any 80-or-more-character line images. Vertical solid lines divide cards into input fields. Vertical dashed lines in input fields of real numbers specify the position of the assumed decimal point if the decimal point is omitted.

4.1 Scene Header

Scene header [Fig. 4.1] is a set of three cards. These cards contain scene scale, title and five computational tolerances. These three cards must be the first cards of every scene.

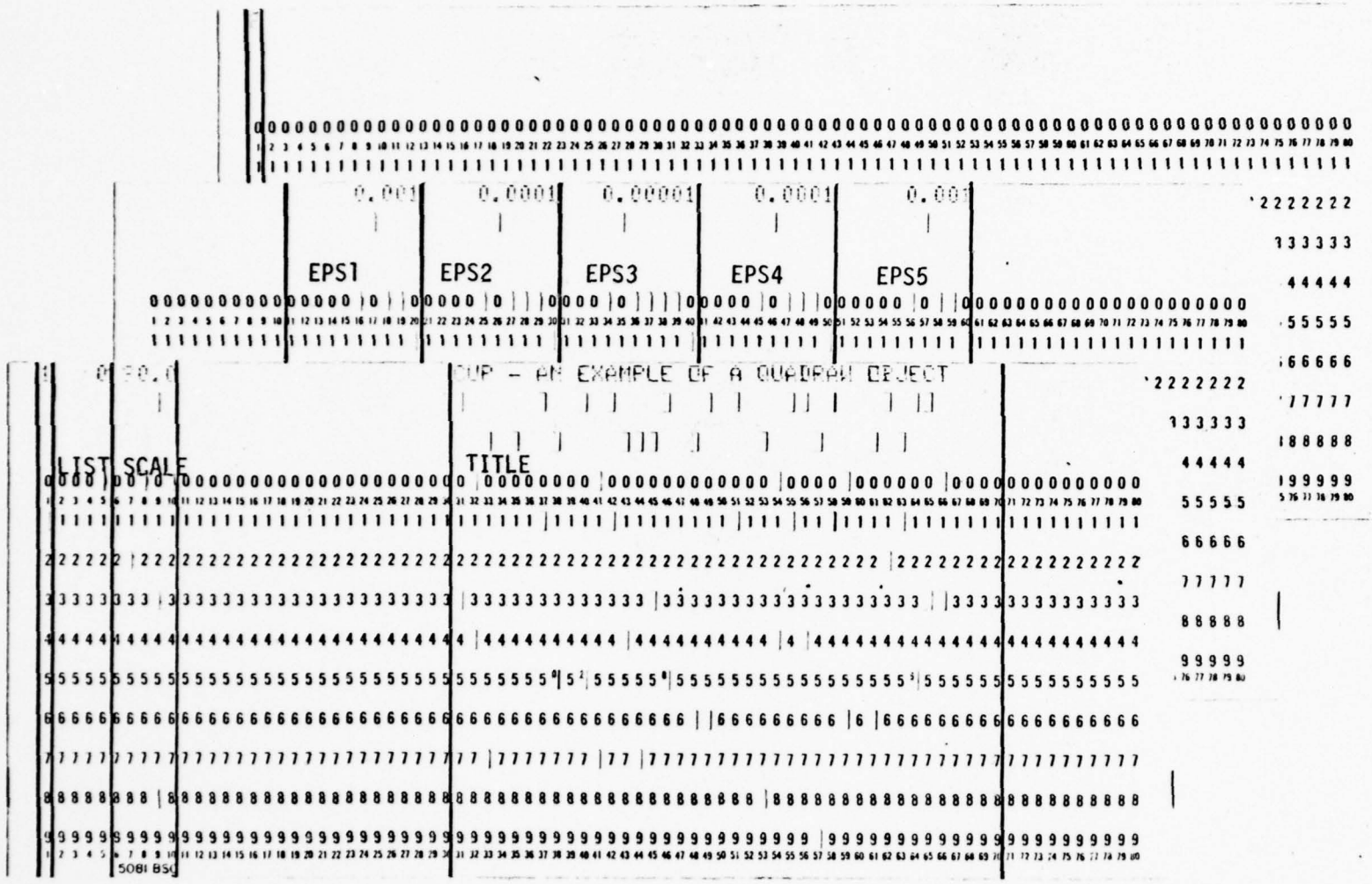


Fig. 4.1 Scene Header

The first card contains delimiter "1" in the first column. LIST is a diagnostics flag. A non-zero value will cause listing of all tables after the scene has been entered. SCALE gives the number of straight-line approximations of quadric curves per inch. Note that small value will give the drawn pictures a coarse appearance. Large value will require large amount of memory. Acceptable value of SCALE has been found to be 15.0-20.0. TITLE is a string of up to 40 characters giving the name of the scene.

The second card contains five computational tolerances: EPS1, EPS2, EPS3, EPS4 and EPS5. Acceptable values of these tolerances have been found to be: 0.001, 0.0001, 0.00001, 0.0001 and 0.001 respectively.

The third card is a delimiter card with "2" in the first column.

4.2 Surface Equations

Every surface equation is specified by a pair of cards [Fig. 4.2]. NS is the number of a surface. It is a positive integer number from the set {1,2,...,NSURFS} for a scene with with NSURFS surfaces. The surface numbers are important because each surface is referred to by its NS number in the remaining sections of scene specifications. ITYPE is the type of the surface: 1 is component surface, 2 is auxiliary surface. Auxiliary surfaces are used, for example, as bounds to eliminate the unwanted sections of cones or hyperboloids of two sheets. They are not drawn in pictures. The equation of

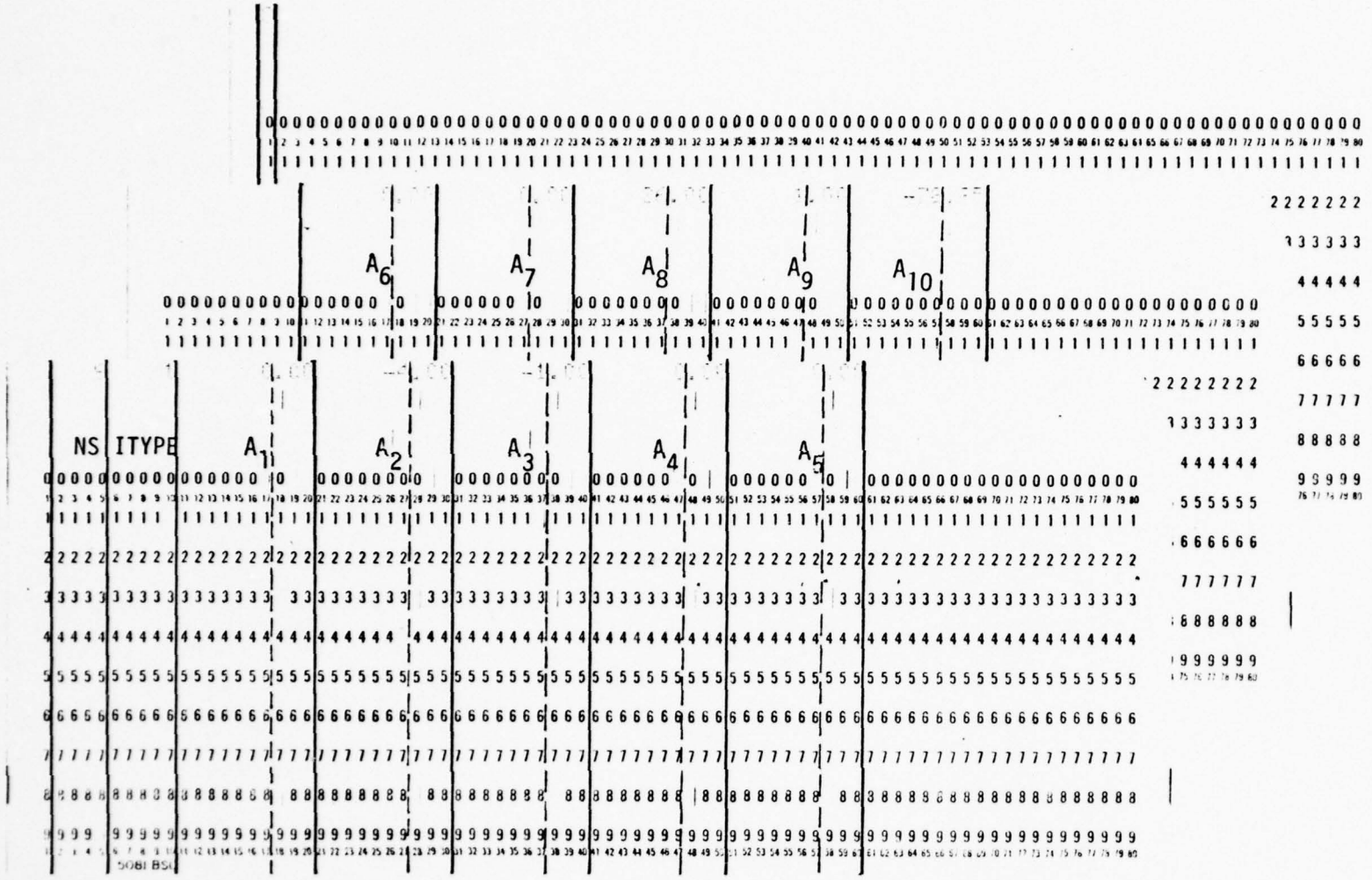


Fig. 4.2 Surface Equations

the surface is specified by:

$$a_1x^2+a_2y^2+a_3z^2+a_4xy+a_5xz+a_6yz+a_7x+a_8y+a_9z+a_{10} = 0$$

Note that the signs of the coefficients are important. Any point outside the surface must yield a positive result when substituted to the above equation. Any point inside the surface must yield a negative result.

The last pair of surface equation cards is followed by a delimiter card with "3" in the first column.

4.3 Surface Bounds

Surface bounds are entered in format shown in Fig. 4.3. For each bounded surface there is at least one card giving the number NS of the surface. The number is followed by up to 15 bound values. If more bound values are given, they are entered on subsequent cards which have NS omitted. A list of bounds must be terminated by the value "9999". If a surface is self-bounding, such as a sphere, its bound card is omitted. The last bound card is followed by a delimiter card with a "4" in column one. If all surfaces in a scene, such as a set of spheres, are self-bounded, only the delimiter card is entered.

Each bound value is a non-zero integer. The absolute value of this integer is the surface number of a surface which bounds surface NS. A positive value means that surface NS is bounded - that is, its existence is limited - to the space outside the bounding surface. A negative value means that surface NS is bounded - its existence is limited - to the space inside the bounding surface. The "outside" and "inside" space

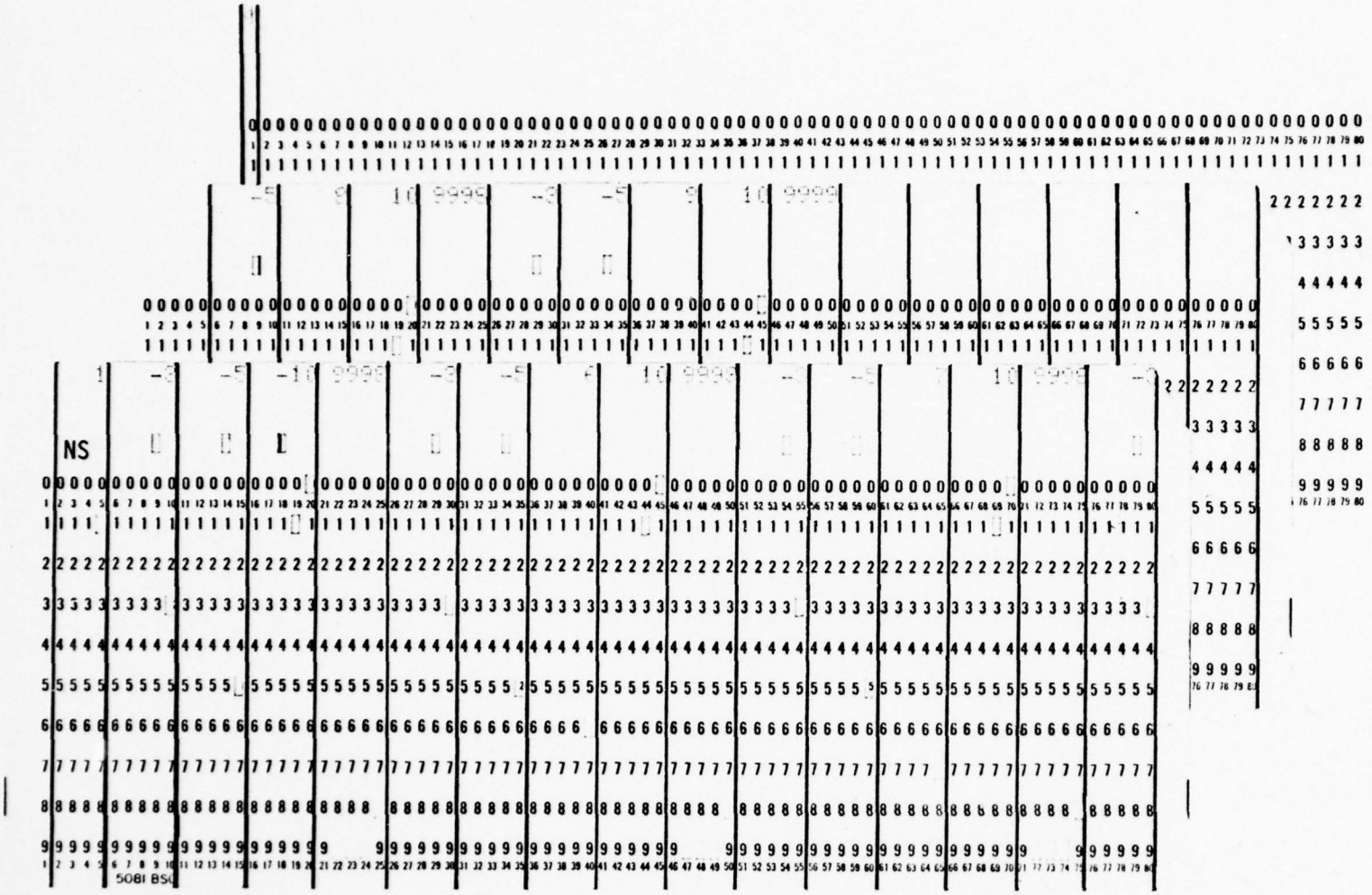


Fig. 4.3 Surface Bounds

of a surface is determined from the surface equation and therefore the signs of the coefficients of the equation are important. A surface may have several sets of bounds which are separated by the bound value "9998". Any point which satisfies all the bounds of any set of bounds is a point on the surface of NS. This indicates that the bound "9998" stands for logical OR operator and that there are implied logical AND operators between bounds in each set.

4.4 Surface Intersections

There is one card as shown in Fig. 4.4 for each intersection of two surfaces. The last intersection card is followed by a delimiter card with a "5" in column one. If a scene, such as a set of non-intersecting spheres, does not have any surface intersection, only the delimiter card is entered.

NS1 and NS2 are the surface numbers of the two intersecting surfaces. ISTYPE gives the type of the intersection as follows:

±1	two planar surfaces
±2	one planar and one non-planar surfaces
±3	two non-planar surfaces, however, the intersection is planar
±4	two non-planar surfaces
>0	obtrusive intersection, that is, the angle between the two surfaces inside the object is convex
<0	intrusive intersection, that is, the angle between the two surfaces inside the object is concave

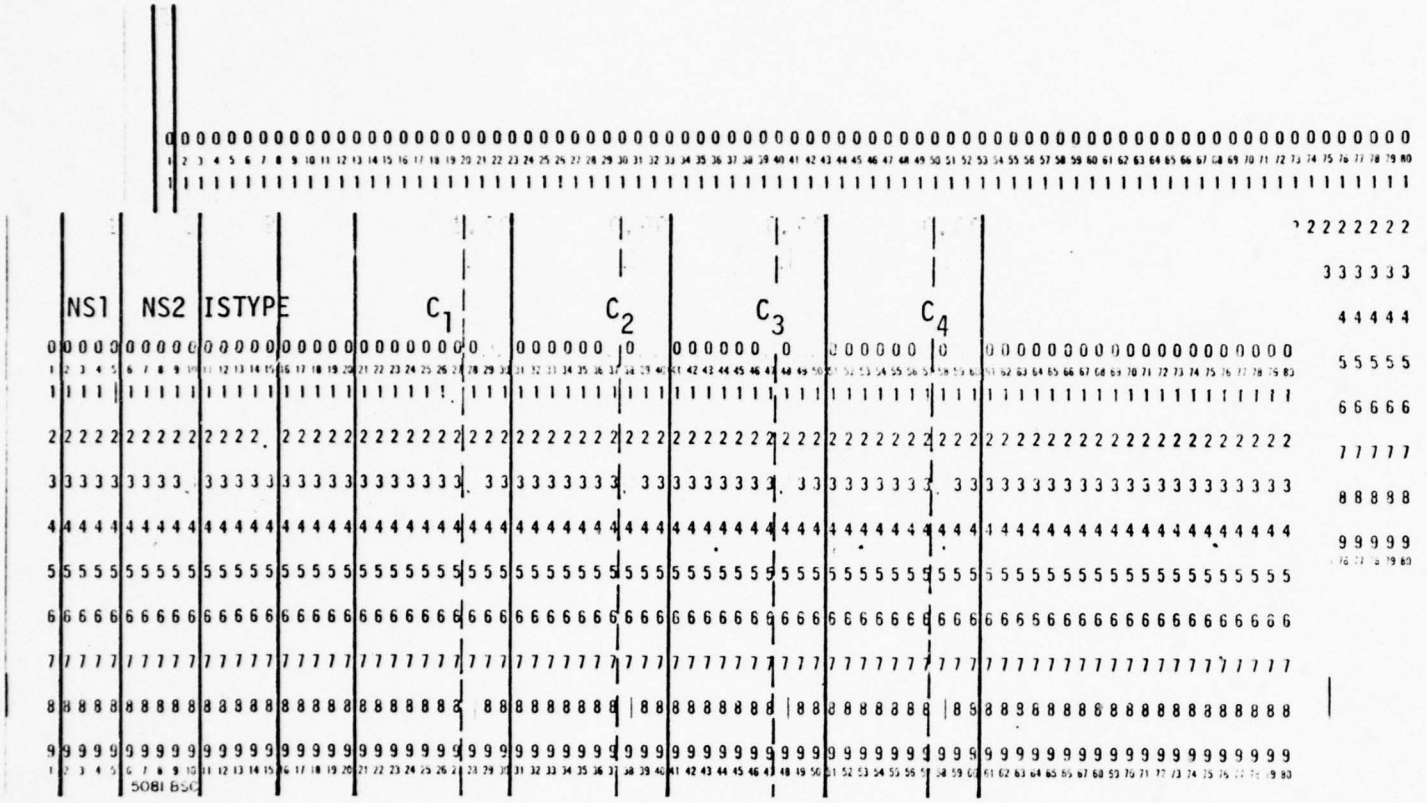


Fig. 4.4 Surface Intersections

If either of the two intersecting surfaces is non-planar, equation of a plane which intersects the intersection must be given. The equation is in the form:

$$c_1x + c_2y + c_3z + c_4 = 0$$

The plane is referred to as a cutting plane because it cuts the intersection and, therefore, determines the point or points on the intersection where the program will begin to trace the intersection.

4.5 Real Vertices

There is one card for each real vertex or a set of real vertices as shown in Fig. 4.5. The last real vertex card is followed by a delimiter card with a "6" in column one. If an object, such as a cylinder, does not have any real vertices, only the delimiter card is entered.

M is the number of real vertices in the object which lie at the intersections of the specificed surfaces $NS_1,...NS_{15}$. There must be at least three surfaces NS_1, NS_2 and NS_3 to determine a real vertex or a set of real vertices. Maximum allowed number of intersecting surfaces is 15. If there are less than 15 intersecting surfaces, the last one must be followed by "9999".

Note that one of the surfaces NS_1, NS_2 or NS_3 must be planar. If none is available, you have to add to the object an auxiliary planar surface which intersects the vertices and specify this surface as one of the first three surfaces on the vertex card.

Fig. 4.5 Real Vertices

4.6 Vantage Points

Each vantage point is specified by a single card as shown in Fig. 4.6. The last vantage point may be followed by an end-of-life mark or by a delimiter card with a "9" in the first column. If the delimiter card is entered it may be followed by an end-of-file mark or by the cards of another scene beginning with the title card.

Each vantage point card contains these fields:

LIST is a diagnostics flag. A non-zero value will cause the contents of all tables to be printed after the vantage point has been processed.

IPTYPE is the type of the drawing to be made by the program:

- ±1 all lines are drawn
- ±2 hidden lines are dashed
- ±3 only visible lines are drawn
- >0 orthographic projection
- <0 perspective projection

PSCL is the picture scale of the drawing. Ratio of the picture scale to the scene scale will determine the size of the drawing.

PHI, THETA and PSI are the azimuth, elevation and twist angles of the vantage point.

DE is the distance from the vantage point to the origin of the object coordinate system.

DEE is the distance from the vantage point to the picture plane.

DE and DEE are used for perspective projection only. They are ignored for orthographic projection and, therefore, may be omitted.

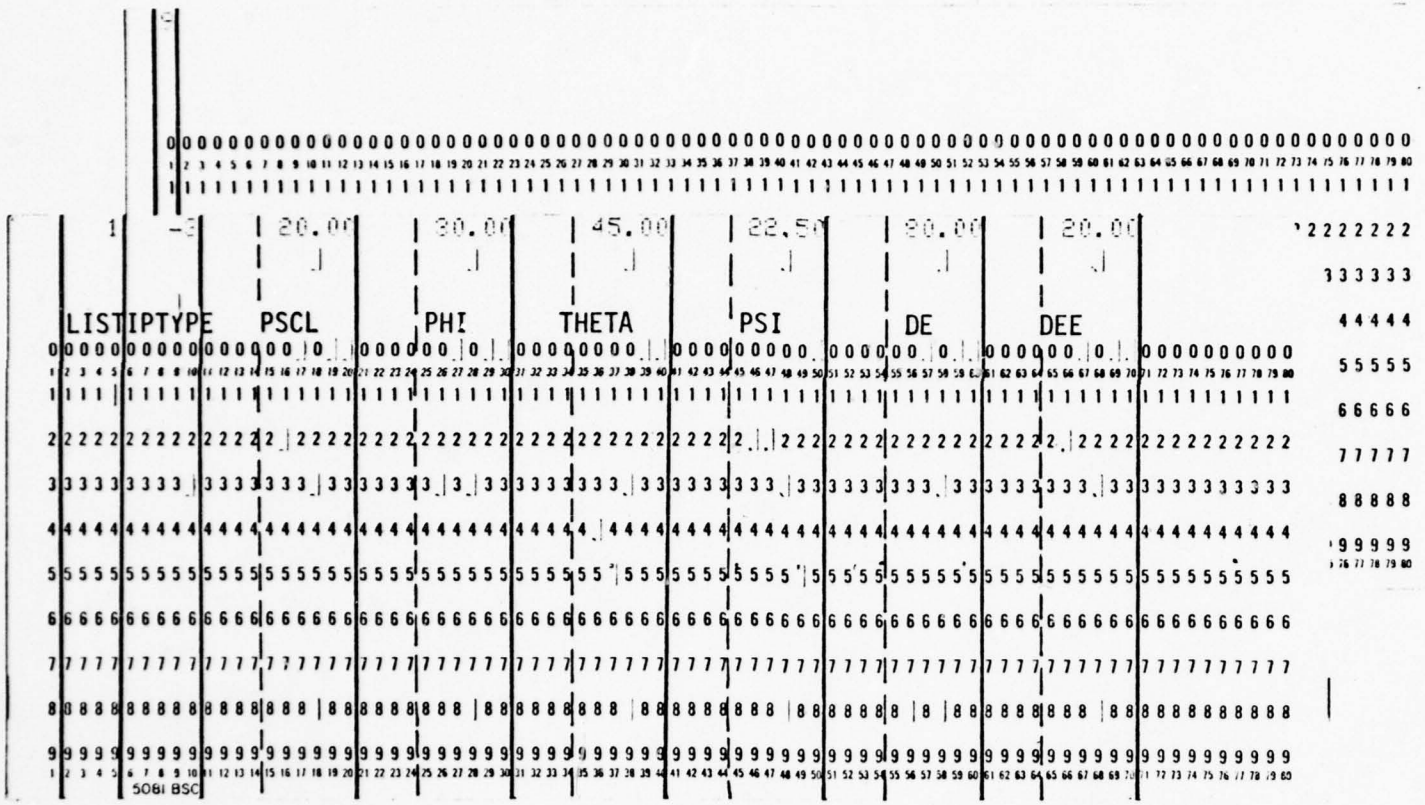


Fig. 4.6 Vantage Points

5. DESIGN OF A QUADRAW SCENE

In this section we demonstrate how to design a QUADRAW scene by building the cup shown in Fig. 5.1-3. The first part of the data file consists of three cards: title card with scene scale of 15.0 and a title string; tolerance card with five tolerances EPS1-5; and a delimeter card.

```
1   1 15.0                          CUP   (MICHAEL POTMESIL)
                0.001     0.001   0.00001    0.0001     0.001
2
```

Next, we have to specify equations of surfaces, surface numbers [Fig. 5.1] and surface types:

Number	Type	Equation	Description
1	1	$x^2 + y^2 - 16 = 0$	Outer cylinder
2	1	$-x^2 - y^2 + 12.25 = 0$	Inner cylinder
3	1	$z - 6 = 0$	Top
4	1	$z + 5.5 = 0$	Inner bottom
5	1	$-z - 6.0 = 0$	Outer bottom
6	1	$x - 0.5 = 0$	Front of handle
7	1	$-x - 0.5 = 0$	Back of handle
8	1	$2.56y^2 + z^2 - 23.04y$ $-z + 36.09 = 0$	Outer surface of handle
9	1	$-4.0y^2 - z^2 + 36y$ $+z - 72.25 = 0$	Inner surface of handle
10	2	$y = 0$	Auxiliary plane

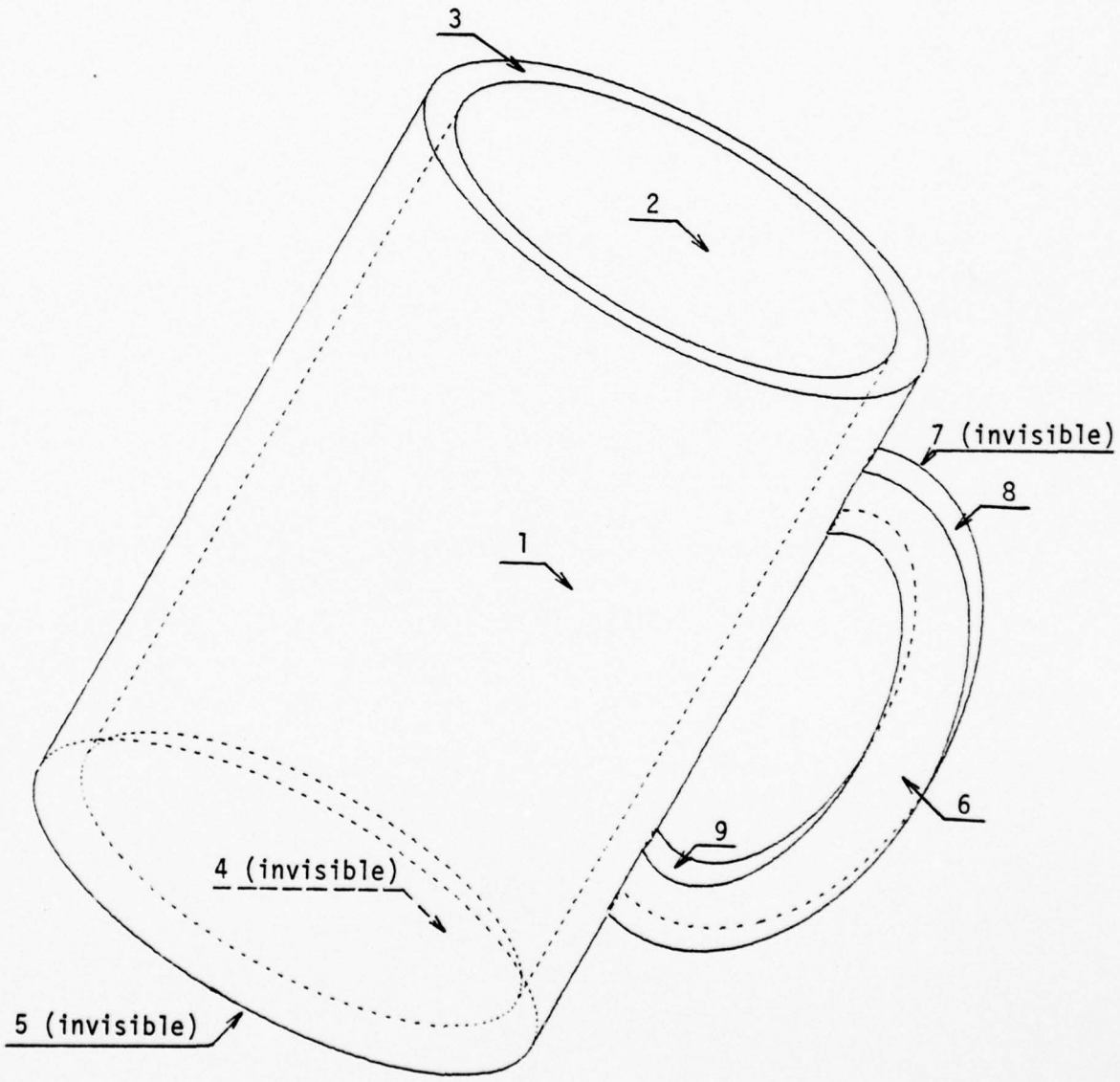


Fig. 5.1 A view of a cup (surfaces labelled)

Note how the signs of coefficients are set up. Surfaces 1 and 2, for example, have coefficients with opposite signs since the space outside the object is "outside" of surface 1 and "inside" surface 2. Surface 10 is an auxiliary transparent surface which cuts the cup in two halves and is used to eliminate ambiguities in surface bounds. We enter these cards specifying surface equations:

1	1	1.00	1.00				OUTSIDE CYL
						-16.00	
2	1	-1.00	-1.00				INSIDE CYL
						12.25	
3	1						TOP RIM
					1.00	-6.00	
4	1						INSIDE BOTTOM
					1.00	5.50	
5	1						OUTSIDE BOTTOM
					-1.00	-6.00	
6	1						HANDLE BOUND
			1.00			-0.50	
7	1						HANDLE BOUND
			-1.00			-0.50	
8	1		2.56	1.00			OUTSIDE HANDLE
				-23.04	-1.00	36.09	
9	1		-4.00	-1.00			INSIDE HANDLE
				36.00	1.00	-72.25	
10	2						AUX. PLANE
				1.00		0.00	
3							

Next, the surface bounds have to be specified. Surface bounds are used to specify the portions of a surface which are part of the object. For example, surface number 3 - the top of the cup - is a planar surface specified by $z - 6 = 0$. However, only the portion of the surface which lies between the outer and inner cylinder is a part of the object. Therefore, surface 3 is bounded by surfaces 1 and 2. The

bounded portion of surface 3 lies "inside" surfaces 1 and 2 and the list of bounds, therefore, is -1 -2 9999. Bounds can be logically grouped by the special bound 9998 which stands for logical OR; logical AND is implied between two bounds; 9999 terminates a list of bounds. Surface 1 - the outer cylinder - has complicated bounds because it is intersected by the four surfaces of the handle. It is also bounded by the top and bottom planes and an auxiliary surface number 10 is used to limit the bounding effects of surfaces 5 and 6 to the right half of the cylinder only. We obtain a rather complicated list of bounds for surface 1:

(-3 AND -5 AND -10) OR (-3 AND -5 AND 6 AND 10) OR (-3 AND -5 AND 7 AND 10) OR (-3 AND -5 AND 8 AND 10) OR (-3 AND -5 AND 9 AND 10)

This limits surface 1 to the points which are between the top and bottom planes and outside the two handle intersections. QUADRAW limits the structure of bound lists to only one OR-AND level as shown above. Similarly, we have to carefully check all the other surfaces, except the auxiliary one, and for this cup we derive the following bounds:

```
1   -3   -5  -10 9998   -3   -5    6   10 9998   -3   -5    7   10 9998   -3
    -5    8   10 9998   -3   -5    9   10 9999
2   -3    4 9999
3   -1   -2 9999
4    2 9999
5   -1 9999
6    1   10   -8   -9 9999
7    1   10   -8   -9 9999
8    1   10   -6   -7 9999
9    1   10   -6   -7 9999
4
```

Following surface bounds, the surface intersections are described. For example, surfaces 1 and 3 intersect in an obtrusive intersection. The intersection curve is a circle; the type is +2 and a cutting plane is x=0. The same applies to the other three circular intersections. Notice that the intersection of surfaces 2 and 4 has type -2 because the intersection is intrusive - the angle between the two surfaces inside the object is greater than 180°. Intersections of the handle surfaces and the outer cylinder do not need any cutting planes since they are terminated by real vertices [Fig. 5.2]. The specified surface intersections are:

```
1    3    2          1.00
1    5    2          1.00
2    3    2          1.00
2    4   -2          1.00
1    6   -2
1    7   -2
1    8   -2
1    9   -2
6    8    2
6    9    2
7    8    2
7    9    2
5
```

The last part of scene design is to specify the real vertices of each object. A real vertex is a point on the surface of the object where three or more surfaces intersect. There are 8 real vertices in the cup. They form 4 sets of real vertices - each set has 2 real vertices. Each set lies at the intersection of the outer cylinder - surface number 1 - and two handle surfaces [Fig. 5.2]. We enter these cards:

```
2    8    6    1 9999
2    8    7    1 9999
2    9    6    1 9999
2    9    7    1 9999
6
```

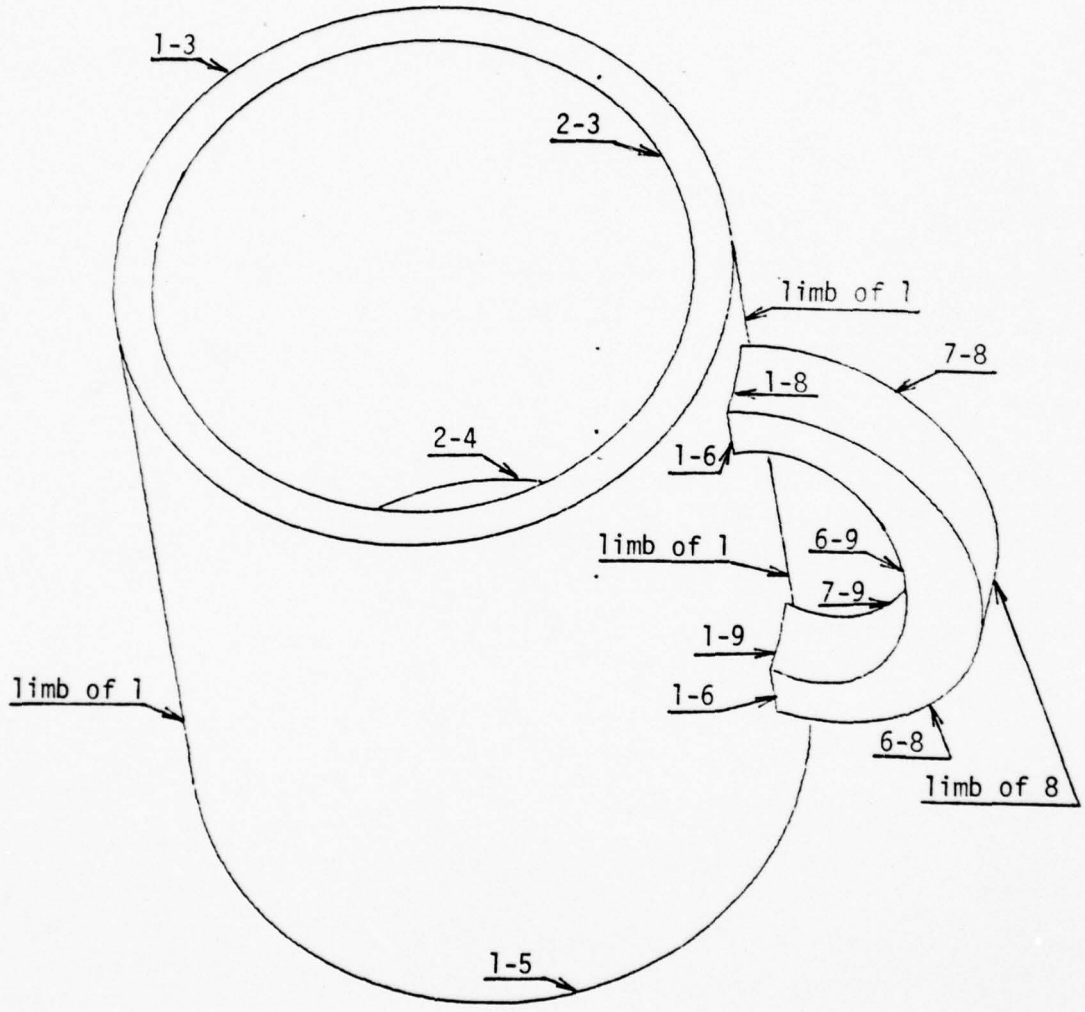


Fig. 5.2 A view of a cup (intersections and limbs labelled)

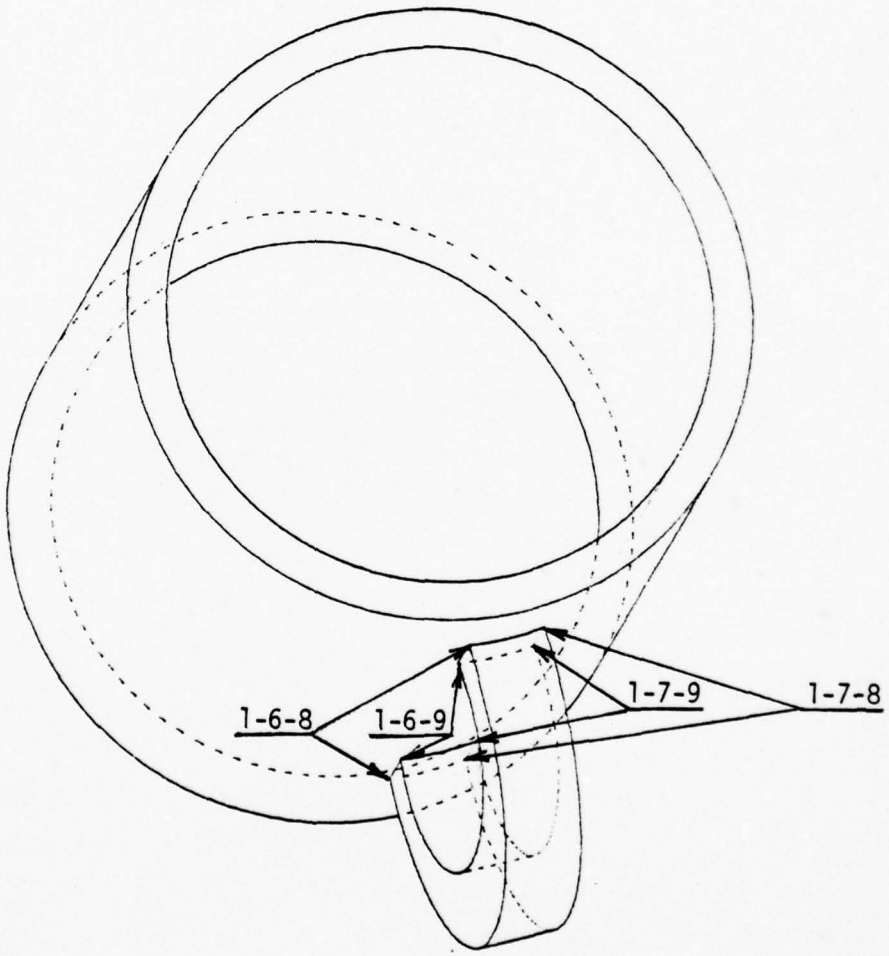


Fig. 5.3 A view of a cup (real vertices labelled)

Following the scene data we add several vantage point specifications and terminate the file by a "9" delimiter card:

```
3    20.00    45.00    45.00    22.50
3    20.00    22.00    22.00     0.00
3    20.00    70.00    10.00    30.00
3    20.00    20.00    60.00  -100.00
3    20.00     0.00    22.50    22.50
2    20.00     0.00    22.50    30.00
3    20.00   100.00   -30.00    80.00
2    20.00    45.00    75.50    30.00
9
```

REFERENCES

1. Woon, P., "A Computer Procedure for Generating Visible-Line Drawings of Solids Bounded by Quadric Surfaces", doctoral dissertation, Dept. of Electrical Engineering, New York University, December 1970 (AD 724 744).

2. Woon, P., Freeman, H., "A Procedure for Generating Visible-Line Projections of Solids Bounded by Quadric Surfaces", Information Processing 71, North-Holland Pub. Co., Amsterdam, 1120-1125, August 1972.

3. Loutrel, P.P., "A Solution to the Hidden-Line Problem for Computer-Drawn Polyhedra", doctoral dissertation, Dept. of Electrical Engineering, New York University, September 1967 (NTIS accession number: N68 13 830).

4. Potmesil, M., "An Implementation of the Loutrel Hidden-Line Algorithm", Technical report CRL-49, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, September 1976.

5. Potmesil, M., "Maintenance of QUADRAW Text on Adage Computer", Documentation Bulletin No. D-041, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, April 1977.

6. Potmesil, M., "Adage, CDC and Univac Magnetic Tape Handling System", Documentation Bulletin No. D-038A, Electrical and Systems Engineering Department, Rensselaer Polytechnic Institute, April 1977.

Appendix A

CDC 6600 Version of QUADRAW

QUADRAW has been run on a CDC 6600 computer under the KRONOS 2.1 or NOS 1.1 operating system and compiled by an FTN 4.3 compiler.

A.1 FORTRAN Source Text Library

The FORTRAN source text of the program is stored as a library file. It is maintained by the CDC text editor UPDATE. The text library contains CDC, Univac and IBM versions of the program. Versions for other computers can be added to this library. UPDATE can write on file COMPILE either the text of the CDC, Univac or IBM version. The CDC version is compiled by FTN from COMPILE into a binary file. The Univac or IBM version is copied from COMPILE to a magnetic tape and transported to a Univac or IBM computer.

The computer-dependent parts of the master text file are written as follows:

```
|   FORTRAN source text common to all versions
↓
*IF DEF,CDC
↑
|   FORTRAN source text for CDC version only
↓
*ENDIF
*IF DEF,UNIVAC
↑
|   FORTRAN source text for Univac version only
↓
*ENDIF
*IF DEF,IBM
↑
|   FORTRAN source text for IBM version only
↓
*ENDIF
↑
|   FORTRAN source text common to all versions
```

These conditional IF-ENDIF blocks are inserted whenever there are computer-dependent sections of QUADRAW. Text editor UPDATE will write the conditional blocks on file COMPILE as a part of the text to be compiled whenever the conditional value of the block, i.e., CDC, UNIVAC or IBM, is defined.

The QUADRAW source text is also maintained on an Adage computer as a back-up copy [5]. It can be written on magnetic tape in various formats [6].

A.2 QUADRAW Execution on CDC 6600

A QUADRAW magnetic tape contains two UPDATE libraries written as two tape files. The first file is the master source text of the program QUADRAW. The second file is the data library containing specifications of several QUADRAW scenes. The magnetic tape is assigned to a job by the following statement:

```
LABEL(TAPEIN,VSN=Txx,MT,D=HY,F=SI,LB=KU,PO=R)
```

where the parameters are:

TAPEIN	Local file name
VSN=Txx	Tape VSN number
MT	7-track
D=HY	800-bpi
F=SI	Format: Scope Internal
LB=KU	Labels: Kronos Unlabelled
PO=R	Read only

a) To copy the first file, update it, compile it and catalog the new UPDATE library and binary file:

```
.......
.......
LABEL(TAPEIN,.,.....)
REWIND(TAPEIN)
COPYBF(TAPEIN,OLDPL,1)
PURGE(QLIB/NA)
DEFINE(NEWPL=QLIB)
UPDATE(F,P=OLDPL,N=NEWPL)
FIN(I=COMPILE,L=0)
PURGE(BQDRAW/NA)
DEFINE(BQDRAW)
COPYEI(LGO,BQDRAW,X)
end-of-record
*DEFINE CDC
*IDENT PLOTS
*DELETE PLOTSBL.1
      CALL PLOTS(LIMIT,'plotter output ident')
end-of-information
```

This run will produce two permanent files:

QLIB	master text library of QUADRAW
BQDRAW	binary file of WUADRAW

b) To copy and update the second file - QUADRAW scenes:

```
.......
.......
LABEL(TAPEIN,....)
REWIND(TAPEIN)
SKIPF(TAPEIN,1)
COPYBF(TAPEIN,OLDPL,1)
PURGE(QSLIB/NA)
DEFINE(NEWPL=QSLIB)
UPDATE(P=OLDPL,N=NEWPL)
end-of-record
*IDENT NEW
end-of-information
```

This run will produce a permanent file QSLIB containing several DECKs. Each DECK contains the specifications of a QUADRAW scene and several vantage points.

c) To execute QUADRAW; input - FORTRAN unit 5 - is read from file COMPILE: output - FORTRAN unit 6 - is written on file OUTPUT:

```
NAME,T100,CM15000.          QUADRAW
ACCOUNT(...,....)
CHARGE(...,....)
REWIND(OUTPUT)
ATTACH(OLDPL=QSLIB)
UPDATE(P=OLDPL,D)
ATTACH(BQDRAW)
LOAD(BQDRAW)
EXECUTE(QDRAW,COMPILE)
end-of-record
*COMPILE CUP
*COMPILE FIG2
end-of-information
```

The COMPILE file must have 80 characters of data per line; therefore, use the D option on the UPDATE card as shown above. Two or more decks of QUADRAW scenes from file QSLIB may be combined by UPDATE on file COMPILE and run together.

d) The Univac or IBM version of QUADRAW are generated from the master source text on a CDC 6600 by the following run:

```
.......
.......
LABEL(TAPEOUT,.....)
REWIND(TAPEOUT)
ATTACH(OLDPL=QLIB)
UPDATE(F,P=OLDPL)
COPYCF(COMPILE,TAPEOUT,1)
ATTACH(OLDPL=QSLIB)
REWIND(COMPILE)
UPDATE(P=OLDPL,D)
COPYCR(COMPILE,TAPEOUT,10)
REWIND(TAPEOUT)
end-of-record
*IDENT UNIVAC or *IDENT IBM
end-of-record
*COMPILE APOLLO
```

```
*WEOR
*COMPILE CUP
*WEOR
*COMPILE SPHR8
*WEOR
.
.
.
.
.
end-of-information
```

Change the density, parity and 7/9 track parameters in the above LABEL statement as required by the Univac or IBM computer.

A.3 SCOOP Plotting Package

QUADRAW uses the following subroutines of the SCOOP plotting package:

```
PLOTS(LIMIT,'your identification string')
PLOT(X,Y,IPEN)
```

PLOTS opens plotting output up to LIMIT inches long and labels it with 'your identification string'.

PLOT moves the pen from the current location to location (X,Y). IPEN can have the following values:

2	solid line(pen down)	and do not reset plot origin
3	skip (pen up)	and do not reset plot origin
-2	solid line(pen down)	and reset plot origin to (X,Y)
-3	skip (pen up)	and reset plot origin to (X,Y)
999	skip (pen up)	and close plotting output

A.4 Timing Routine

MSTIME calls FTN library subroutine SECOND(TIME) which returns the time since the beginning of program execution in seconds in floating point variable TIME.

Notes

The current CDC implementation of QUADRAW has these features:

a) Each picture is drawn on one 12 x 12 inch sheet of paper.

b) PLOTS opens output on green graph paper. PLOTSBL with the same arguments as PLOTS opens output on white blank paper.

c) LIMIT, i.e., the maximum total length of the paper, is set to 600 inches.

Appendix B

Univac 1110 Version of QUADRAW

QUADRAW has been run on a Univac 1110 computer, in demand mode under EXEC-8.

There are two types of magnetic tape files from which QUADRAW can be run on a Univac computer:

a) Univac program file written in COPY format containing symbolic and relocatable elements of all QUADRAW routines and symbolic elements of all QUADRAW scenes.

b) File written in BCD code on a CDC 6600 computer. This file is read and converted by a Univac as described in [6].

B.1 A QUADRAW Run

The following EXEC-8 statements illustrate how to run QUADRAW on a Univac 1110 computer from a demand terminal. Assume that we have a 9-track QUADRAW tape number Txxxxx written in Univac COPY format. We assign an unregistered mass storage file called QDRAW and copy the tape file.

```
@ASG,UP   QDRAW.
@MSG,W    MAY I USE TAPE 'Txxxxx' 9-TRK FOR 5 MIN
@ASG,T    TAPEIN.,T,Txxxxx
@COPY,G   TAPEIN.,QDRAW.
@FREE     TAPEIN.
```

File QDRAW contains all symbolic and relocatable elements of QDRAW. We may list the directory of QDRAW:

```
@PRT,T    QDRAW.
```

If it is necessary to edit or compile any element use the editor and compiler:

```
@ED    QDRAW.eltname,QDRAW.eltname
 .
 .

@FOR   QDRAW.eltname,QDRAW.eltname
 .
 .
```

When all relocatable elements are updated we create an absolute element:

```
@PREP  QDRAW.
@MAP   QDRAW.QDRAW,QDRAW.QABS
 IN    QDRAW.
 LIB   GRAIL$*LIB.
 END
```

An absolute element called QABS was added to file QDRAW. This element can be executed after the plotting package was directed to plot its output on a digital plotter:

```
@GRAIL$*LIB.PLOTMODE,C
@XQT   QDRAW.QABS
```

QUADRAW is now waiting for the first scene to be entered. We may enter the CUP and APOLLO scenes by:

```
@ADD   QDRAW.CUP
 .
 .
 .
 .

@ADD   QDRAW.APOLLO
 .
 .
 .
 .
@EOF
```

The execution of the program was terminated. The file QDRAW may be saved back on the same tape as the second tape file by commands:

```
@ASG,T    TAPEOUT.,T,TxxxxxR
@TSKIP    TAPEOUT.
@COPY,GM QDRAW.,TAPEOUT.
@FREE     TAPEOUT.
```

B.2 GRAIL Plotting Package

QUADRAW uses the following subroutines of the GRAIL plotting system at the Univac 1110 installation:

```
SETPLT (300.0,11.0,0)
ORIGIN(XORG,YORG,MODE)
SCRIBE(X,Y)
SLEW(X,Y)
ENDPLT
```

SETPLT opens plotting output and establishes 300 x 11 inch maximum plotting area. ORIGIN positions the pen to the center of a new 11 x 11 inch picture area. SCRIBE draws a line from the current position of the pen to position X,Y. SLEW moves the pen (in the up position) to position X,Y. ENPLT closes the plotting output.

B.3 Timing Routine

MSTIME calls the system subroutine CPTIME(M) which returns the elapsed execution time of the run in integer milliseconds in variable M.

Appendix C

IBM System/360-370 Version of QUADRAW

QUADRAW has been run on an IBM 360/67 computer using the Alpha time-sharing system.

The IBM version of QUADRAW is written on magnetic tape by a CDC computer in BCD code. It is read and listed by an IBM computer with the following JCL statements:

```
//QDRAW     JOB...
//STEP1     EXEC    PGM = IEBPTPCH,REGION=32K
//SYSPRINT  DD      SYSOUT=A
//SYSIN     DD      DUMMY
//SYSUT1    DD      UNIT=(TAPE,1,DEFER)
                    VOL=(,RETAIN,1,SER=xxxxxx)
                    DSN=QDRAW,
                    LABEL=(1,NL)
                    DISP=(OLD,KEEP,KEEP)
                    DCB=(RECFM=FB,LRECL=80,BLKSIZE=80,
                        BUFNO=2,DEN=2,TRTCH=ET)
//SYSUT2    DD      SYSOUT=A
//SYSIN     DD      *
/*
To punch the file on cards use this SYSUT2 statement
//SYSUT2    DD      SYSOUT=B
```

C.1 CALCOMP Plotting Package

QUADRAW uses the following subroutines of the CALCOMP plotting package:

```
PLOTS(0,0,0)
PLOT(X,Y,IPEN)
```

PLOTS opens plotting output. The plotted paper is labelled at both ends with the job name.

PLOT moves the pen from the current location to location (X,Y). IPEN can have the following values:

2	solid line	(pen down)	and do not reset plot origin
3	skip	(pen up)	and do not reset plot origin
-2	solid line	(pen down)	and reset plot origin to (X,Y)
-3	skip	(pen up)	and reset plot origin to (X,Y)
999	skip	(pen up)	and close plotting output

C.2 Timing Routine

MSTIME calls system routine TIMEH(M) which returns the time since the beginning of program execution in tenths of seconds in integer variable M.

Notes

The current IBM implementation has these features:

a) Each picture is drawn within an 8 x 8 inch area of paper.

b) The plotter uses 28-inch-wide paper. Therefore, three pictures are drawn along the width of the paper.

Appendix D

QUADRAW SCENES

A set of ten scenes has been used to test the program. Each scene description is followed by several vantage point specifications which are terminated by a "9" delimiter card. Several scenes can be combined for a single program run. The scenes for which descriptions exist are:

APCLLO - a model of the Apollo spacecraft [Fig. D.1-2].

CUP - a model of a cup [Fig. 5.1-3].

SPHR8 - a scene made of 8 spheres which are cut by two planes [Fig. 1.2 and D.3].

CYLIN4 - a scene made of 4 cylinders [Fig. D.4].

FIG1 - a sphere intersecting with a cylinder [Fig. D.5].

FIG2 - two cone parts intersecting with a cylinder [Fig. D.6].

ELPHYP1 - four elliptic hyperboloids of one sheet [Fig. D.7].

ELPHYP2 - four elliptic hyperboloids of two sheets [Fig.D.8].

CYLIN4P - four cylinders with perspective-projection vantage points [Fig. D.9].

SPHERES - a scene made of eight non-intersecting spheres [Fig. D.10].

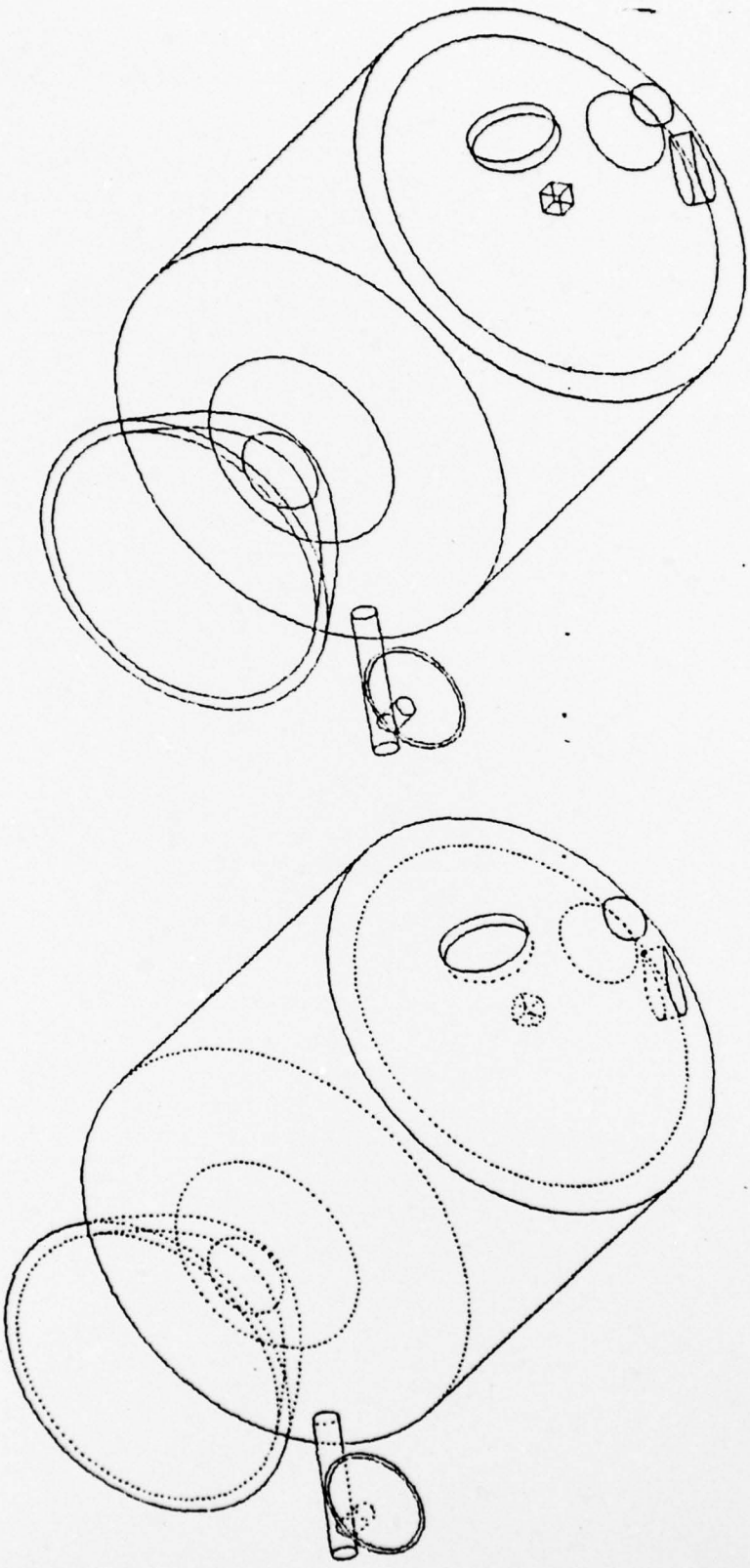


Fig. D.1 Two views of scene APOLLO (all lines and hidden-lines dashed drawn)

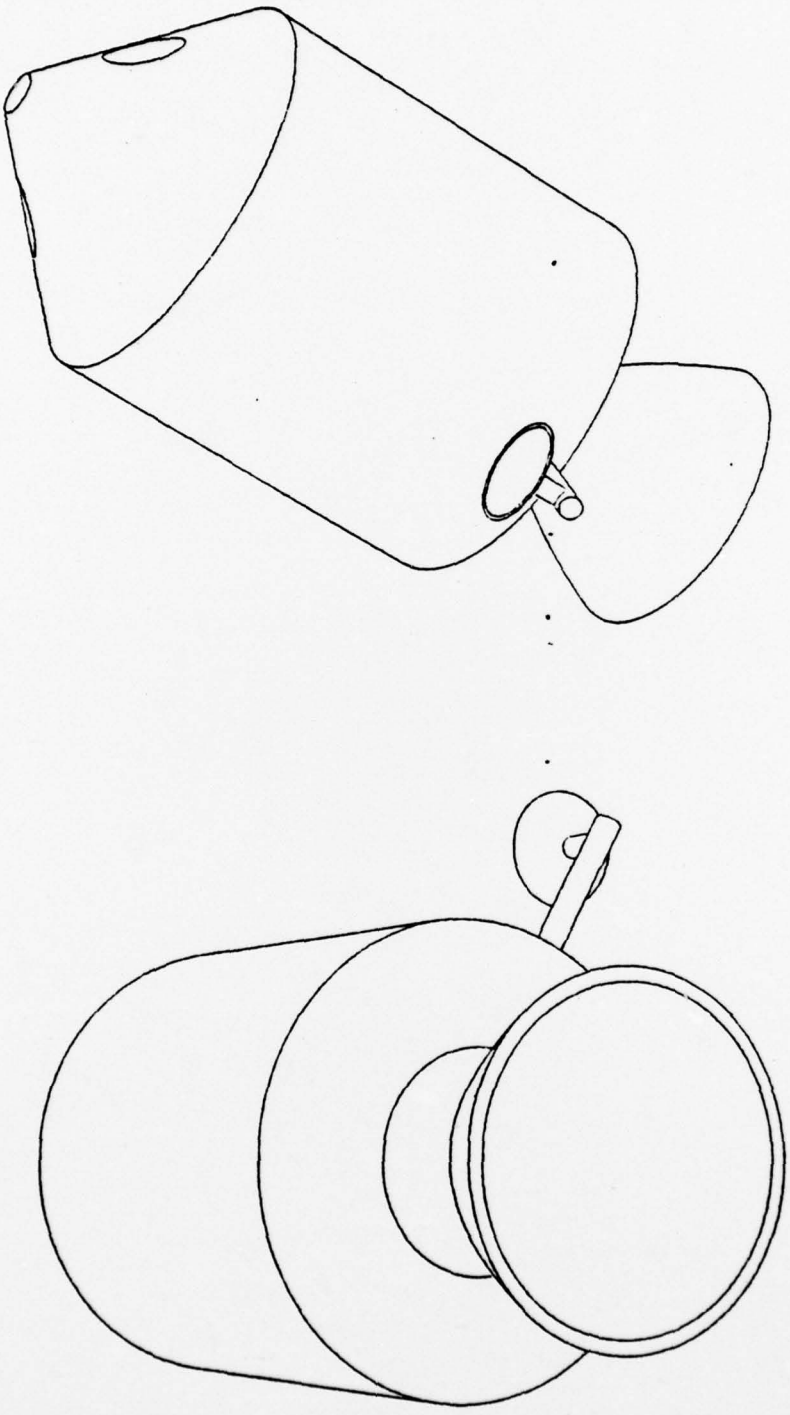


Fig. D.2 Two views of scene APOLLO (only visible lines drawn)

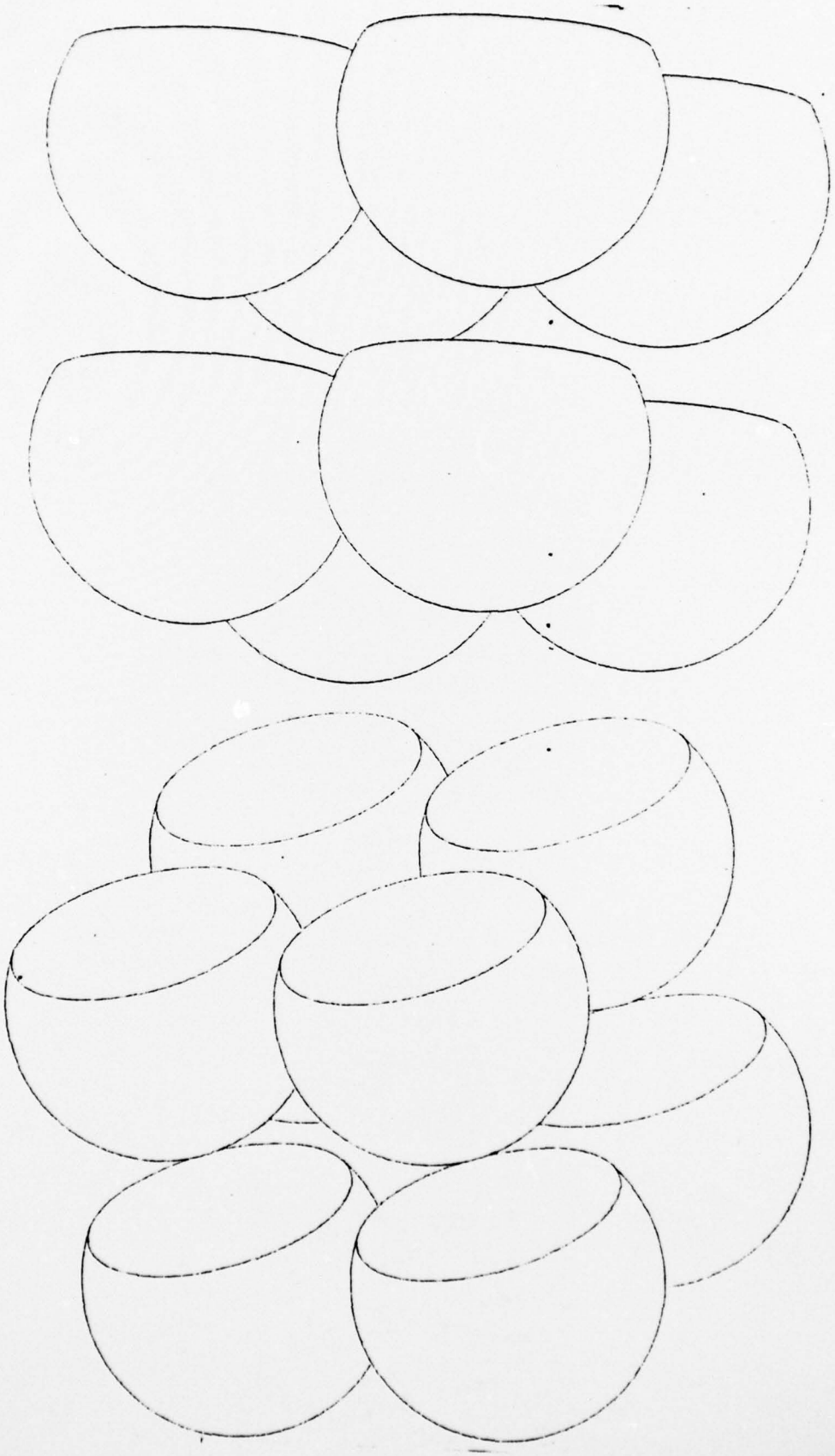


Fig. D.3 Two views of scene SPHR8

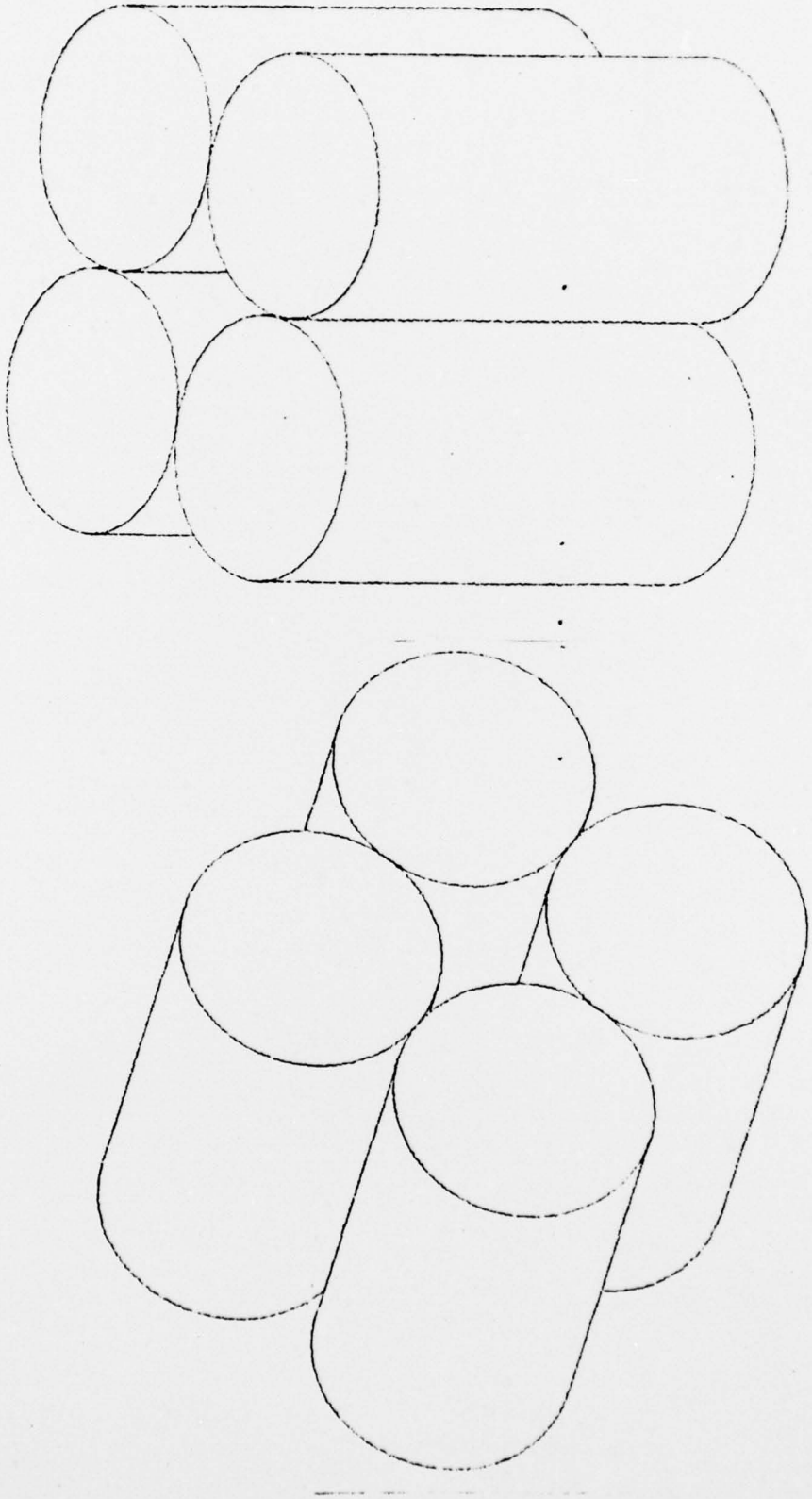


Fig. D.4 Two views of scene CYLIN4

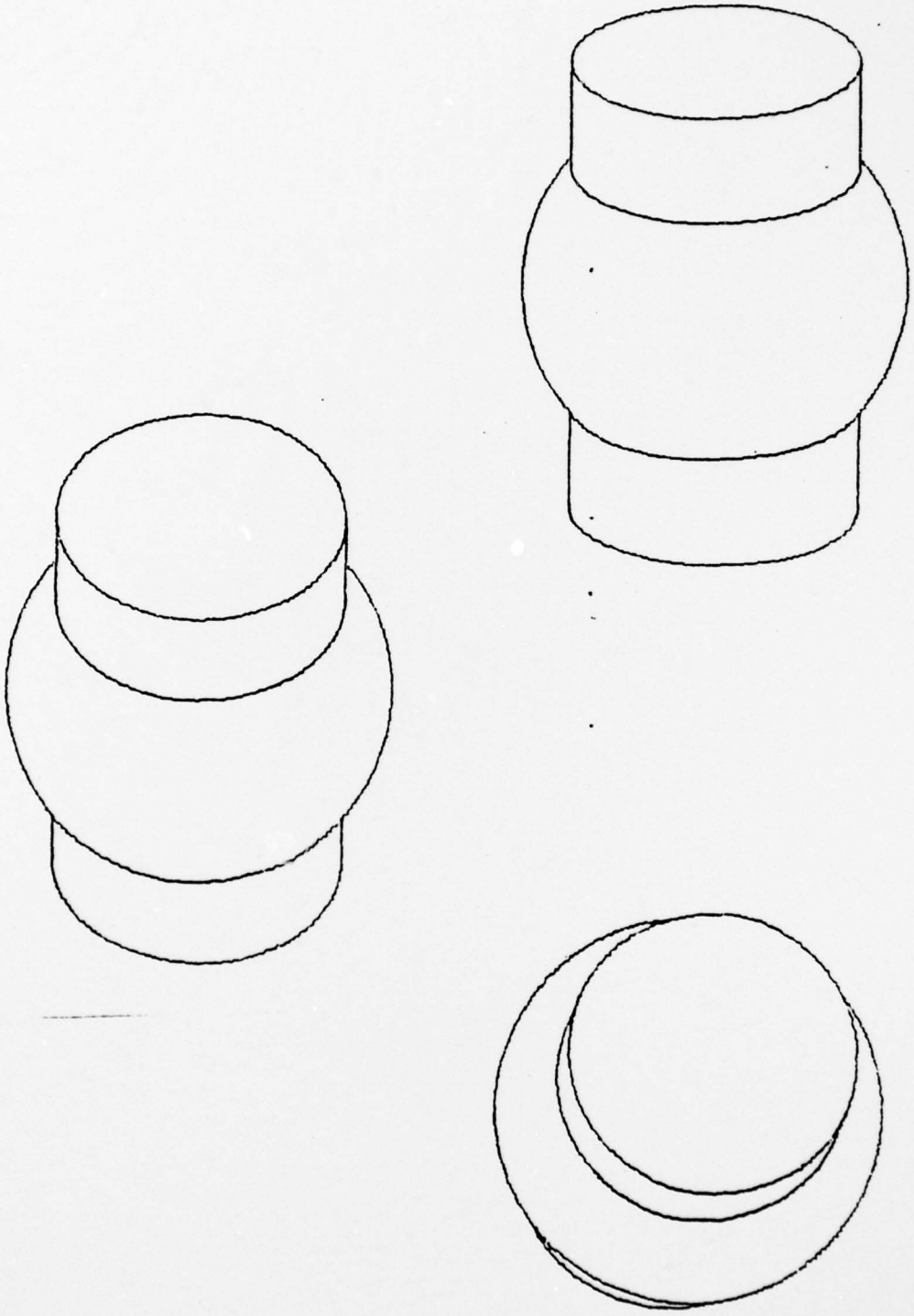


Fig. D.5 Three views of scene FIG1

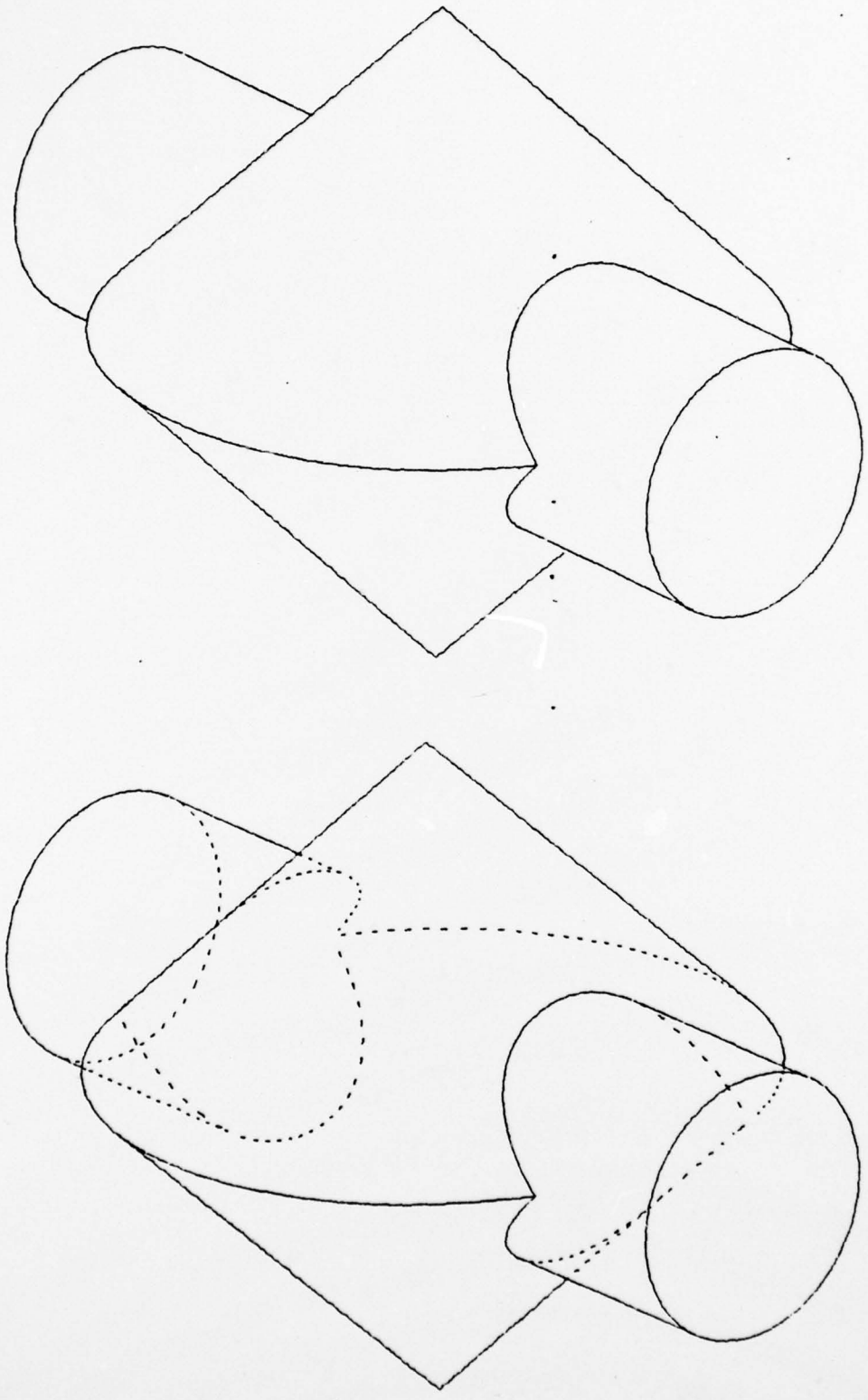


Fig. D.6 Two views of scene FIG2

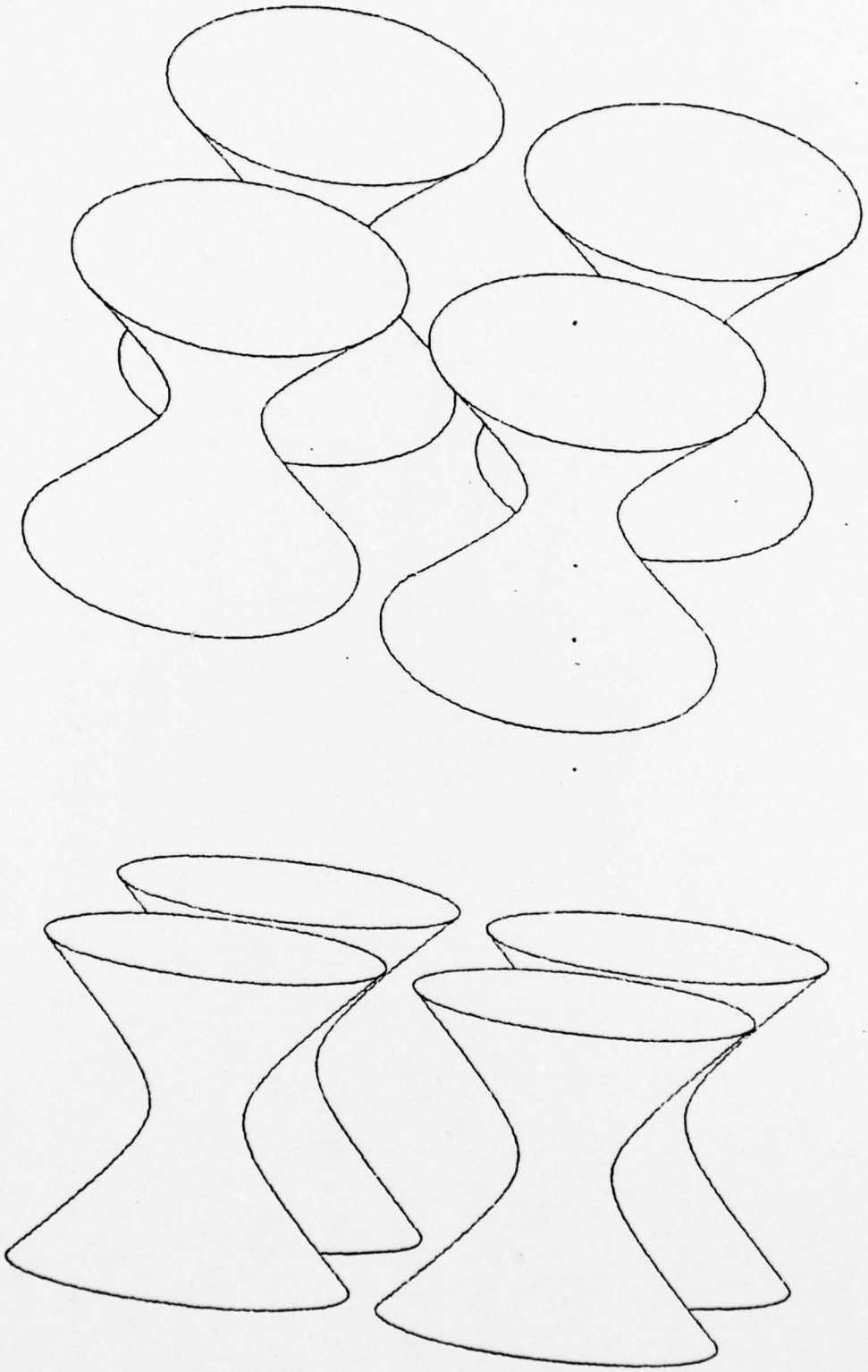


Fig. D.7 Two views of scene ELPHYP1

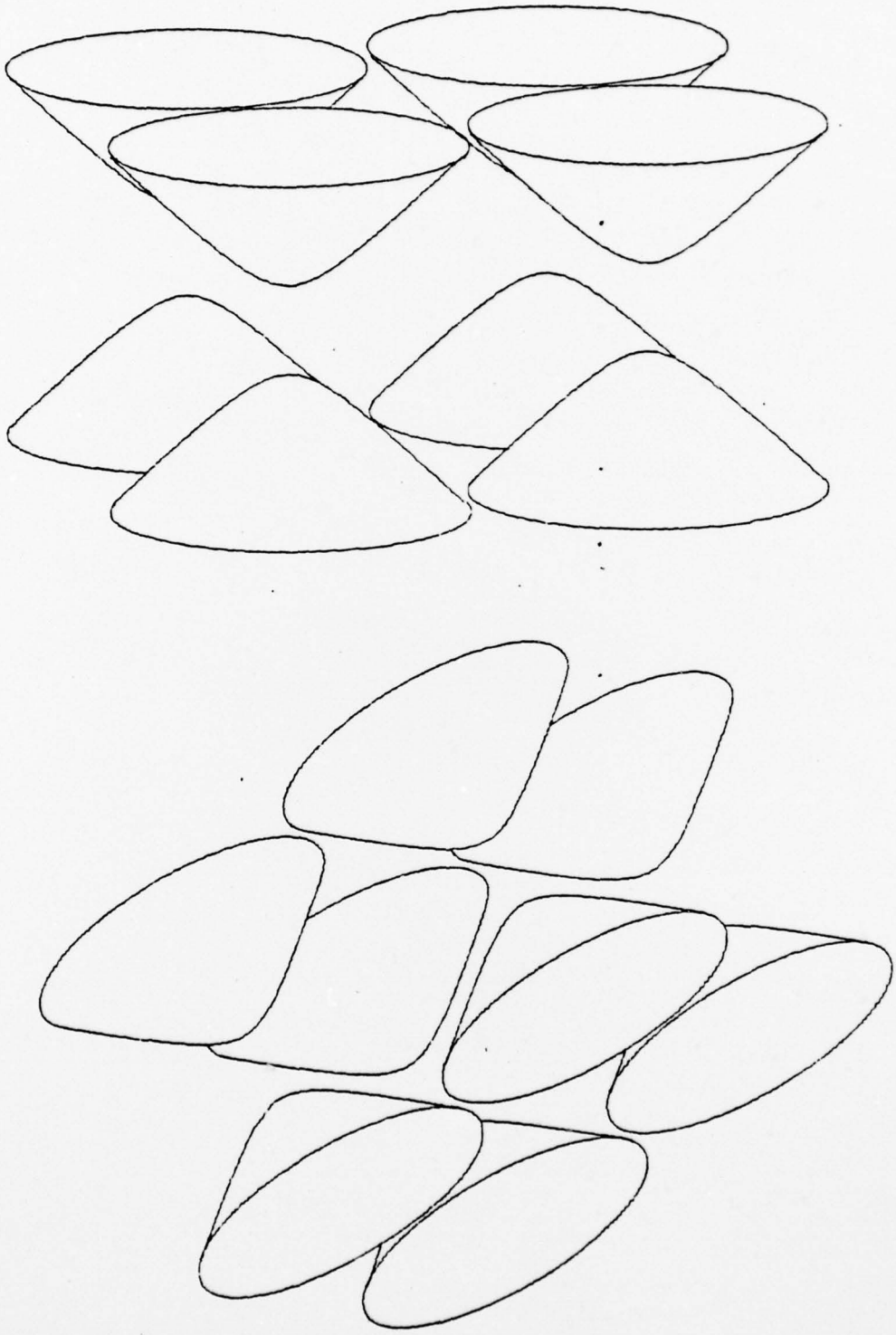


Fig. D.8 Two views of scene ELPHYP2

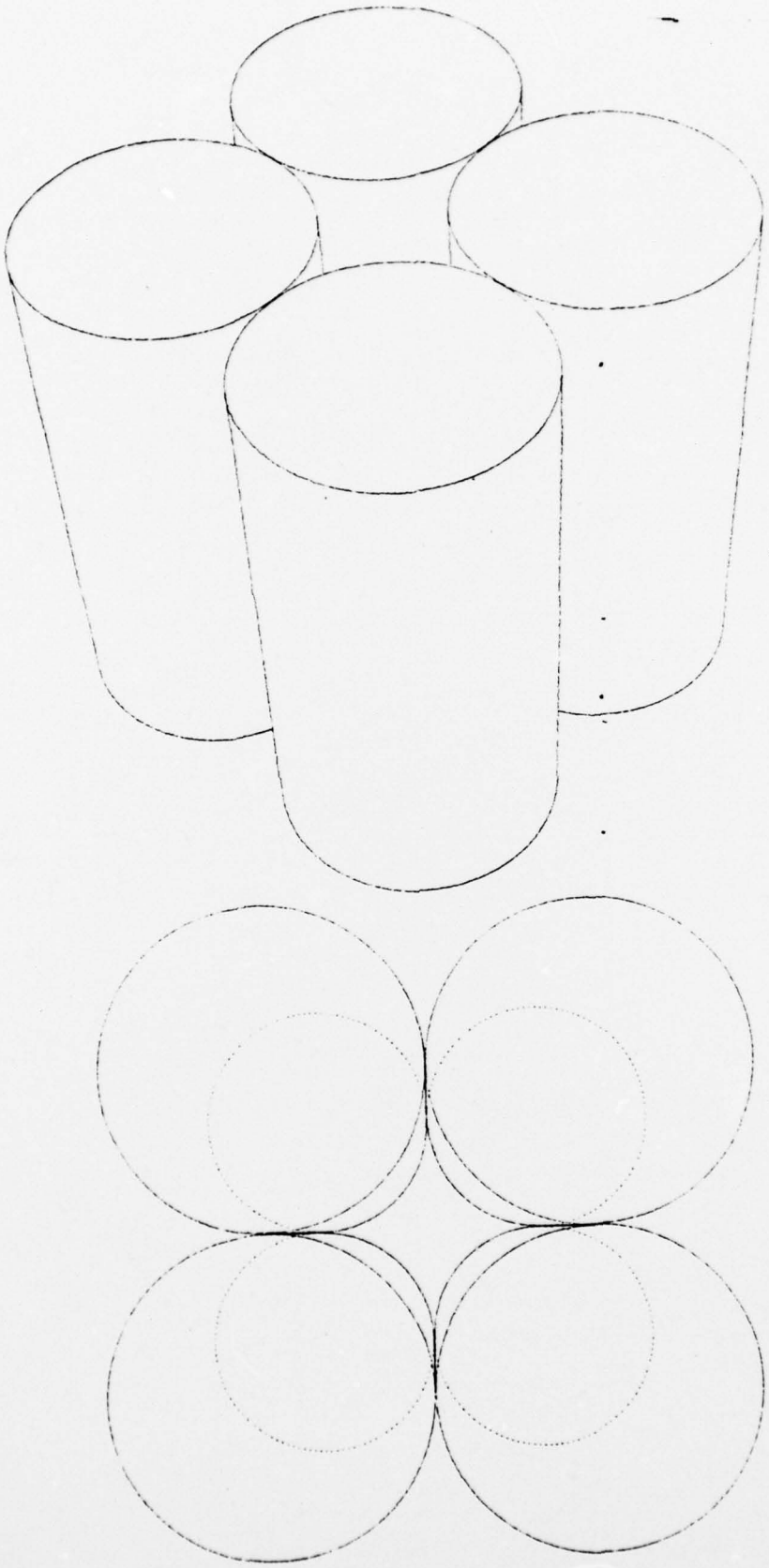


Fig. D.9 Two views of scene CYLIN4P
(perspective projections)

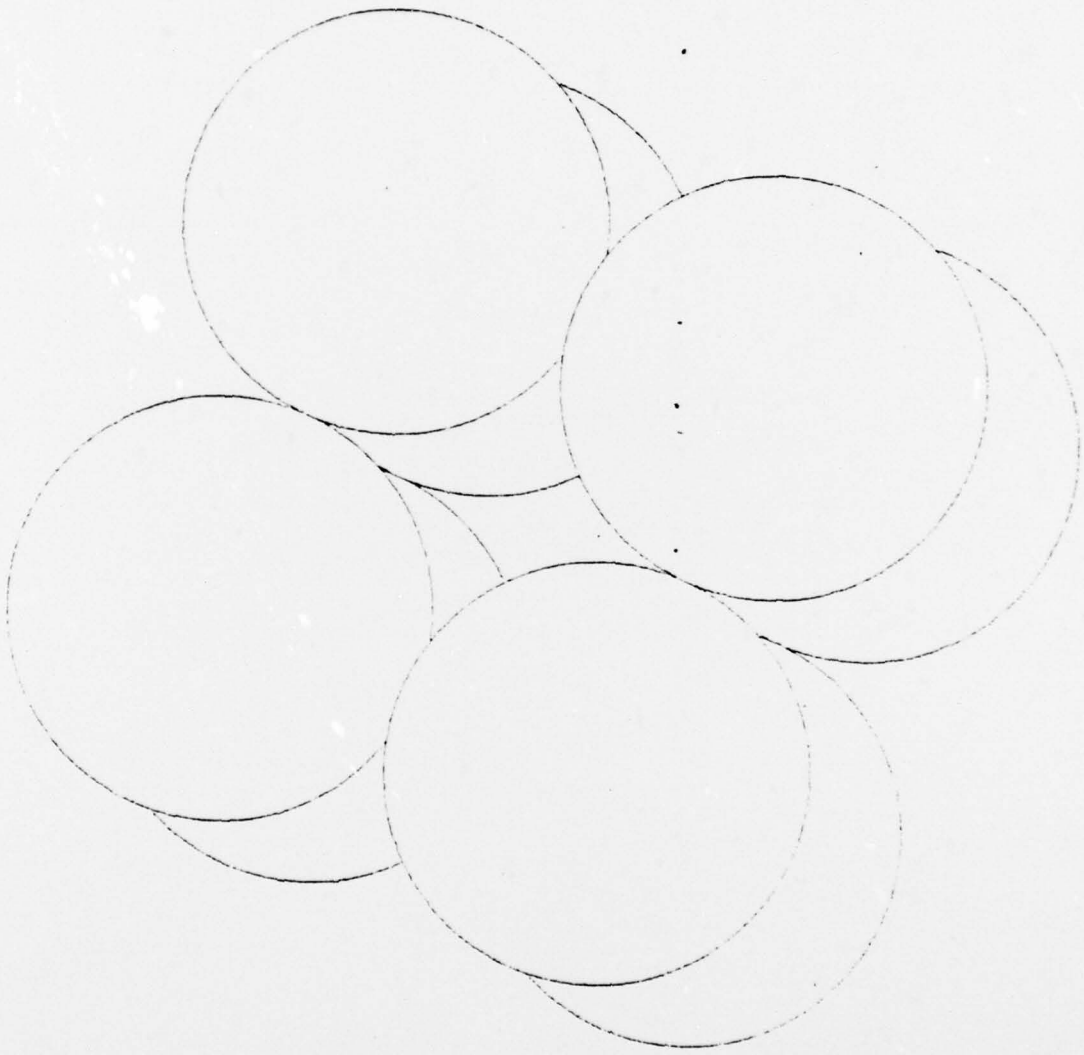


Fig. D.10 A view of scene SPHERES

Index of QUADRAW Routines

34.	INPOB	2.2.1
35.	INPVW	2.2.4
36.	INSIDE	2.2.8
37.	INTBL9	2.4.8
38.	INVIS	2.3.9
39.	IPNT	2.5.12
40.	MSTIME	2.7.1
41.	NEWT	2.5.5
42.	POLAR	2.5.11
43.	PWRT	2.5.6
44.	POLRT	2.5.6
45.	QDRAW	2.1.1 (main program)
46.	REDGE	2.2.6
47.	ROOT2	2.5.3
48.	ROTPT	2.3.4
49.	RVERT	2.2.2
50.	SINT	2.2.3
51.	SORT1	2.3.13
52.	STO11	2.4.3
53.	STO12	2.4.3
54.	STO21	2.4.3
55.	STO31	2.4.3
56.	STO32	2.4.3
57.	STO41	2.4.3
58.	STO42	2.4.3
59.	STO51	2.4.3
60.	STO61	2.4.3
61.	STO71	2.4.3
62.	STO72	2.4.3
63.	STO73	2.4.3
64.	STO81	2.4.3
65.	STO85	2.4.3
66.	ST151	2.4.4
67.	ST152	2.4.5
68.	SUBS1	2.5.4

69.	TABLE	2.4.7
70.	TRNSF	2.2.5
71.	VEDGE	2.2.7
72.	WDUMP	2.3.10

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: AFOSR-TR-77-1243
2. GOVT ACCESSION NO.:
3. RECIPIENT'S CATALOG NUMBER: Technical Rept.
4. TITLE (and Subtitle): QUADRAW User Manual,
5. TYPE OF REPORT & PERIOD COVERED: Interim
6. PERFORMING ORG. REPORT NUMBER: CRL-53
7. AUTHOR(s): Michael Potmesil
8. CONTRACT OR GRANT NUMBER(s): AFOSR-76-2937
9. PERFORMING ORGANIZATION NAME AND ADDRESS:
Rensselaer Polytechnic Inst
Electrical & Systems Engineering Dept
Troy, NY 12181
Comp. Res Lab
10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 61102F 2304/A2 A2
11. CONTROLLING OFFICE NAME AND ADDRESS:
Air Force Office of Scientific Research/NM
Bolling AFB DC 20332
12. REPORT DATE: Jun 77
13. NUMBER OF PAGES: 77p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)
15. SECURITY CLASS. (of this report): UNCLASSIFIED
15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)
Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

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

This report is a user manual for the program QUADRAW which draws visible-line-projection pictures of objects bounded by sections of quadric surfaces. The report describes three versions of the program, one each for the CDC 6600, Univac 1110 and IBM System/360-370 computers. It also provides the necessary documentation to permit one to modify the program for any other computer.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

409952