HEINRICH-HERTZ-INSTITUT — BERLIN-CHARLOTTENBURG

Technischer Bericht Nr. 148

FLAVIS – A hidden line algorithm for displaying spatial constructs given by point sets

by

Dipl.-Ing. Peter Mahnkopf Dr.-Ing. José Luis Encarnaçao

Berlin

1973

HEINRICH-HERTZ-INSTITUT BERLIN-CHARLOTTENBURG

Technischer Bericht Nr. 148

FLAVIS - A hidden line algorithm for displaying spatial constructs given by point sets

by

Dipl.-Ing. Peter Mahnkopf Dr.-Ing. José Luis Encarnaçao

Berlin
1973

Technischer Bericht Nr. 148

FLAVIS - A hidden line algorithm for displaying spatial constructs given by point sets

Summary:

The suppression of hidden lines is, from the visual standpoint, of great importance when displaying and manipulating threedimensional objects. The FLAVIS method has been developed as a solution to this problem. It is a point set method, i.e. the objects to be displayed are defined as a set of points in space. The method is very general and independent of the form of the input data. In order to minimize the computer time required, the point set is projected onto a cartesian grid with vertex length n. During the visibility investigation, only those areas of the surface whose intersection with the grid square in which the test point is located is non zero are considered. The computing time required by FLAVIS is (in a first approximation) inversely proportional to the square of the grid size n.

The authors wish to thank Dipl.-Ing. R. Eckert /6/, /7/ and Dipl.-Ing. E. Kniepen /8/, /9/ for their contributions, which help essentially to develope this method.

Heinrich-Hertz-Institut für Schwingungsforschung
Abteilung Informationsverarbeitung

The Authors:

(P. Mahnkopf) (J. Encarnacao)

Der Abteilungsleiter
(W. Giloi)

Der Institutsdirektor
(G. Boerger)

Berlin-Charlottenburg, 31.12.1973

Contents:

1. Introduction

The displaying and manipulation of three-dimensional objects is an important problem of computer graphics.
In order to obtain a truly spatial impression of the object being displayed, the suppression of hidden lines is of great importance. Many algorithms have been developed to solve this problem /3/,/4/. In 1968, J. Encarnaçao began research in this area at the Institut für Informationsverarbeitung I of the TUB /5/. He implemented the first versions of FLAVIS. Under his supervision, this project was continued and the method generalized by R. Eckert /6/,/7/ and E. Kniepen /8/,/9/. The optimization of FLAVIS was then undertaken by P. Mahnkopf /10/, Heinrich-Hertz-Institut - Abteilung Informationsverarbeitung - in Berlin, in close cooperation with the other author of this report. The purpose of this report is to describe the algorithm and the present implementation of FLAVIS and, through program listings, make it available to a larger group.

In the present state of the art, there are four types of visibility algorithms which differ mainly in the types of basic data used.

1. Line methods
2. Surface methods
3. Grid methods
4. Point-set methods.

In addition to this problem of the choice of basic elements there is the problem of the definition of the object to be displayed. We can distinguish between objects:
given by a set of plane surfaces,
approximated by a set of plane surfaces,
given by a set of equations, or
approximated by polygons.

There are, thus, two basic definitions of objects:

a) Objects given as polygons and
b) Objects given by point sets obtained or calculated in some manner.

The line methods /11/, /12/, investigate the various lines of the polygons for visibility relative the rest of the surfaces. Since the surfaces describing the objects are almost always concave, these methods require either that the surfaces be subdivided into convex parts or that complex sorting algorithms be used in order to take all concave properties into account. The grid methods have been developed to solve these difficulties which cause increased computer time. These methods subdivide a given projection by a grid (of 256 lines, for example). The intersections of the polygonal lines with the grid lines are compared with one another in several lists, in order to determine visibility according to various criteria.

The next possibility is to substitute all points to be investigated for visibility into the equations defining the object to see whether there are other points also fulfilling the equations which hide the point under cosideration. These methods assume the equations to be given however and, due to the edges of the objects, require the solution of complicated inequalities.

The fourth and most general method is applicable when the objects to be displayed are considered as a set of points. The point set can be obtained by interactive input, by digitalization, by the calculation of explicit or implicit equations or by interpolation, for example.

FLAVIS has been developed for this type of object definition. It has not only the advantage of general applicability but also requires no special form of input data. The data exist in the form of sextuples, i.e. a point P is given by its x-, y- and z-coordinates, the u and v lines surrounding it, and its object if more than one object is to be displayed at the same time.

$$P = P(x,y,z,u,v,L)$$

Thus, one works with sets of points. The algorithm cannot be a point method, however, since this would lead to unacceptable computation time /2/. In order to overcome this difficulty, the set of points to be displayed is projected onto a cartesian grid with vertex size n. If a point is to be tested for visibility, only those patches having a non-zero intersection with the grid square containing that point are used. The line joining two test points is, in turn, subdivided into test points. In order not to obtain too many test points, lines are subdivided according to their length. It has been determined that the computer time needed by FLAVIS is inversely proportional to the square of the grid vertex length n:

$$T \sim \frac{1}{n^2}$$

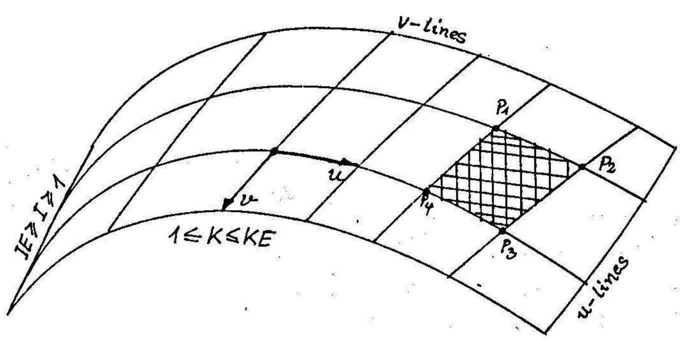
(This equation is an approximation of $T \sim \frac{k_1}{n^2} + \frac{k_2}{n} + k_3 + k_4 \cdot n$ for small n)

The current implementation for the C90/40 requires computer time in the order of minutes. An implementation for larger machines (i.e. IBM /360, /370 or CDC 6000 -serie) yields time in the order of seconds.

2. Description of the method

This method has been developed for curved boundaries. It is applicable to general point sets (i.e. Coons' surfaces, etc.). Several surfaces may also be treated using FLAVIS; the surfaces must be represented by a parallel projection. In the following, one particular representation form is described which was implemented by an interpreter. The FLAVIS method, however, is independent of this choice.

The structure to be displayed is given by a u-v line grid Each node P is determined by its x,y, and z coordinates and by its associated u and v lines.



$P_1 = P_1(x, y, z)$

$x = x(u, v)$

$y = y(u, v)$

$z = z(u, v)$

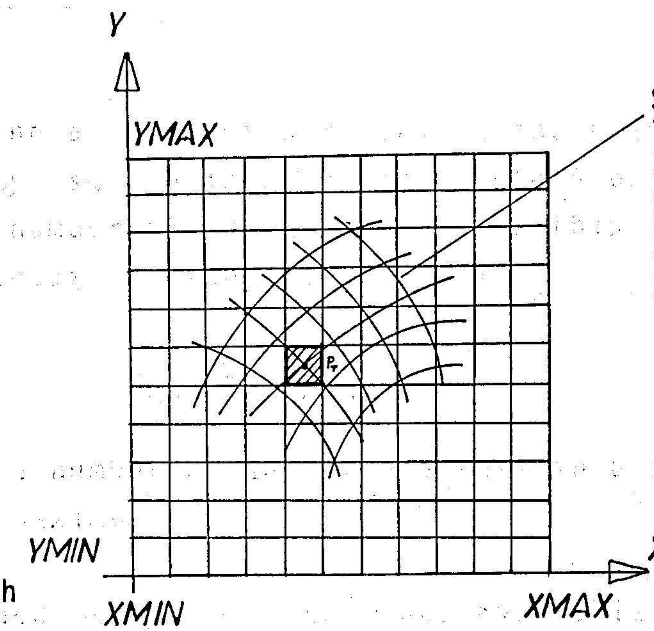
Point	Indices
P_1	I, K
P_2	$I, k+1$
P_3	$I+1, K$
P_4	$I+1, k+1$

An 11 x 11 cartesian grid is superimposed to the display area. The size of the grid is of great importance as far as computer time is concerned. To a first approximation and up to a certain limit the time required by this method decreases quadratically with the square of the number of grid elements.

In principle, however, the grid size can be freely chosen. The surface to be displayed is then imbedded in this cartesian screen as a parallel projection.

First, surface elements (patches) are assigned to grid squares (subroutine INTER). The patches are then indexed; for reasons of storage economy 2 indices are stored in each word of the index list (subroutine ASSIGN). The indices are used later to identify the patches. In another list, the number of patches in each grid square is stored. The visibility algorithm considers only one grid square at a time, not the whole surface. It must search for all patches which have points in the grid square under consideration.

P_T = Test point in grid element 5,6



Surface-defining line-grid
In one grid patch there can be several patches, which can belong to several surfaces

▨ grid -patch for further visibilty testing

If $F_{u,v}$ denotes the set of all points,belonging to one Δu- Δv-element

$K_{\nu,\varkappa}$ denotes the set of all points, belonging to one grid patch

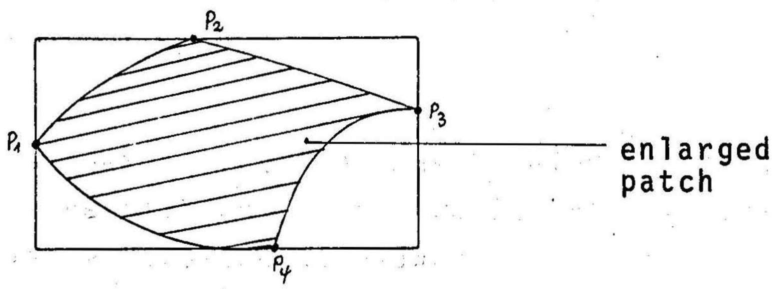
$\nu,\varkappa$ denotes indices for the grid lines,

then $$D_{u,v} = F_{u,v} \cap K_{\nu,\varkappa}$$

Thus, the algorithm searches for points which belong to the parallel projection of the surface and lie within the grid square being investigated (INTER), i.e.

$$D_{u,v} \neq Q,\ Q = \text{ empty set.}$$

To keep computation time for this step as low as possible, each patch is approximated by a rectangle which contains it.(MINMAX).

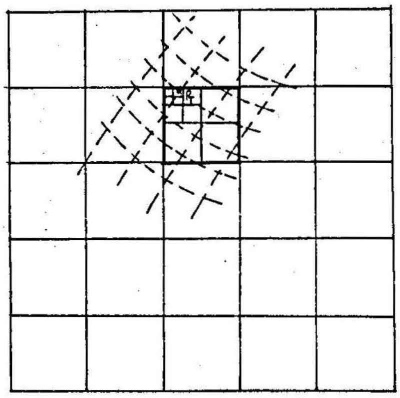


Furthermore, using an idea of Warnock, in the present implementation the grid square is subdivided into 4 parts (up to 3 times) if the number of Δu- Δv elements within the square is too large. The square is subdivided until

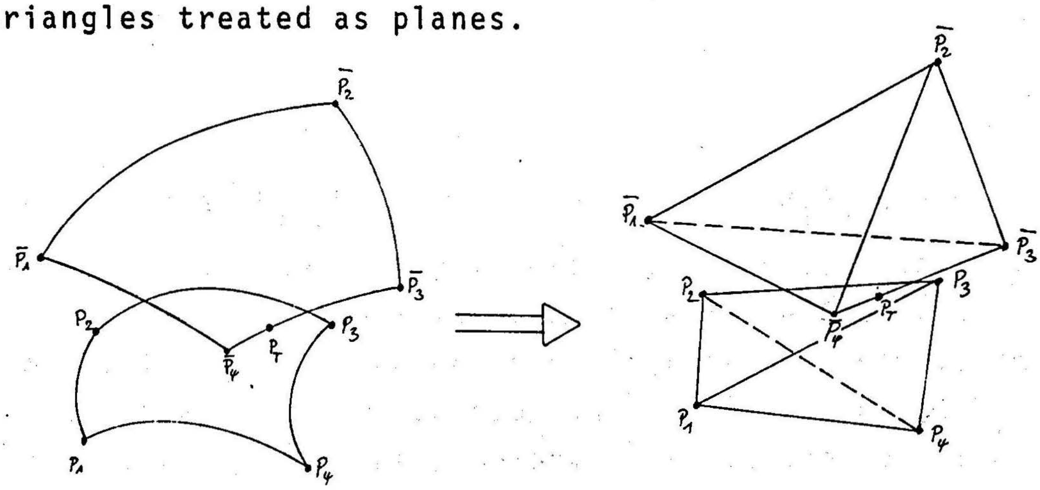
$$n_{\nu,\varkappa} \leqslant n_{max}$$

where $n_{\nu,\varkappa}$ is the number of Δu- Δv elements in a square and n_{max} is a fixed maximum.

n_{max} is determined by the storage space available and is generally an estimated value. The computer time required is inversely proportional to the square of the number of grid squares originally given or obtained by subdivision (see chap. 4). Currently, for 100 patches, the time required is 4 - 5 minutes.



In the next step, the Δu-Δv elements (patches) under consideration are divided, in two ways, into two parts, approximated by triangles treated as planes.

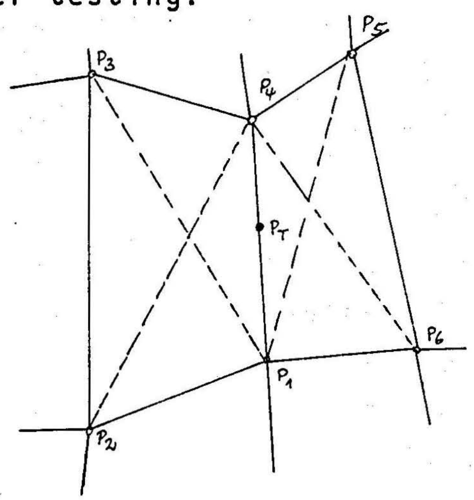


Only those triangles containing the test point P_T are considered further. They are found by the subroutine SEARCH. Whether or not a test point P_T lies within a patch or a part of a patch is determined in the fast assembler program INFLAS using a very simple algorithm.

Before INFLAS is called, trivial cases are treated, i.e.:

If all z-coordinates of a patch are greater than the z-coordinate of P_T, P_T is not visible. If all z-coordinates of a patch are smaller than the z-coordinate of P_T, the patch or the respective sub-triangle is irrelevant for further testing.

Another trivial case can be dealt with easily in SEARCH with the aid of INFLAS:
If a test point P_T is on the boundary or at an end point of a patch, the directly adjoining subdomains of this patch cannot cover the test point. Therefore, these sub-triangles are irrelevant for further testing.



The triangles $P_1P_3P_4$, $P_1P_2P_4$, $P_1P_4P_5$ and $P_1P_4P_6$ cannot cover P_T.
As this case occurs very frequently - the test point is moved along a u- or v-line - it means a decisive reduction in computing time. Only in cases which do not permit such simple decisions, are the plane projection-coordinates of test point $P_T(x,y,z)$ taken. The z-coordinates belonging to it are computed from the plane equations of the triangles:

$$\begin{vmatrix} x1 & x2 & x3 & x \\ y1 & y2 & y3 & y \\ z1 & z2 & z3 & z \\ 1 & 1 & 1 & 1 \end{vmatrix} = 0 \qquad (1)$$

In (1) z is set equal to zv and the determinant is resolved to zv.

A test point $P_T(x,y,z)$ is visible if

$$z - zv \geqslant \emptyset \qquad (2)$$

This is determined in the subroutine DIF. Before, however, the plane determinant (equ. 1) is solved, trivial cases are excluded or solved by simple IF-statements in DIF. If, for example, all z-coordinates of the end point of a triangle are larger (smaller) than the z-coordinates of test point P_T, the test point is invisible (visible). This means a considerable reduction of computing time.

For the visibility testing, the test points P_T are taken along the u- or v-lines. The distance between the test points is not given but defined by the distance between the nodes that are to be connected; i.e. if two nodes are far apart, more test points are taken. The intermediate points (test points) are taken sequentially along a u- or v-line. During this testing, various trivial cases are excluded to optimize computing time (SEARCH and DIF).

The filling of the display-file is also made optimally, i.e. dummy points are only filled in if the end point of the last vector is more than EPS = 10^{-5} away from the new position.

The visibility procedure is generally valid for both convex and concave patches. There are no geometric restrictions. The procedure, moreover, allows the processing of several general surfaces (at present 10 surfaces with 100 nodes each).

In the figures, the intersection lines were drawn in. In its actual form, FLAVIS does not display these curves. The points forming the curves are computed in FLAVIS, but they are lost. It is in principle no problem to save these points and to display the intersection lines using them. We are working on a program that adds the intersection lines, however, into the display-file.

3. Program descriptions:

In this chapter the programs

FLAVIS,
MINMAX,
ASSIGN,
VIDRAW,
GREVIS,
VISKRI,
DIF,
ERROR,
INTER, and
ZERLEG are described.

Moreover, the subroutine-package INFLAS consisting of the programs

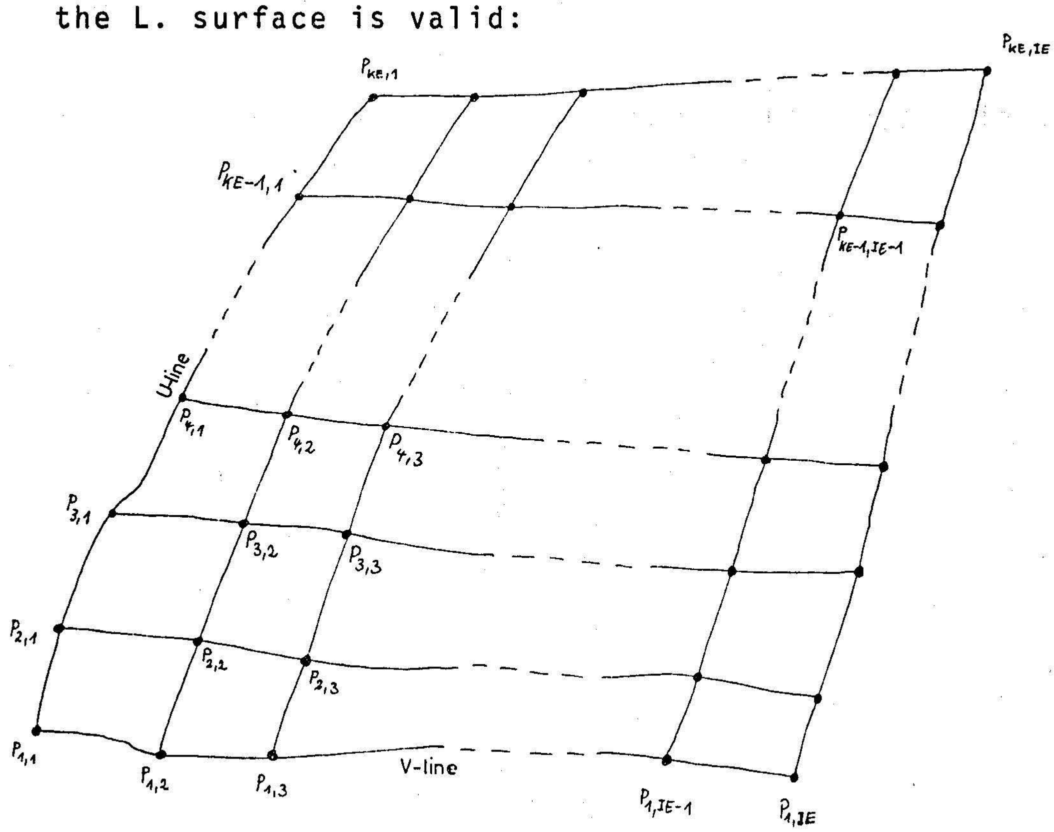
INFLIN,
INFLAP,
INFLA3,and
INFLA4 (in the assembler language META 92∅) is described.

General remarks, explanations of the COMMON-sizes and the arrays etc. are made in chapter 6.

Before a FLAVIS-call in the main program is made, the following sizes have to be defined:

size	significance	purposes
XMIN XMAX YMIN YMAX	minimum x-coordinate maximum x-coordinate minimum y-coordinate maximum y-coordinate	display scalation
IE KE LEMAX	number of U-lines number of V-lines number of surfaces	end indices for DO-loops
XT(I,K,L) YT(I,K,L) ZT(I,K,L)	coordinate fields of surface knots	

In the present form of the program up to 10 surfaces with 100 knots at a time are representable. (10 knots in U-direction and 10 knots in V-direction). For the indexing of the knot coordinates the L. surface is valid:



Here we have (L = surface index)

$$P_{1,1} = P\,[XT\,(1,1,L),\ YT\,(1,1,L),\ ZT\,(1,1,L)]$$
$$P_{2,1} = P\,[XT\,(2,1,L),\ YT\,(2,1,L),\ ZT\,(2,1,L)]$$
$$\vdots$$
$$P_{IE-1,1} = P\,[XT\,(IE-1,1,L),\ YT(IE-1,1,L),\ ZT\,(IE-1,1,L)]$$
$$P_{IE,1} = P\,[XT\,(IE,1,L),\ YT\,(IE,1,L),\ ZT\,(IE,1,L)]$$

$$P_{1,2} = P\,[XT\,(1,2,L),\ YT\,(1,2,L),\ ZT\,(1,2,L)]$$
$$\vdots$$
$$P_{1,KE-1} = P\,[XT\,(1,KE-1,L),\ YT\,(1,KE-1,L),\ ZT\,(1,KE-1,L)]$$
$$P_{1,KE} = P\,[XT\,(1,KE,L),\ YT\,(1,KE,L),\ ZT\,(1,KE,L)]$$

For the coordinates has to be hold

$$XMIN \leqslant XT\,(I,K,L) \leqslant XMAX$$
$$YMIN \leqslant YT\,(I,K,L) \leqslant YMAX.$$

In order to loose no picture parts after a possible rotation, for the z-coordinates has to be hold

$$XMIN \leqslant ZT\,(I,K,L) \leqslant XMAX \quad \text{and}$$
$$YMIN \leqslant ZT\,(I,K,L) \leqslant YMAX.$$

Furthermore it holds that

$$IE \leqslant IEMAX = 1\emptyset$$
$$KE \leqslant KEMAX = 1\emptyset$$
$$LEMAX \leqslant 1\emptyset$$

3.1 FLAVIS

Name:	FLAVIS
Key-word:	Surface with visibility
Language:	FORTRAN II
Call :	CALL FLAVIS
Parameter:	-
COMMON:	IN, I, K, L, IE, KE, LEMAX, XMIN, YMIN, XMAX, YMAX, DELTAX, DELTAY, TEILX, TEILY, SIGMAX, SIGMAY, THETAX, THETAY, DIVIS, IOPT1, IOPT2, IOPT3, LIHIWI, INDLI, NPUF, KEMAX, NANZAL
Subroutines:	MINMAX, ASSIGN, VIDRAW, Display-Software, INFLIN
Storage location:	534

---

1. part: to compute constants and to restore arrays

For an acceleration of the visibility testing, a cartesian screen is computed, as then the testings only refer to one part of the screen and no longer to the entire region of existence. In the main program the minimum- and maximum-values of the surfaces have to be computed in X- and Y-direction XMIN, YMIN, XMAX, YMAX. The sizes DIFFX, DIFFY result from them, with the aid of which other minimum- and maximum-values for the cartesian screen XMIN, XMAX, YMIN, YMAX are computed.

These new minimum- and maximum-values result in a cartesian screen, which is a little greater than the surface. Thus it is avoided that surface points lie on the screen border (zero interrogation of floating point sizes). The distance between two screen lines - there are 12 of them in X- and 12 in Y-direction - is computed in X-direction DELTAX and in Y-direction DELTAY.

For a refinement of the screen, which is made in the subroutines VISKRI and ZERLEG, 1/2, 1/4,1/8 of the distance in X- and Y-direction is computed.

In the subroutine GREVIS the size DIVIS and in the subroutine VISKRI the sizes IOPT1, IOPT2, IOPT3 are used, to which values are allocated. The arrays INDLI (1024) and LIHIWI (11,11,2) are set to zero, to obtain unobjectionable conditions for each new visibility testing.

With CALL INFLIN(XC(1),YC(1),IN) the addresses of the X-, Y-ARRAYS and the address of IN for the subroutines INFLAP, INFLA3,and INFLA4 are transferred (see description of INFLAS).

<u>2. part: allocation of the patches to the screen</u>

In a DO-loop the allocation of the patches in the screen is made:

First a branch-parameter ISPRU = ISPR-1 is set, here ISPR is the DO-loop of the outermost loop. For ISPRU = Ø in a triplicate index for each patch the minimum and maximum X- and Y-value XMI, YMI, XMA, YMA is computed. Besides, the subroutine MINMAX is called, that yields XMI, YMI, XMA, YMA as output parameters. For the knot coordinates of the surface the one-dimensional representation with the index IQ is used.

In subroutine MINMAX it is called for the knot coordinates XC, YC with the indices (IQ), (IQ+1), (IQ+KEMAX+1), (IQ+KEMAX). These indices correspond in the XT- YT-representation to (I,K,L), (I+1,K,L), (I+1, K+1, L), (I,K+1, L). In the DO-loop I only runs to IE(L)-1, K only to KE(L)-1. It would be senseless, to let I and K run to IE(L) resp. KE(L), as then, in subroutine MINMAX, for example, it would be called for a knot with the indices (IE(L)+1, KE(L)+1, L), which is not at all defined in the main program.

The variables XMI, YMI, XMA, YMA form a triangle, containing the patch. Thus the patch is enlarged and it is safe to say that all points have been collected. Required are the screen patches containing the points of the triangle XMI, YMI, XMA, YMA. As both, screen and triangle have axially parallel boundaries, the problem is considerably simplified. The variables IRA, IRE, KRA, KRE are formed, which indicate, over which screen patches the triangle is extended.

IRA and IRE are beginning- and end-value in X-direction for IRX; KRA and KRE are beginning- and end-value in Y-direction for IRY. IRX, IRY, which specify the screen patch, compute in a double DO-loop the number of patches per screen patch, which are stored in the indication array LIHIWI(IRY,IRX,1). At the end of each DO-loop, in which K is varied, by an increase of IQ for (KEMAX+1 - KE(L)), and at the end of each DO-loop, in which I is varied by an increase of IQ for (KEMAX * (IEMAX+1 - IE(L))) those storage locations that do not correspond to any knot-coordinates, are skipped.

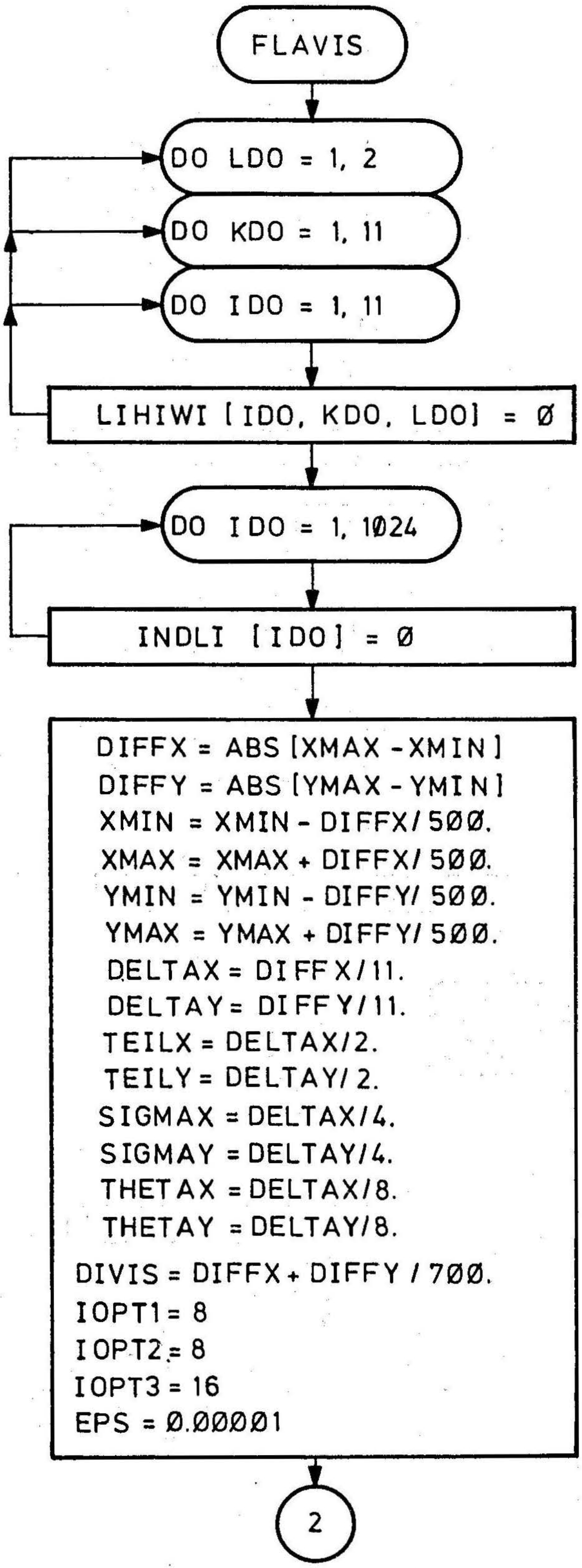
The number of patches in all screen patches is summed up and stored in array LIHIWI(IRY,IRX,2). At the same time the array LIHIWI(IRY,IRX,1), the values of which were taken over by the array LIHIWI(IRY,IRX,2), is set to zero. This is necessary, as the array LIHIWI(IRY,IRX,1) has to be built up once more in the 2nd run through the outermost loop (ISPRU=1).

In the second run through the outermost loop the program passes as in the first run the program part up to the generation of the indication array. Here too it is computed for each patch over which screen patches it is extended and this is stored in the indication array LIHIWI(IRY,IRX,1). With these values and the LIHIWI(IRY,IRX,2) computed in the first call, for each patch having points in a screen patch, a patch-counter IANZ is computed.

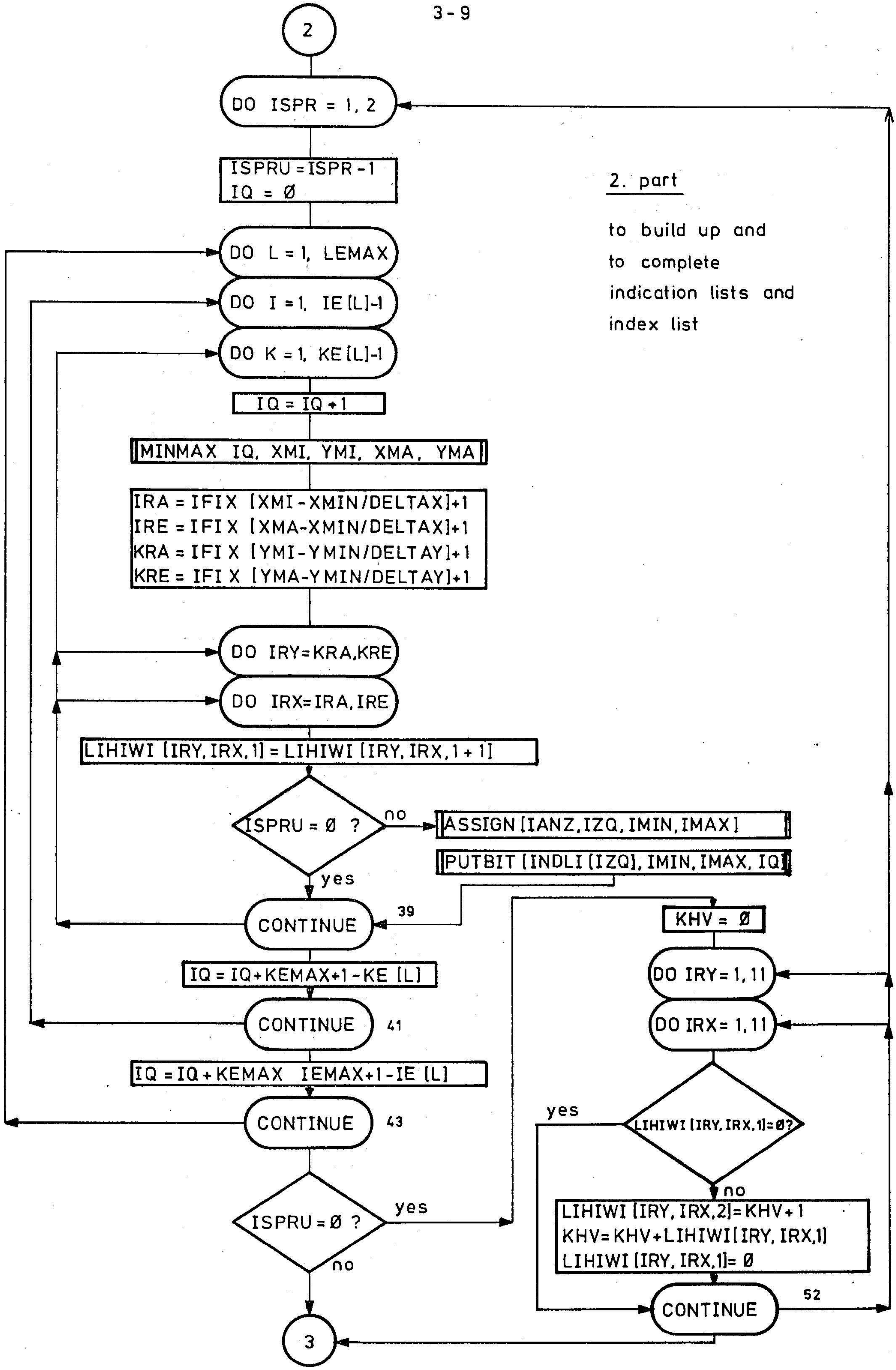
IANZ, as an input parameter is transferred to the subroutine ASSIGN, that yields IZQ, IMIN, IMAX as output parameters. With the system dependent program PUTBIT the patch-index IQ is stored in the storage unit INDLI(IZQ) of the index list INDLI between bits IMIN and IMAX. At the end of the DO-loops, the IQ-values, which are no indices of knot-coordinates are skipped.

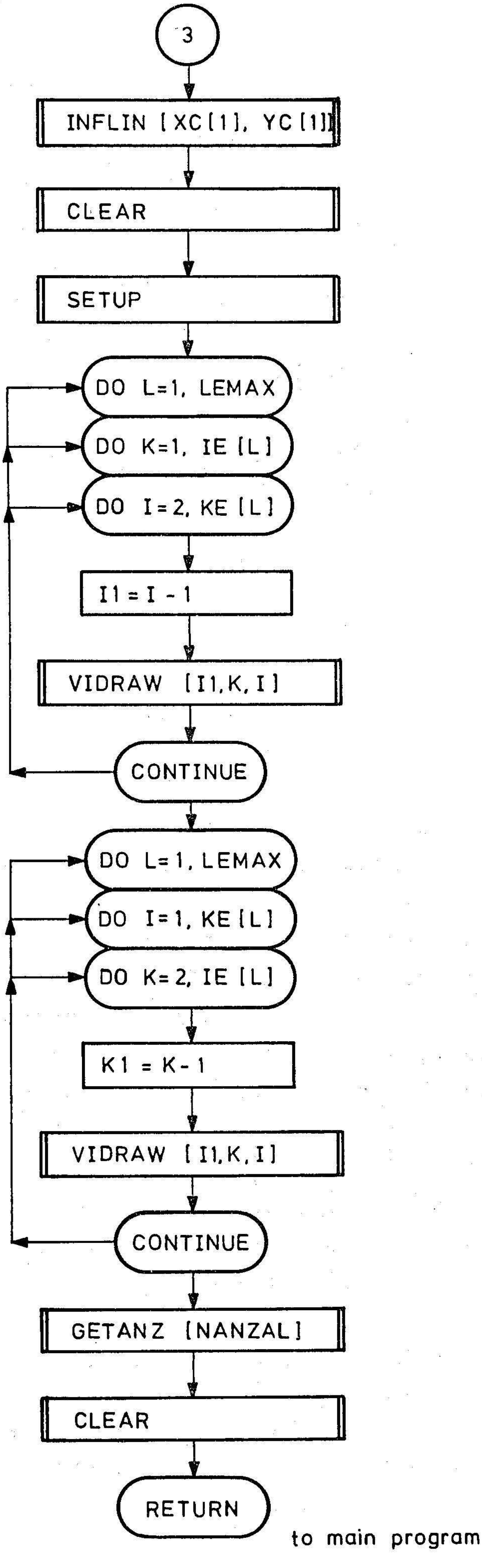
3. part: Drawing of the visible picture elements

Because of the small storage space, no dynamic picture output is possible, so that the existing buffer storage has to be erased by CLEAR and a new display-picture has to be prepared by call of the subroutine SETUP. The drawing of the visible line elements is made by the subroutine VIDRAW. First the visible U-line elements are drawn, for this the variables I1 = I-1 and K1 = K are set, afterwards the visible V-line elements are drawn, I1 = I and K1 = K-1.



<u>1. part</u>
to restore lists
to compute constants


2
DO ISPR = 1, 2
ISPRU = ISPR - 1
IQ = Ø
2. part
to build up and
to complete
indication lists and
index list
DO L = 1, LEMAX
DO I = 1, IE [L]-1
DO K = 1, KE [L]-1
IQ = IQ + 1
MINMAX IQ, XMI, YMI, XMA, YMA
IRA = IFIX [XMI-XMIN/DELTAX]+1
IRE = IFIX [XMA-XMIN/DELTAX]+1
KRA = IFIX [YMI-YMIN/DELTAY]+1
KRE = IFIX [YMA-YMIN/DELTAY]+1
DO IRY=KRA,KRE
DO IRX=IRA,IRE
LIHIWI [IRY, IRX, 1] = LIHIWI [IRY, IRX, 1 + 1]
ISPRU = Ø ?
no
ASSIGN [IANZ, IZQ, IMIN, IMAX]
PUTBIT [INDLI [IZQ], IMIN, IMAX, IQ]
yes
CONTINUE
39
IQ = IQ+KEMAX+1-KE [L]
CONTINUE
41
IQ = IQ + KEMAX IEMAX+1-IE [L]
CONTINUE
43
ISPRU = Ø ?
yes
no
3
KHV = Ø
DO IRY = 1, 11
DO IRX = 1, 11
LIHIWI [IRY, IRX, 1]= Ø?
yes
no
LIHIWI [IRY, IRX, 2]= KHV + 1
KHV= KHV+LIHIWI [IRY, IRX, 1]
LIHIWI [IRY, IRX, 1]= Ø
CONTINUE
52



3. part

drawing of the visible picture elements

3.2 MINMAX

Name:	MINMAX
Key-word:	Limits of a patch
Language:	FORTRAN II
Call:	CALL MINMAX (IMM, XMI, YMI, XMA, YMA)
Parameters:	Input parameter: IMM Output parameter: XMI, YMI, XMA, YMA
COMMON:	KEMAX, XC, YC
Subroutines:	-
Storage location:	116

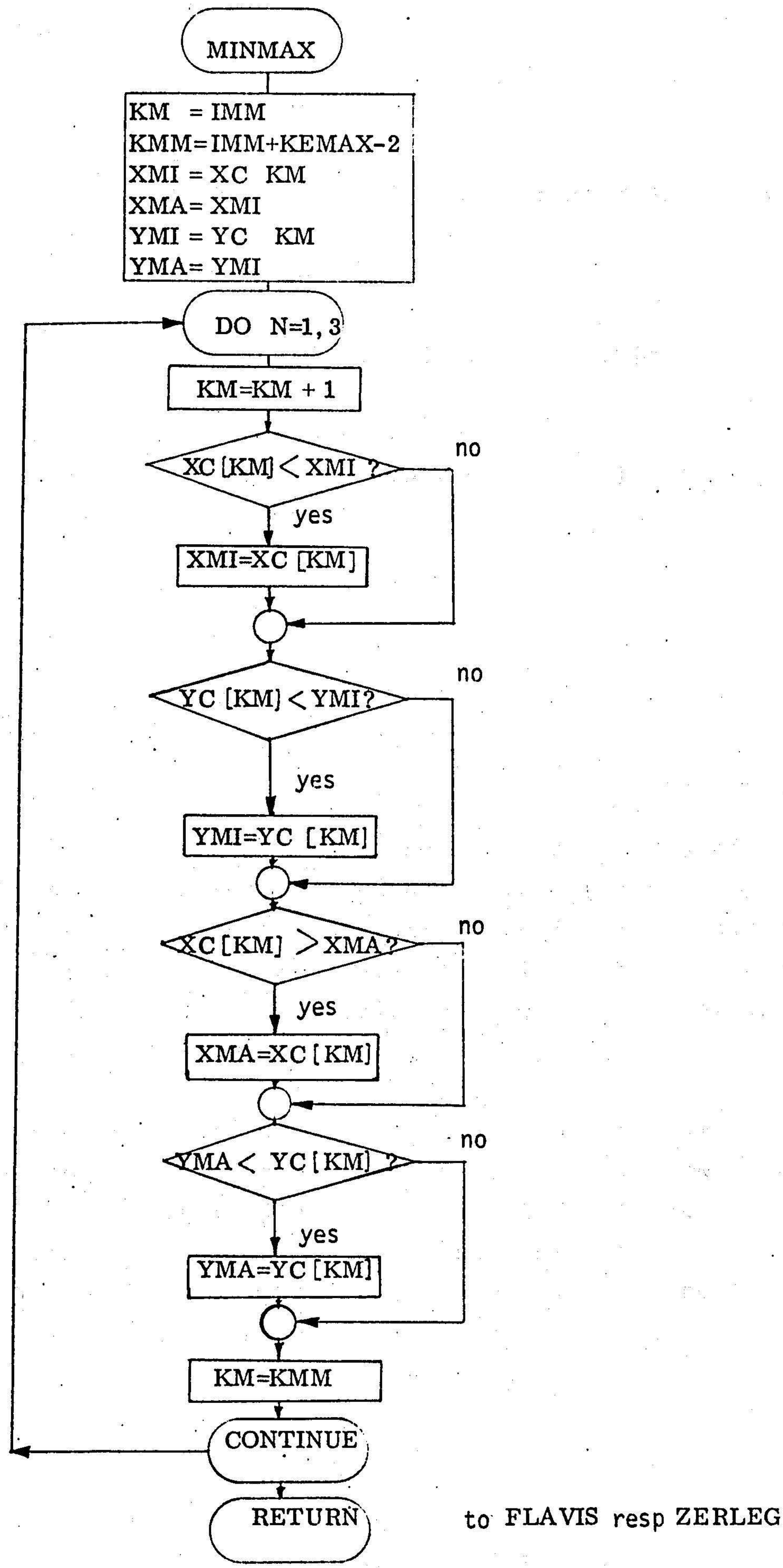
To save storage space and to reduce computing time in MINMAX the one-dimensional representation for the knots is used.

As the input parameter IMM may not be changed it is transferred to the local size KM. Then, for the DO-loop a size KMM=IMM+KEMA-2 is computed. Before the DO-loop

XMI = XC(KM) (Minimum in X-direction),
XMA = XMI (Maximum in X-direction),
YMI = YC(KM) (Minimum in Y-direction), and
YMA = YMI (Maximum in Y-direction) is set.

In a DO-loop, the previous minima and maxima are compared with the three other corner-coordinates and after respective branches the new minima and maxima are allocated to the parameters XMI,XMA, YMI, YMA.

In the first run the index for XC resp. YC is equal IMM+1. At the end of the DO-loop KM = IMM + KEMAX-2 = KMM is set. Thus, in the 2nd run the index gets the value IMM + KEMAX. For the 3rd run KM = IMM + KEMAX-2 is set again and in the 3rd run the index has the value IMM + KEMAX+1.

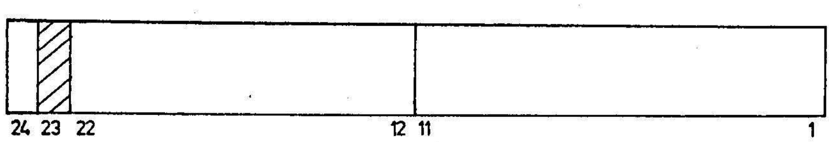

MINMAX
KM = IMM
KMM=IMM+KEMAX-2
XMI = XC KM
XMA= XMI
YMI = YC KM
YMA= YMI
DO N=1, 3
KM=KM + 1
XC [KM] < XMI ?
no
yes
XMI=XC [KM]
YC [KM] < YMI?
no
yes
YMI=YC [KM]
XC [KM] > XMA ?
no
yes
XMA=XC [KM]
YMA < YC [KM] ?
no
yes
YMA=YC [KM]
KM=KMM
CONTINUE
RETURN
to FLAVIS resp ZERLEG

3.3 ASSIGN

Name:	ASSIGN
Key-word:	Assignment for system dependent programs
Language:	FORTRAN II
Call:	CALL ASSIGN (IANZ, IZQ, IMIN, IMAX)
Parameters:	Input parameter: IANZ Output parameter: IZQ, IMIN, IMAX
COMMON:	-
Subroutines:	-
Storage locations:	59

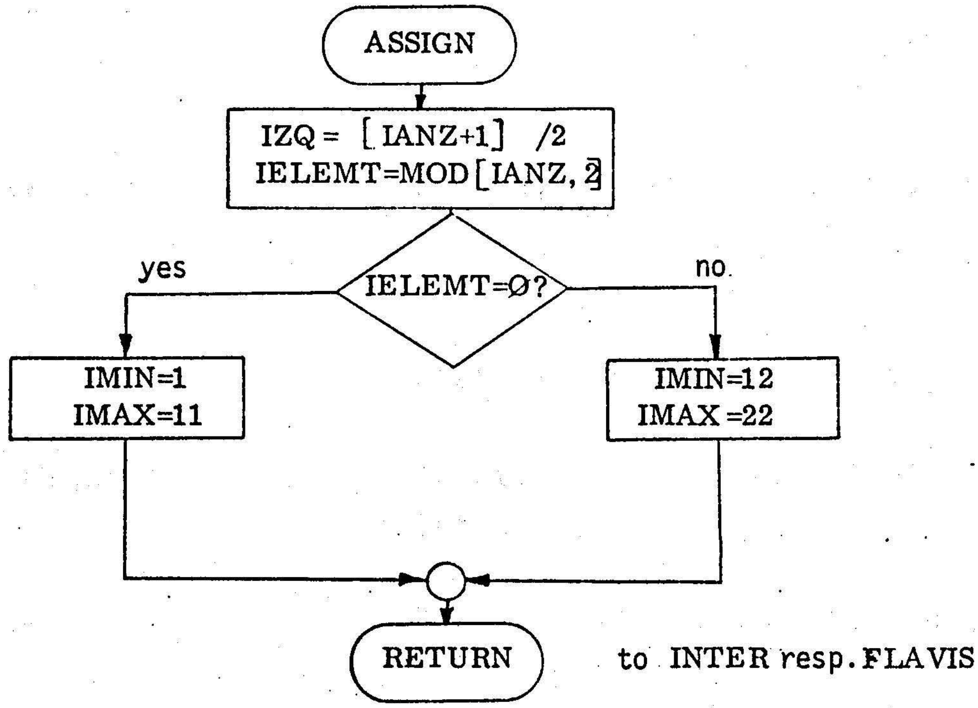
The input parameter IANZ is a measure for ascertaining how many patches are allocated to how many screen patches. A more exact computation see in subroutine FLAVIS. From IANZ the variable IZQ is formed in a way that always two successive values of IANZ result in a value of IZQ. IZQ is the index of the index list INDLI.

It is the meaning of the subroutine ASSIGN, to compute the bits for the storage locations of the index list in a way that always two indices IQ can be loaded into a storage cell INDLI(IZQ). IMIN is the first bit, IMAX the last, the index IQ may take. If the value of IANZ is an odd number, IMIN=1 and IMAX=11 is set, if it is an even number IMIN=12 and IMAX=22 is set.



In the partition of the storage cell according to this method, 22 of the 24 available bits are used, the 23. remains uncovered and the 24. bit as sign bit is not to be covered. The representation with two indices I,K can only cover 20 bits of the 24 existing, therefore the procuct of two indices I,K can be maximally half as great as the index in the one-dimensional representation. Hence it follows that when applying two indices, the index list can only compute half of the knots, which could be computed with one index.

In the computation with one index IQ its value and thus the number of patches, which can be computed by the index list, can be doubled once more, if, after a suitable interrogation, 11 bits of a storage cell are allocated to the indices IQ smaller than 2^{11}, and if 12 bits are allocated to the indices IQ greater than 2^{11} but smaller than 2^{12}. With 2^{12}=4096 knots here too the limit is reached up to which the indices IQ can be loaded into the index list.


ASSIGN
IZQ = [IANZ+1] /2
IELEMT=MOD[IANZ, 2]
IELEMT=Ø?
yes
no
IMIN=1
IMAX=11
IMIN=12
IMAX =22
RETURN
to INTER resp. FLAVIS

3.4 VIDRAW

Name:	VIDRAW
Key-word:	Drawing of the visible picture elements
Language:	FORTRAN II
Call:	CALL VIDRAW (I1, K1, IKANF)
Parameters:	Input parameter: I1, K1, IKANF Output parameter: -
COMMON:	EPS, I, III, K, KZZ, L, NPUF, XT, YT, ZT
Subroutines:	VISKRI, GREVIS, Display-Software
Storage location:	253

First the beginning- resp. end-point of a line is exchanged out of the three-dimensional ARRAY. Here it is hold:

If I1 = I-1 and K1 = K, U-lines are tested for visibility.
If I1 = 1 and K1 = K-1, V-lines are tested for visibility.

The 1. point is only tested at the beginning of a U- or V-line, resp. if the subroutine GREVIS was left previously (IVK = -1, because beginning- and end-point lie closely together). The creterion, if a beginning point is concerned, is yield by the parameter IKANF (IKANF=2). It is only necessary to test the beginning point of the respective U- or V-line for visibility, as the subroutine GREVIS anyway computes the visibility of the end point of a line element. At the next line element this end-point becomes the new beginning point. If the first point is invisible, the subroutine GREVIS is called by the input parameter IWAHL = -1, if it is visible, IWAHL has the value Ø.

With each call of GREVIS, XD1, YD1, ZD1, and XD2, YD2, ZD2 are transferred as input parameters. The output parameters XSTR, YSTR, ZSTR are the X-, Y-, Z-coordinates of the point, which marks the boundary of the visible line element. This can be a point up to which the line element is visible, or a point from which off the line element becomes visible.

<u>1. case:</u> If IWAHL is -1, after the call of GREVIS it is queried, whether the beginning- and end-point of the line element in question lie within an EPSILON-distance. If this is realized, the variable III is greater than Ø and it is returned at once into the calling program FLAVIS. If III is not greater Ø, the X-, Y-, Z-coordinates of boundary point XSTR, YSTR, ZSTR are regarded as coordinates XD1, YD1, ZD1 of a new beginning point.

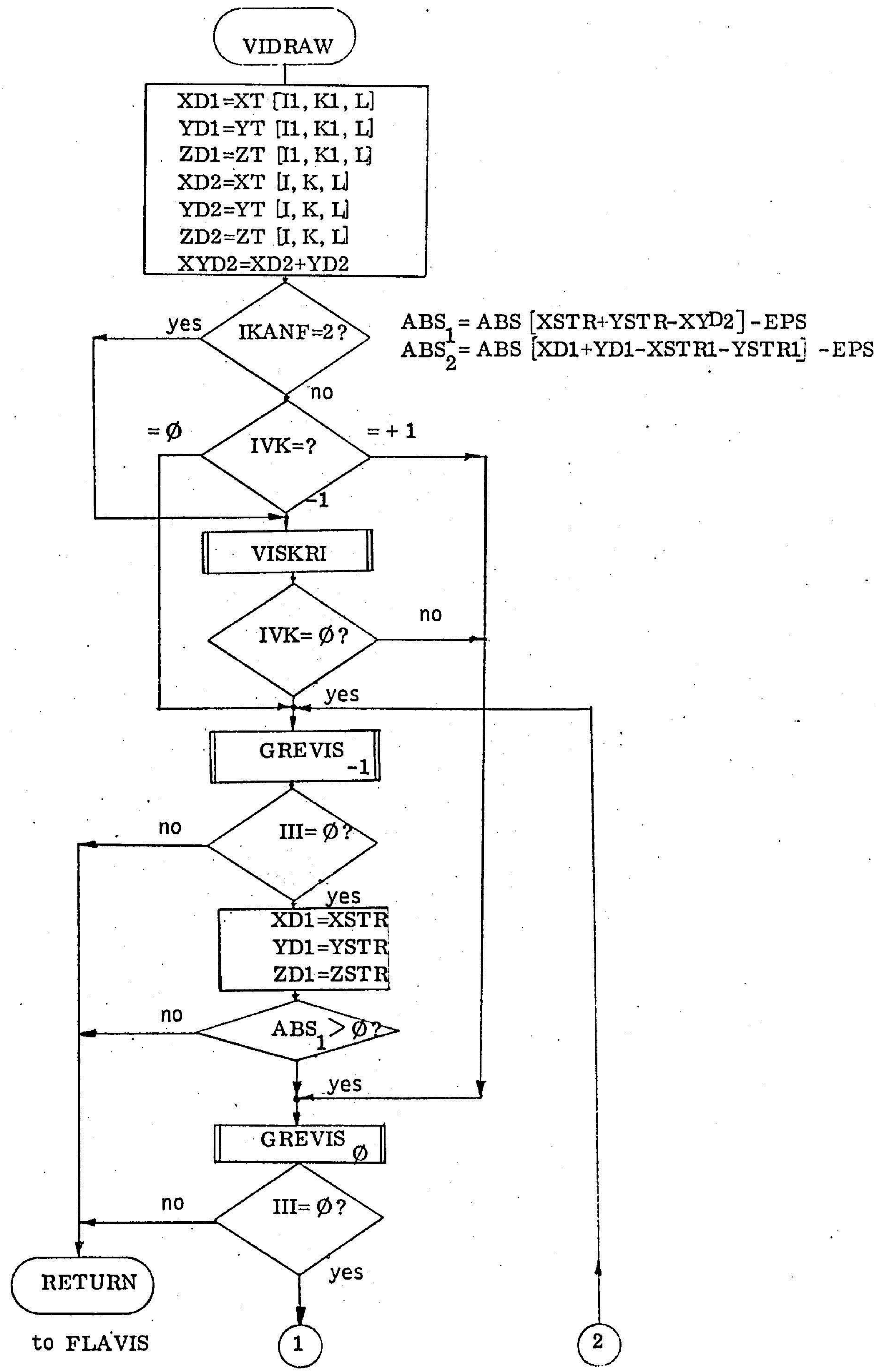
If the boundary point of visibility coincides with the second knot, it is returned into the calling program FLAVIs, as the entire connection-line has been tested for visibility. If it does not coincide with it, it is proceeded according to case 2.

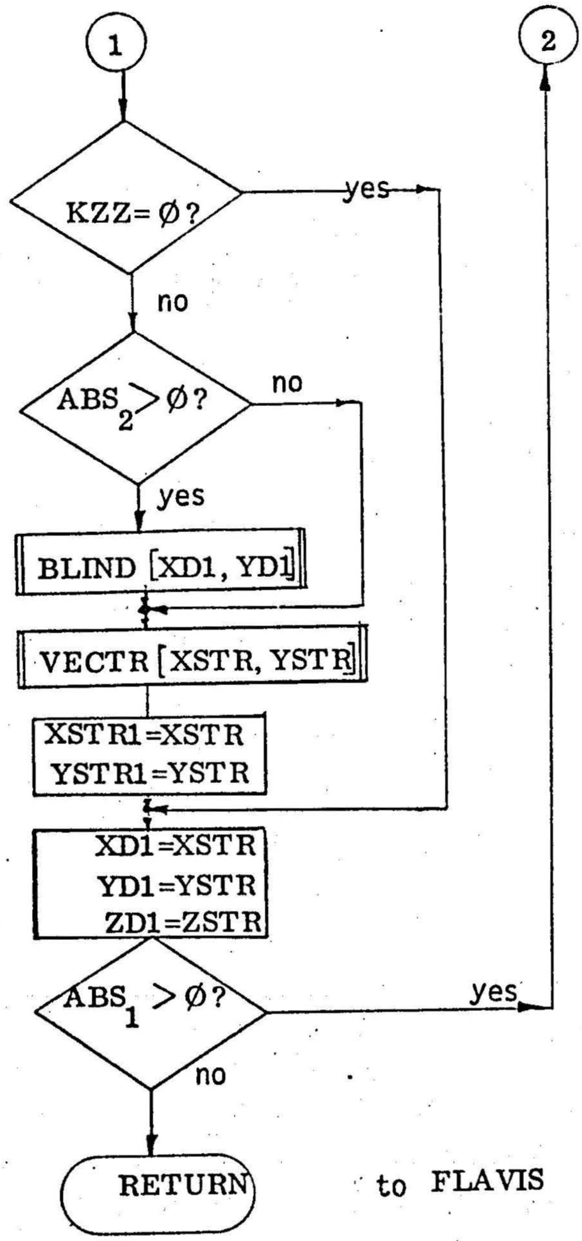
<u>2. case:</u> If IWAHL is Ø, after the return from subroutine GREVIS, it is queried - as described above - whether beginning- and end-point lie within an EPSILON-distance.
In subroutine GREVIS a loop counter KZZ, beginning from Ø, is encreased by one at each step from the beginning- to the end point. If KZZ at the return is Ø, that means if only one step away from the beginning point the line element is visible, the part in between should not be drawn. In the program CALL BLIND and CALL VECTR are skipped then.

To keep the buffer storage NPUF as little as possible, the subroutine BLIND should be called only, if the new vector is not drawn from the end point of the last vector.

It is queried , whether the beginning point XD1, YD1 of the new vector coincides with the end point XSTR1, YSTR1 of the last vector drawn. If this is not the case, first BLIND is called and then VECTR, otherwise VECTR is called directly, for the subroutine VECTR has the characteristic to begin the vector always at the end point of the last drawn vector.

CALL VECTR(XSTR, YSTR) is always followed by a CALL SHOW(Ø), to ascertain after each drawn vector possibly occurring errors. The end point XSTR, YSTR of the drawn vector is stored in XSTR1, YSTR1, in order to be available for a later comparison with the beginning point of the next vector. (see above). The coordinates of the beginning point XD1, YD1, ZD1 get the values of the boundary point of visibility XSTR, YSTR, ZSTR. If the boundary point conincides with the second knot, it is returned into the calling program FLAVIS; if this is not the case it is proceeded according to case 1.


VIDRAW
XD1=XT [I1, K1, L]
YD1=YT [I1, K1, L]
ZD1=ZT [I1, K1, L]
XD2=XT [I, K, L]
YD2=YT [I, K, L]
ZD2=ZT [I, K, L]
XYD2=XD2+YD2
IKANF=2?
yes
no
ABS1 = ABS [XSTR+YSTR-XYD2] - EPS
ABS2 = ABS [XD1+YD1-XSTR1-YSTR1] - EPS
IVK=?
= Ø
= + 1
-1
VISKRI
IVK= Ø ?
no
yes
GREVIS -1
III= Ø ?
no
yes
XD1=XSTR
YD1=YSTR
ZD1=ZSTR
ABS1 > Ø ?
no
yes
GREVIS Ø
III= Ø ?
no
yes
RETURN
to FLAVIS
1
2


1
2
KZZ= Ø ?
yes
no
ABS2 > Ø ?
no
yes
BLIND [XD1, YD1]
VECTR [XSTR, YSTR]
XSTR1=XSTR
YSTR1=YSTR
XD1=XSTR
YD1=YSTR
ZD1=ZSTR
ABS1 > Ø ?
yes
no
RETURN
to FLAVIS

3.5 GREVIS

Name:	GREVIS
Key-word:	limit of visibility
Language:	FORTRAN II
Call:	CALL GREVIS (IWAHL, IVK, XD1, YD1, ZD1, XD2, YD2, ZD2, XSTR, YSTR, ZSTR)
Parameters:	Input parameter: IWAHL, XD1, YD1, ZD1, XD2, YD2, ZD2 Output parameter:XSTR, YSTR, ZSTR, IVK
COMMON:	DIVIS, EPS, III, KZZ
Subroutines:	VISKRI
Storage location:	239

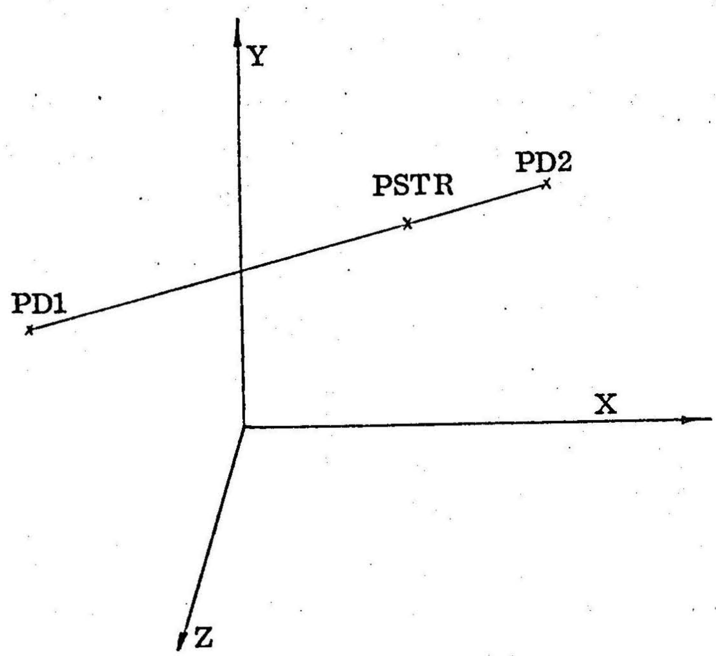
First the variables III = Ø and KZZ = Ø are set to have unique beginning conditions. Beginning point XD1, YD1, ZD1 and end point XD2, YD2, ZD2 of the connection line are transferred as input parameters. Its distance in all three coordinates is computed. The absolute value of the distance in X- and Y-direction is formed, whereby the sizes XABS and YABS are produced. Both are added and thus result in the variable BETRAG.
Now it is queried, whether BETRAG is less than or equal to EPS = 10^{-5}, that means beginning- and end point lie in the epsilon-criterion. If that is the case, III = 1 and IVK =-1 is set and it is branched to VIDRAW. IVK =-1 means for VIDRAW that a knot has to be once more tested for visibility. III = 1 causes in VIDRAW the return to FLAVIS.

If a line element is tested for visibility, the number of test points on it should correspond to its lenth.

Long line elements should have more test points than short ones. A measure for the line length and thus for the number of test points is the variable MBET which is directly proportional to size BETRAG and inversely proportional to variable DIVIS. The variable DIVIS was fixed in subroutine FLAVIS (see there). It is proportional to the absolute sizes of the surfaces in the X-, Y-plane.

The variable MBET, by which later has to be divided, may not have the value Ø. If, however, the calculation results in the value Ø, MBET = 1 is set.

The subroutine GREVIS is to compute the limiting point of visibility between a beginning point XD1,YD1,ZD1 and an end point XD2,YD2,ZD2. Beginning- and end point lie on the periphery of a patch. Because of the approximately infinitesimal character of the patch its peripheries may be regarded as straight lines. Therefore, for a computation of the test points the line equation in space between two defined points is sufficient.



$$\frac{XSTR-XD1}{XD2-XD1} = \frac{YSTR-YD1}{YD2-YD1} = \frac{ZSTR-ZD1}{ZD2-ZD1} \qquad (1)$$

$$XSTR = (YSTR-YD1)\frac{XD}{YD} + XD1 = (ZSTR-ZD1)\frac{XD}{YD} + XD1 \qquad (2)$$

$$YSTR = (XSTR-XD1)\frac{YD}{ZD} + YD1 = (ZSTR-ZD1)\frac{YD}{ZD} + YD1 \quad (3)$$

$$ZSTR = (YSTR-YD1)\frac{ZD}{YD} + ZD1 = (XSTR-XD1)\frac{ZD}{XD} + ZD1 \quad (4)$$

with $XD = XD2 - XD1$ (5)

$YD = YD2 - YD1$ (6)

$ZD = ZD2 - ZD1$ (7)

The drawing of the surfaces with and without visibility is made in the X-,Y-plane. To reach a higher accuracy, always this X- or Y-coordinate is changed,for which the distance between beginning- and end point is greater. The other two coordinates X,Z or Y,Z are computed by the line equation.

Now it is queried whether XABS > YABS. If XABS > YABS, the X-coordinate is changed, otherwise the Y-coordinate. This interrogation is necessary to avoid a division by zero at the computation of the line equation. The case XABS and YABS = Ø was already excluded by querying for the size of BETRAG.

Before the actual variation of the coordinates begins, constant sizes as auxiliary sizes are computed for the loop. These are for the variation of the Y-coordinate

$XYD = XD/YD$ ($= (XD2 - XD1) / (YD2 - YD1)$)

$ZYD = ZD/YD$ ($= (ZD2 - ZD1) / (YD2 - YD1)$)

$YBET = YD/BET$ ($= (YD2 - YD1) / FLOAT (MBET)$)

These are for the variation of the X-coordinate

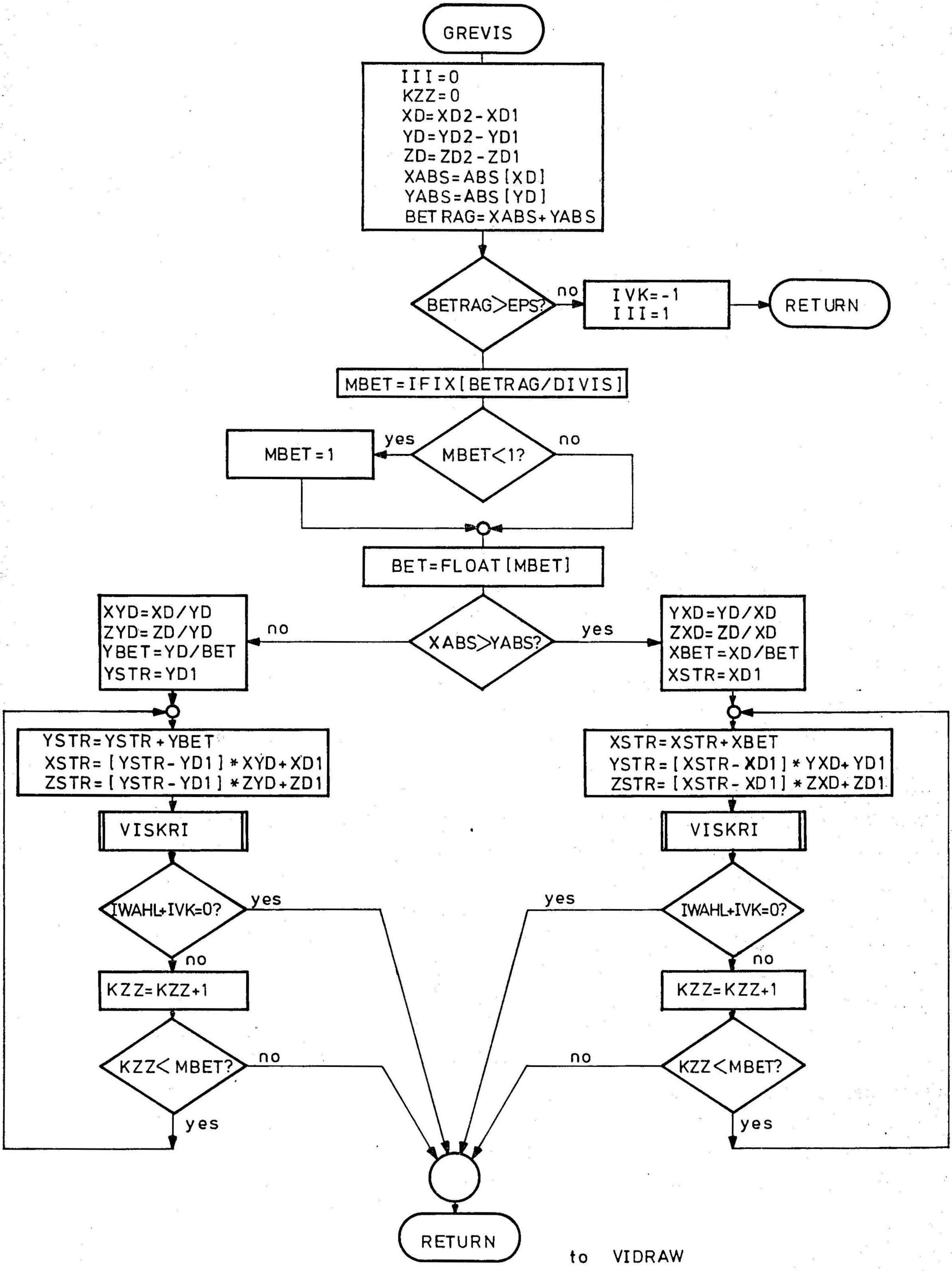
$YXD = YD/XD$ ($= (YD2 - YD1) / (XD2 - XD1)$)

$ZXD = ZD/XD$ ($= (ZD2 - ZD1) / (XD2 - XD1)$)

By these actions unnecessary computations are avoided.

Now the coordinates XSTR, YSTR, ZSTR are computed from the line equation and transferred to the subroutine VISKRI, the output parameter IVK of which indicates whether the test point is visible or not.
IVK = 1 means it is visible, IVK = Ø means it is invisible.
If it is visible and if IWAHL has the value -1, which means that the beginning point is invisible, the limit of visibility is reached, and it is returned into the calling program. If IWAHL has the value Ø, which means that the beginning point is visible, it is moved one step forward towards the end point.
If the test point is invisible and IWAHL has the value -1 it is moved one step forward towards the end point too. If IWAHL has the value Ø the limit of visibility is reached and it is returned into the calling program.

At each step along the connection line, the loop counter KZZ, beginning at Ø, is encreased by one. If KZZ reached the value MBET the test point coincides with the end point and it is returned into the calling program. If KZZ is less than MBET it is moved one step forward towards the end point.


GREVIS
III=0
KZZ=0
XD=XD2-XD1
YD=YD2-YD1
ZD=ZD2-ZD1
XABS=ABS[XD]
YABS=ABS[YD]
BETRAG=XABS+YABS
BETRAG>EPS?
no
IVK=-1
III=1
RETURN
MBET=IFIX[BETRAG/DIVIS]
MBET<1?
yes
no
MBET=1
BET=FLOAT[MBET]
XABS>YABS?
no
yes
XYD=XD/YD
ZYD=ZD/YD
YBET=YD/BET
YSTR=YD1
YXD=YD/XD
ZXD=ZD/XD
XBET=XD/BET
XSTR=XD1
YSTR=YSTR+YBET
XSTR=[YSTR-YD1]*XYD+XD1
ZSTR=[YSTR-YD1]*ZYD+ZD1
XSTR=XSTR+XBET
YSTR=[XSTR-XD1]*YXD+YD1
ZSTR=[XSTR-XD1]*ZXD+ZD1
VISKRI
VISKRI
IWAHL+IVK=0?
yes
no
IWAHL+IVK=0?
yes
no
KZZ=KZZ+1
KZZ=KZZ+1
KZZ<MBET?
no
yes
KZZ<MBET?
no
yes
RETURN

to VIDRAW

3.6 VISKRI

Name: VISKRI

Key-word: visibility creterion

Language: FORTRAN II

Call: CALL VISKRI (A, B, C, IVK)

Parameters: Input parameter: A, B, C
Output parameter: IVK

COMMON: KEMAX, XMIN, YMIN, DELTAX, DELTAY, TEILX, TEILY, SIGMAX, SIGMAY, THETAX, THETAY, IOPT1, IOPT2, IOPT3, IHQ, KWQ, LFQ, MBQ, ICA, LSC, MVC, NRC

Subroutines: INTER, ZERLEG, SEARCH, DIF

Storage location: 523

The input parameters A, B, C are the X-, Y-,Z-coordinates of the test point which is to be tested for visibility. It is fixed in which part of the screen it lies, that is regarding 1) DELTAX, DELTAY, 2) TEILX, TEILY, 3) SIGMAX, SIGMAY, 4) THETAX, THETAY. Thus the appertaining variables IRX, IRY, KDX, KDY, LPX, LPY, MGX, MGY arise, which mark the respective screen patch, 1/4 of the screen patch, 1/16 of the screen patch, 1/64 of the screen patch in the X-,Y-plane.

If the test point lies in the same screen patch, 1/4 screen patch, 1/16 screen patch, 1/64 screen patch,as the last point that was tested for visibility, it is not necessary to compute by a subroutine call which patches lie in the respective screen part, for this is still known from the last call.

The variables IRXA, IRYA, KDXA, KDYA, LPXA, LPYA, MGXA indicate the screen part in which the last tested test point is situated. It is queried succesively, whether IRX coincides with IRXA, IRY with IRYA, KDX with KDXA, KDY with KDYA, LPX with LPXA, LPY with LPYA, MGX with MGXA, MGY with MGYA.

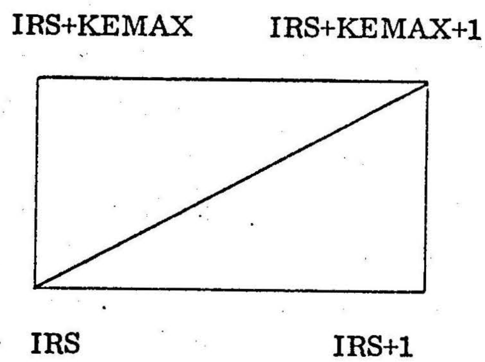
If this always is the case, the subroutine SEARCH can be called, that computes which partial triangles of patches contain the test point and excludes trivial cases. If there is no coincidence in the interrogations, it is branched to the respective subroutine that computes the patches for the fixed screen part.

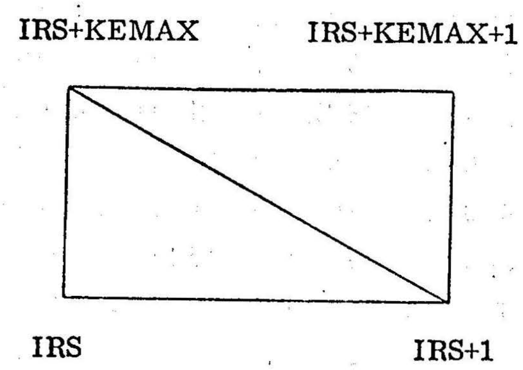
Supposed, already in the comparison with IRXA or IRYA there would be no coincidence, the subroutine INTER(IRY,IRX,NIHQ) would be called. NIHQ indicates the number of patches having point quantities in the screen patch fixed by IRY, IRX. The indices of the surface elements are stored in field IHQ. If number NIHQ is less than IOPT1, a quadruple division of the screen patch is not advantageous.

IOPT1, IOPT2, IOPT3 are optimal values, which are fixed in the subroutine FLAVIS in a way that with a number of patches in the screen part, which is less than the appertaining optimal value, a quadruple division of the screen part involves no reduction of computing time. If the number is greater than the optimal value, the division is made, because then the computing time can be reduced.

If no division is made, the subroutine SEARCH is called to which NIHQ, IHQ, A, B, C are transferred as input parameters. Output parameters are the sizes NICA, NLSC, NMVC, NNRC, ISPRU, which indicate the number of triangles containing the test point. Each patch being a quadrangle can be divided into triangles in two different ways.

In SEARCH even in trivial cases an unique decision about the visibility can be made. Then ISPRU = Ø (covering) resp. ISPRU = -1 (no covering possible) is set and after IVK was set accordingly it is returned into subroutine GREVIS. This causes for trivial cases a decisive reduction of computing time.





The following agreements are made:

Indices of corner points			number of triangles	Array of indices
IRS,	IRS+KEMAX,	IRS+KEMAX+1	NICA	ICA
IRS,	IRS+1,	IRS+KEMAX+1	NLSC	LSC
IRS+1,	IRS+KEMAX,	IRS+KEMAX+1	NMVC	MVC
IRS,	IRS+1,	IRS+KEMAX	NNRC	NRC

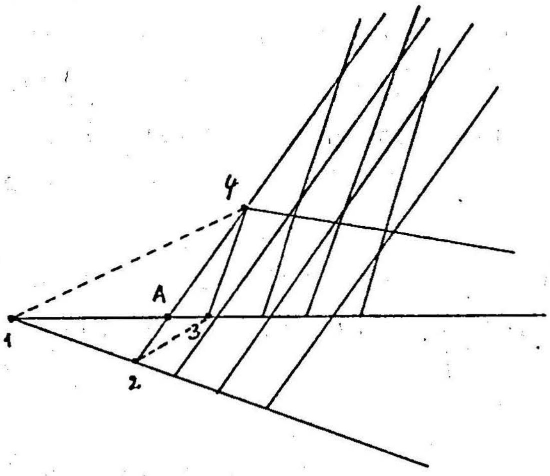
After the call of SEARCH the screen lines IRX, IRY are stored in IRXA, IRYA, for being available for the next call of subroutine VISKRI.
If number NIHQ is greater than IOPT1, the screen patch is divided into four parts. Thus the number of patches in the smaller screen parts becomes smaller and there is a reduction in computing time. In addition, the subroutine ZERLEG is called,

to which NIHQ, IHQ, TEILX, TEILY, KDX, KDY are transferred as input parameters. Output parameters are the sizes NKWQ and KWQ. In field KWQ the indices of those patches are stored, which lie in the 1/4 screen patch fixed by KDX, KDY,and the number of which is NKWQ.

After the call of ZERLEG it is computed, whether the value of the variable NKWQ is smaller than IOPT2. If that is the case no further division of the screen part is made. The subroutine SEARCH with the input parameters NKWQ, KWQ, A, B, C is called. Output parameters are again the variables NICA, NLSC, NMVC, NNRC, ISPRU. After the call,IRX, IRY, KDX, KDY are stored in equal sequence in IRXA, IRYA, KDXA, KDYA. If the variable NKWQ is greater than IOPT2, the subroutine ZERLEG with the input parameters NKWQ, KWQ, SIGMAX, SIGMAY, LPX, LPY and the output parameters NLFQ and LFQ is called. In field LFQ are the indices of patches, which lie in the 1/16 screen patch marked by LPX and LPY,and the number of which is NLFQ.

If the variable NLFQ is smaller than size IOPT3 no division of the 1/16 screen patch is made. The subroutine SEARCH with the input parameters NLFQ, LFQ, A, B, C and the output parameters NICA, NLSC, NMVC, NNRC, ISPRU is called. After the return, IRX, IRY, KDX, KDY, LPX, LPY are allocated in equal sequence to sizes IRXA, KDXA, KDYA, LPXA, LPYA. If the variable NLFQ is greater than IOPT3 the subroutine ZERLEG with the input parameters NLFQ, LFQ, THETAX, MGX, MGY and the output parameters NMBQ, MBQ is called. In field MBQ are the indices of the patches which lie in the 1/64 screen patch fixed by MGX, MGY, and the number of which is NMBQ.

After the third division the subroutine SEARCH with the input parameters NMBQ, MBQ, A, B, C and the output parameters NICA, NLSC, NWVC, NNRC, ISPRU is called. After the return, IRX, IRY, KDX, KDY, LPX, LPY, MGX, MGY are allocated in equal sequence to the variables IRXA, IRYA, KDXA, KDYA, LPXA, LPYA, MGXA, MGYA.



Before the actual visibility of the test point is computed, another consideration has to be made. We regard the opposite patch 1, 2, 3,4. Point 3 shall have a smaller Z-coordinate perpendicular to the drawing plane than point 2. Point 3 shall lie "behind" point 2. As you can see from the picture, then lines 1-2 and 2-4 are visible, line 3-4 is invisible. Line 1-3 is visible in section 1-A and invisible in section A-3.
As shown above, each four-cornered patch can be divided into triangles in two different ways. If patch 1,2,3,4 is divided into triangles 1,2,3 and 2,3,4, triangle 2,3,4 only covers section A-3. If it is divided into triangles 1,2,4 and 1,3,4, triangle 1,2,4 covers section 1-A, which is visible.

This shows that a point can be actually visible, if it is only covered by one triangle. It is invisible, if it is covered by at least two triangles. For this the point has to be contained in two triangles, what can be only satisfied, if each patch is divided into triangles in two different ways.

The output parameter IVK of subroutine VISKRI indicates whether a test point is visible or not. IVK = Ø means it is invisible, IVK = 1 means it is visible. In the program first IVK = 2 is set and at each covering of the test point by a triangle IVK is reduced by 1. If the test point is only covered once, the result is IVK = 1, and it is visible. If it is covered twice, the result is IVK = Ø, and it is invisible. Then it is returned into subroutine GREVIS.

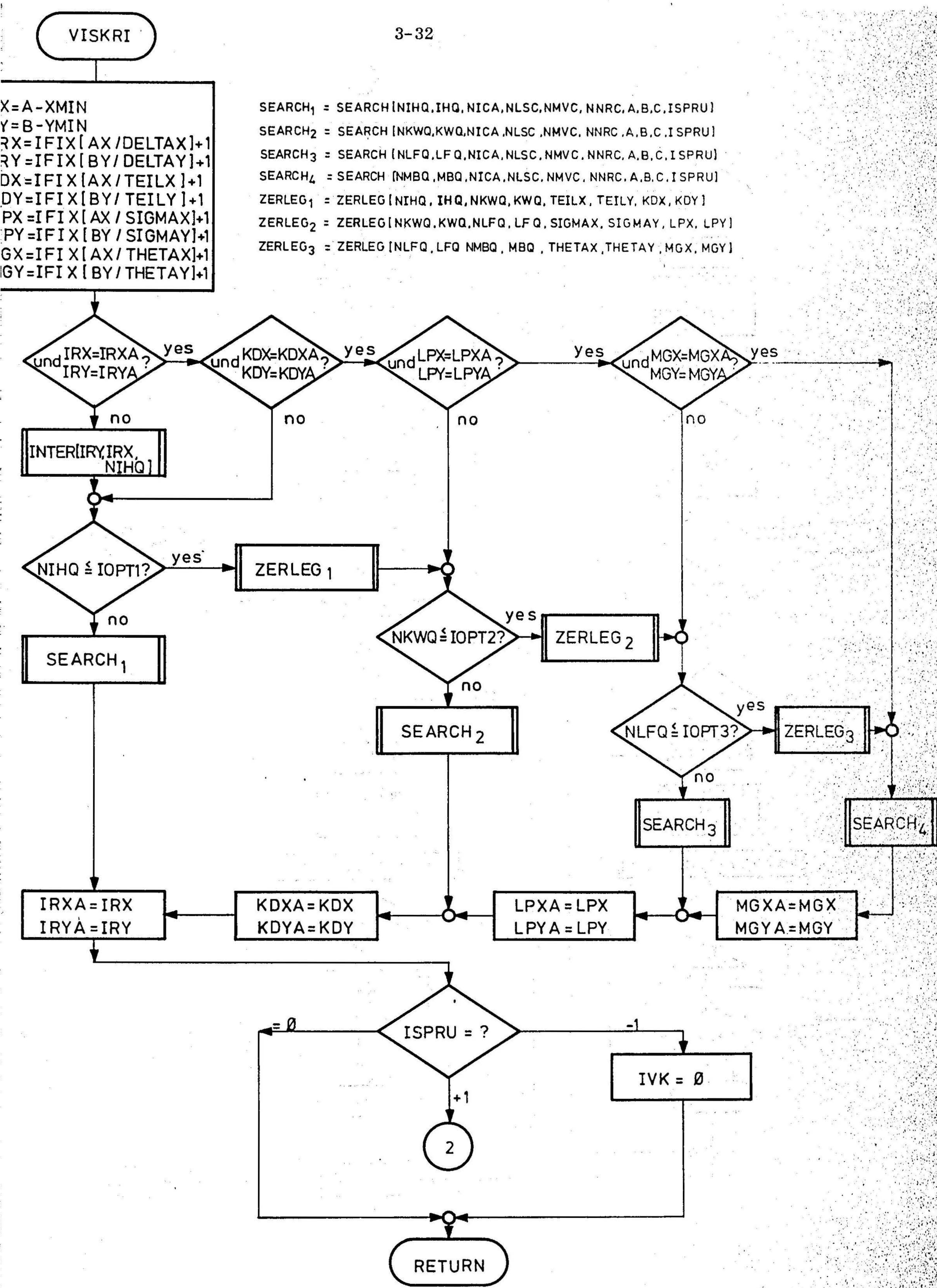
If number NICA of the triangles, the indices of which are in array ICA, = ∅, it is interrogated next, whether number NLSC = ∅ too. If NICA is not ∅, for all indices in field ICA the indices of the cornerpoint-coordinates XC, YC, ZC are computed.

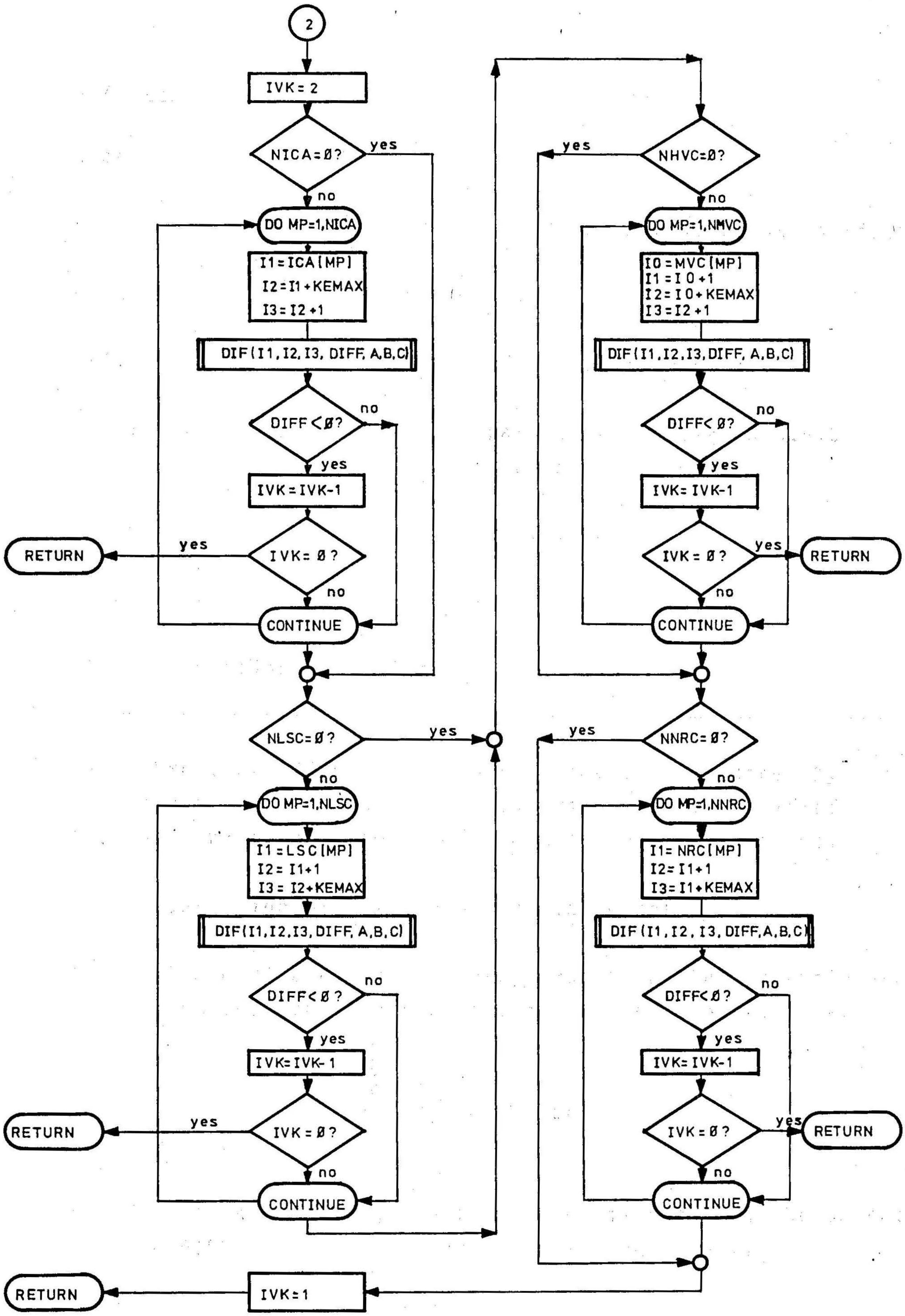
The cornerpoints have the indices ICA(MP), ICA(MP)+KEMAX, ICA(MP)+KEMAX+1, where MP is a DO variable running from 1 to NICA.

Then the subroutine DIF is called. The indices of the cornerpoint-coordinates and the test point-coordinates are transferred to it as input parameters. As initial size one gets the parameter DIFF which indicates whether the test point was covered by the respective triangle or not:

$$\text{DIFF} < \emptyset \quad \text{invisible}$$
$$\text{DIFF} \geq \emptyset \quad \text{visible.}$$

The arrays LSC, MVC, NCR with the numbers NLSC, NMVC, NNRX are considered in the same way as before arrays ICA with the number NICA. If all arrays are processed, IVK has the value 1 or 2. In both cases IVK=1 is set, the test point thus is recognized as visible and it is returned into the calling programm GREVIS, unless already previously with IVK=∅ was returned.


VISKRI
X=A-XMIN
Y=B-YMIN
RX=IFIX[AX/DELTAX]+1
RY=IFIX[BY/DELTAY]+1
DX=IFIX[AX/TEILX]+1
DY=IFIX[BY/TEILY]+1
PX=IFIX[AX/SIGMAX]+1
PY=IFIX[BY/SIGMAY]+1
GX=IFIX[AX/THETAX]+1
GY=IFIX[BY/THETAY]+1
SEARCH1 = SEARCH[NIHQ,IHQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
SEARCH2 = SEARCH[NKWQ,KWQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
SEARCH3 = SEARCH[NLFQ,LFQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
SEARCH4 = SEARCH[NMBQ,MBQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
ZERLEG1 = ZERLEG[NIHQ,IHQ,NKWQ,KWQ,TEILX,TEILY,KDX,KDY]
ZERLEG2 = ZERLEG[NKWQ,KWQ,NLFQ,LFQ,SIGMAX,SIGMAY,LPX,LPY]
ZERLEG3 = ZERLEG[NLFQ,LFQ NMBQ,MBQ,THETAX,THETAY,MGX,MGY]
IRX=IRXA und IRY=IRYA ?
yes
no
KDX=KDXA und KDY=KDYA ?
yes
no
LPX=LPXA und LPY=LPYA ?
yes
no
MGX=MGXA und MGY=MGYA ?
yes
no
INTER[IRY,IRX,NIHQ]
NIHQ ≦ IOPT1?
yes
no
ZERLEG 1
SEARCH 1
NKWQ ≦ IOPT2?
yes
no
ZERLEG 2
SEARCH 2
NLFQ ≦ IOPT3?
yes
no
ZERLEG3
SEARCH3
SEARCH4
IRXA=IRX
IRYA=IRY
KDXA=KDX
KDYA=KDY
LPXA=LPX
LPYA=LPY
MGXA=MGX
MGYA=MGY
ISPRU = ?
= Ø
-1
+1
IVK = Ø
2
RETURN


2
IVK = 2
NICA = Ø?
yes
no
DO MP=1,NICA
I1 = ICA(MP)
I2 = I1 + KEMAX
I3 = I2 + 1
DIF(I1, I2, I3, DIFF, A,B,C)
DIFF < Ø?
no
yes
IVK = IVK-1
IVK = Ø ?
yes
RETURN
no
CONTINUE
NLSC = Ø ?
yes
no
DO MP=1,NLSC
I1 = LSC(MP)
I2 = I1+1
I3 = I2+KEMAX
DIF(I1, I2, I3, DIFF, A, B, C)
DIFF < Ø ?
no
yes
IVK = IVK- 1
IVK = Ø ?
yes
RETURN
no
CONTINUE
RETURN
IVK = 1
NHVC = Ø?
yes
no
DO MP=1,NMVC
I0 = MVC(MP)
I1 = I0 +1
I2 = I0 + KEMAX
I3 = I2 +1
DIF(I1, I2, I3, DIFF, A,B,C)
DIFF < Ø?
no
yes
IVK = IVK-1
IVK = Ø ?
yes
RETURN
no
CONTINUE
NNRC = Ø ?
yes
no
DO MP=1,NNRC
I1 = NRC(MP)
I2 = I1+1
I3 = I1 + KEMAX
DIF (I1, I2, I3, DIFF, A, B, C)
DIFF < Ø ?
no
yes
IVK = IVK-1
IVK = Ø ?
yes
RETURN
no
CONTINUE

3.7 DIF:

Name:	DIF
Key-word:	Testing whether a test point is visible
Language:	FORTRAN II
Call:	CALL DIF (I1,I2,I3,DIFF,A,B,C)
Parameters:	Input parameters: I1,I2,I3,A,B,C Output parameter: DIFF
COMMON:	EPS, XC, YC, ZC
Subroutines:	(ERROR)
Storage Location:	228

The subroutine DIF gets as input parameters 3 indices by which the corner point coordinates of a triangle are specified. The coordinates of a test point are further input sizes.

First subroutine DIF tests the trivial cases:

1. All Z-coordinates of the corner points of the triangle are smaller than the Z-coordinate of the test point, i. e.

   $$ZC(I1) < C,$$
   $$ZC(I2) < C, \text{ and}$$
   $$ZC(I3) < C.$$

   If this case occurs, the test point is visible, the output size DIFF is set =+1∅., and it is returned to VISKRI.

2. All Z-coordinates of the corner points of the triangle are greater than the Z-coordinate of the test point, i. e.

$$ZC\ [I1] > C,$$
$$ZC\ [I2] > C \text{ and}$$
$$ZC\ [I3] > C\ .$$

If that case occurs, the test point is invisible, the output size DIFF is set = - 1 Ø., and it is returned to VISKRI.

If none of the two cases occurred, the test point has to be tested more detailed. To this some preliminary reflections:

Let P1(X1,Y1,Z1), P2(X2,Y2,Z2), and P3(X3,Y3,Z3) be three points in space. These three three points form a plane. A fourth point P(X,Y,Z) lies in the plane if and only if

$$\begin{vmatrix} X1 & X2 & X3 & X \\ Y1 & Y2 & Y3 & Y \\ Z1 & Z2 & Z3 & Z \\ 1 & 1 & 1 & 1 \end{vmatrix} = \emptyset \qquad (1)$$

In DIF by solving equation (1) it is ascertained, whether the Z-coordinate of the test point corresponds to Z, i.e. (1) becomes

$$\begin{vmatrix} XF1 & XF2 & XF3 & A \\ YF1 & YF2 & YF3 & B \\ ZF1 & ZF2 & ZF3 & ZV \\ 1 & 1 & 1 & 1 \end{vmatrix} = \emptyset \qquad (2)$$

(2) is solved to ZV:

$$ZV = ZF1 + \big[(B-YF1)\cdot(XX1\cdot ZZ2-XX2\cdot ZZ1)$$
$$(A-XF1)\cdot(YY1\cdot ZZ2-YY2\cdot ZZ1)\big]\ /DIVID \qquad (3)$$

with $XX1 = XF2 - XF1 = XC(I2) - XC(I1)$ (4a)

$$XX2 = XF3 - XF1 = XC(I3) - XC(I1) \quad (4b)$$

$$YY1 = YF2 - YF1 = YC(I2) - YC(I1) \quad (4c)$$

$$YY2 = YF3 - YF1 = YC(I3) - YC(I1) \quad (4d)$$

$$ZZ1 = ZF2 - ZF1 = ZC(I2) - ZC(I1) \quad (4e)$$

$$ZZ2 = ZF3 - ZF1 = ZC(I3) - ZC(I1) \quad (4f)$$

and $DIVID = XX1 \cdot YY2 - XX2 \cdot YY1$ (5)

To avoid rounding errors, after each partial size (equations (4a) to (4f)) the subroutine ERROR is called, which controls whether the coordinate-difference has already become smaller than epsilon. If that occurs, the coordinate-difference is set Ø. This is necessary to avoid - when having many little patches lying closely together - that the picture parts lying in front of the patches are recognized as invisible. (so called "moth effect"). Mathematically this effect is to explain in the following way: the three points forming the test plane lie, within the limits of the computing accuracy, so closely together that the plane is no longer determined uniquely. If such cases shall not be considered, you can renounce to call ERROR.

Moreover it has to be queried whether size DIVID has become Ø, to avoid a division through Ø.

If DIVID = Ø that means

1. P1 (X1, Y1, Z1) = P2 (X2, Y2, Z2),

or 2. P1 (X1, Y1, Z1) = P3 (X3, Y3, Z3),

or 3. points P1,P2,and P3 lie on a line.

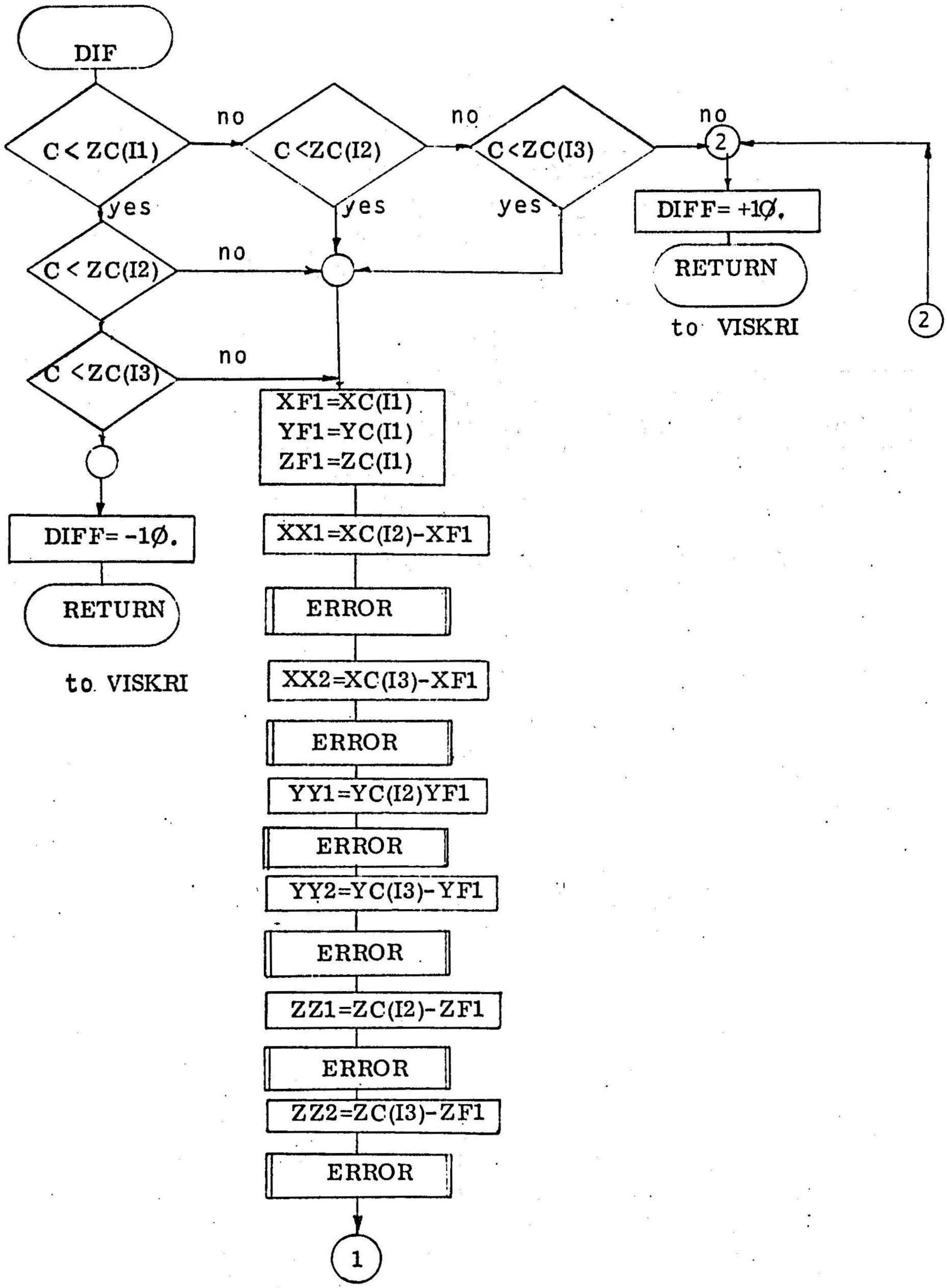
If one of the three cases occurred, the test plane is no longer determined uniquely (plane degenerated to straight line). In this case DIFF = +1 Ø. is set (i.e. visible) and it is returned to VISKRI.

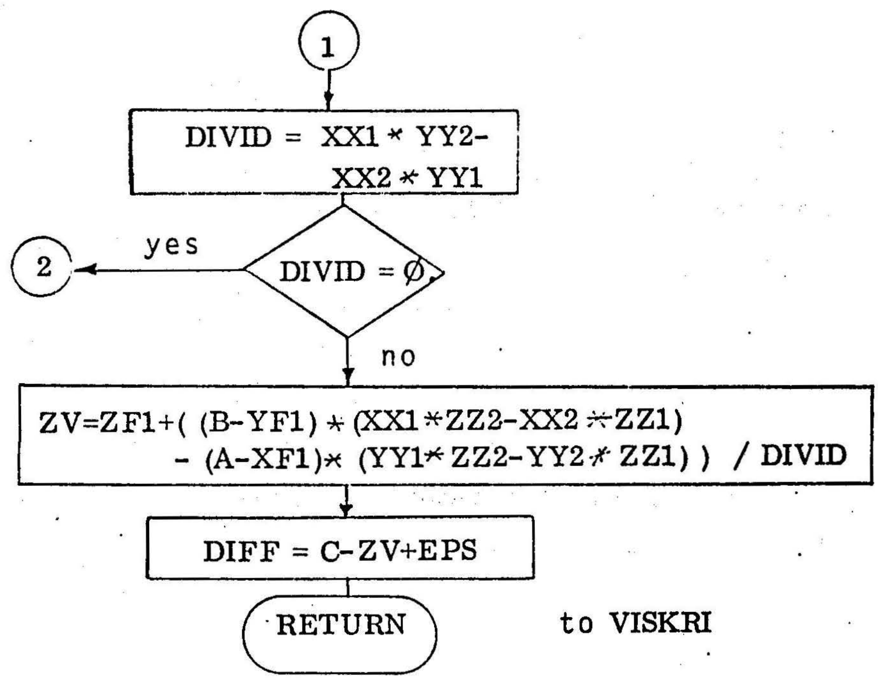
In all other cases after the solution of equation (3) size DIFF is computed by taking equations (4a) to (4f), and (5):

$$DIFF = C - ZV \qquad (6)$$

Thereafter it is returned to VISKRI, where size DIFF is evaluated:

DIFF $<$ Ø test point invisible: IVK=IVK-1
DIFF $\geqslant$ Ø test point visible: IVK=IVK.


DIF
C < ZC(I1)
no
C <ZC(I2)
no
C <ZC(I3)
no
2
yes
yes
yes
DIFF= +1Ø.
RETURN
to VISKRI
2
C < ZC(I2)
no
C <ZC(I3)
no
XF1=XC(I1)
YF1=YC(I1)
ZF1=ZC(I1)
DIFF= -1Ø.
RETURN
to VISKRI
XX1=XC(I2)-XF1
ERROR
XX2=XC(I3)-XF1
ERROR
YY1=YC(I2)YF1
ERROR
YY2=YC(I3)-YF1
ERROR
ZZ1=ZC(I2)-ZF1
ERROR
ZZ2=ZC(I3)-ZF1
ERROR
1


1
DIVID = XX1 * YY2-
XX2 * YY1
yes
2
DIVID = Ø
no
ZV=ZF1+((B-YF1) * (XX1*ZZ2-XX2 *ZZ1)
- (A-XF1)* (YY1*ZZ2-YY2* ZZ1)) / DIVID
DIFF = C-ZV+EPS
RETURN
to VISKRI

3.8 ERROR

Name: ERROR

Key-word: Elimination of computing inaccuracy

Language: FORTRAN II

Call: CALL ERROR (VAL, EPS)

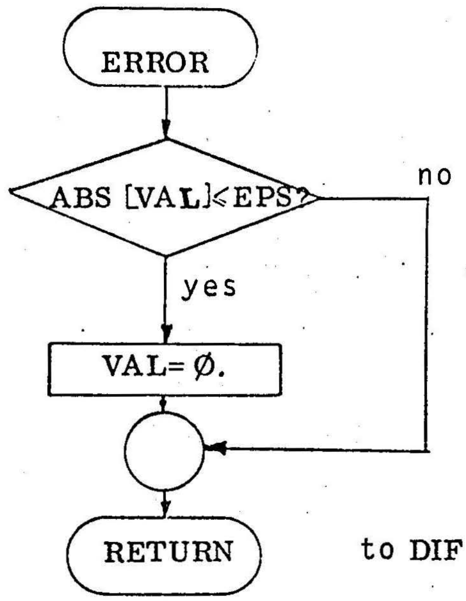
Parameters: Input parameters: VAL, EPS
Output parameter: VAL

COMMON: -

Subroutines: -

Storage location: 35

The subroutine ERROR tests the input parameter VAL, whether it is absolutely already smaller than EPS (=$1Ø^{-5}$). If that is the case, VAL=Ø is set, otherwise VAL is preserved.


ERROR
ABS [VAL]< EPS?
no
yes
VAL= Ø.
RETURN
to DIF

3.9 INTER

Name:	INTER
Key-word:	patches in the screen patch
Language:	FORTRAN II
Call:	CALL INTER(IRY, IRX, NIHQ)
Parameters:	Input parameters: IRY, IRX Output parameter: NIHQ
COMMON:	LIHIWI, INDLI, IHQ
Subroutines:	ASSIGN, LODBIT
Storage location:	92

The subroutine INTER computes the patches in a screen patch, which is fixed by the input parameters IRX, IRY. The output parameter NIHQ indicates the number of patches searched, the indices of which are collected in field IHQ. As generally the number of patches is different in each screen patch, it has to be newly computed in each INTER-call. In order to have the correct beginning-value for it, in the beginning NIHQ=Ø is set.

The number of patches in the respective screen patch is also given by the value of the indication array LIHIWI (IRY, IRX,1), which is computed in subroutine FLAVIS.

In a DO-loop, beginning with 1 and ending with the value LIHIWI (IRY, IRX, 1)=LI1, in each run NIHQ is increased by 1 and the patch counter IANZ is computed. The value of IANZ results from the start value for each screen patch LI2=LIHIWI(IRY,IRX,2) and NIHQ IANZ is transferred as input parameter to subroutine ASSIGN, which yields as output parameters IZQ, IMIN, IMAX. The system dependent program LODBIT loads into cell IHQ(ML) the index, which in cell INDLI(IZQ) of the index list is situated between bits IMIN and IMAX.

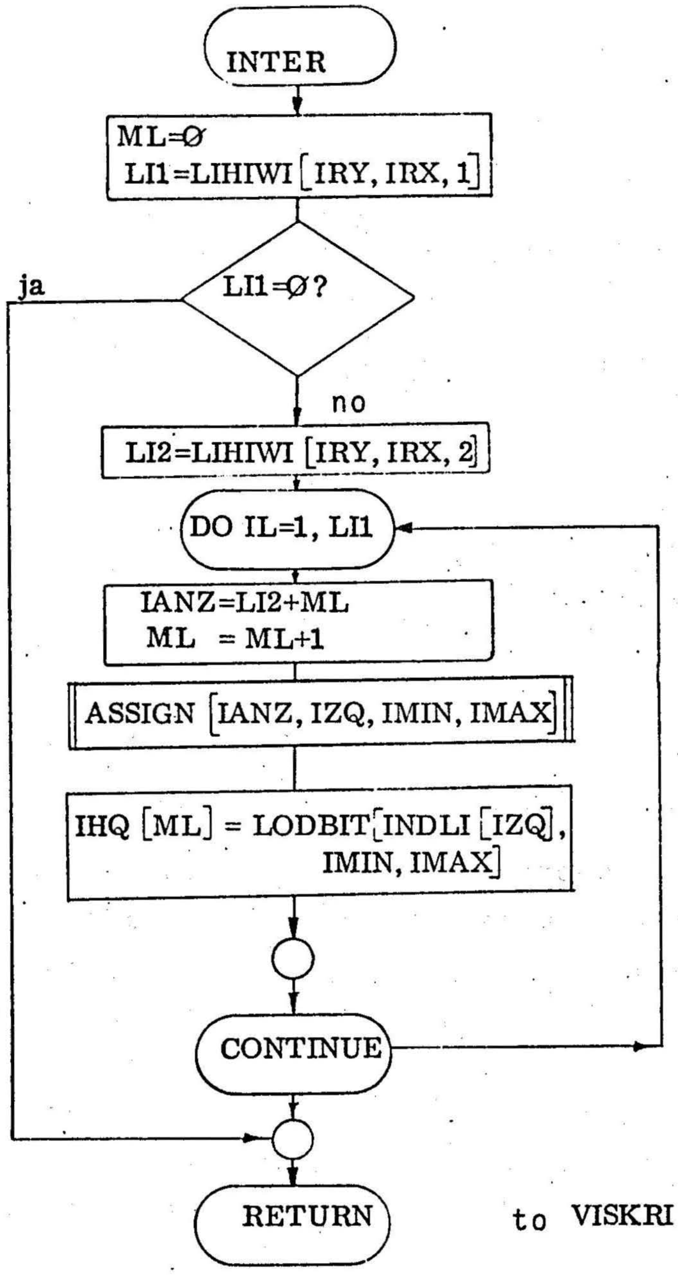
3.9 INTER

Name:	INTER
Key-word:	patches in the screen patch
Language:	FORTRAN II
Call:	CALL INTER(IRY, IRX, NIHQ)
Parameters:	Input parameters: IRY, IRX Output parameter: NIHQ
COMMON:	LIHIWI, INDLI, IHQ
Subroutines:	ASSIGN, LODBIT
Storage location:	92

The subroutine INTER computes the patches in a screen patch, which is fixed by the input parameters IRX, IRY. The output parameter NIHQ indicates the number of patches searched, the indices of which are collected in field IHQ. As generally the number of patches is different in each screen patch, it has to be newly computed in each INTER-call. In order to have the correct beginning-value for it, in the beginning NIHQ=Ø is set.

The number of patches in the respective screen patch is also given by the value of the indication array LIHIWI (IRY, IRX,1), which is computed in subroutine FLAVIS.

In a DO-loop, beginning with 1 and ending with the value LIHIWI (IRY, IRX, 1)=LI1, in each run NIHQ is increased by 1 and the patch counter IANZ is computed. The value of IANZ results from the start value for each screen patch LI2=LIHIWI(IRY,IRX,2) and NIHQ IANZ is transferred as input parameter to subroutine ASSIGN, which yields as output parameters IZQ, IMIN, IMAX. The system dependent program LODBIT loads into cell IHQ(ML) the index, which in cell INDLI(IZQ) of the index list is situated between bits IMIN and IMAX.


INTER
ML=Ø
LI1=LIHIWI [IRY, IRX, 1]
LI1=Ø?
ja
no
LI2=LIHIWI [IRY, IRX, 2]
DO IL=1, LI1
IANZ=LI2+ML
ML = ML+1
ASSIGN [IANZ, IZQ, IMIN, IMAX]
IHQ [ML] = LODBIT[INDLI [IZQ],
IMIN, IMAX]
CONTINUE
RETURN
to VISKRI

3.10 ZERLEG

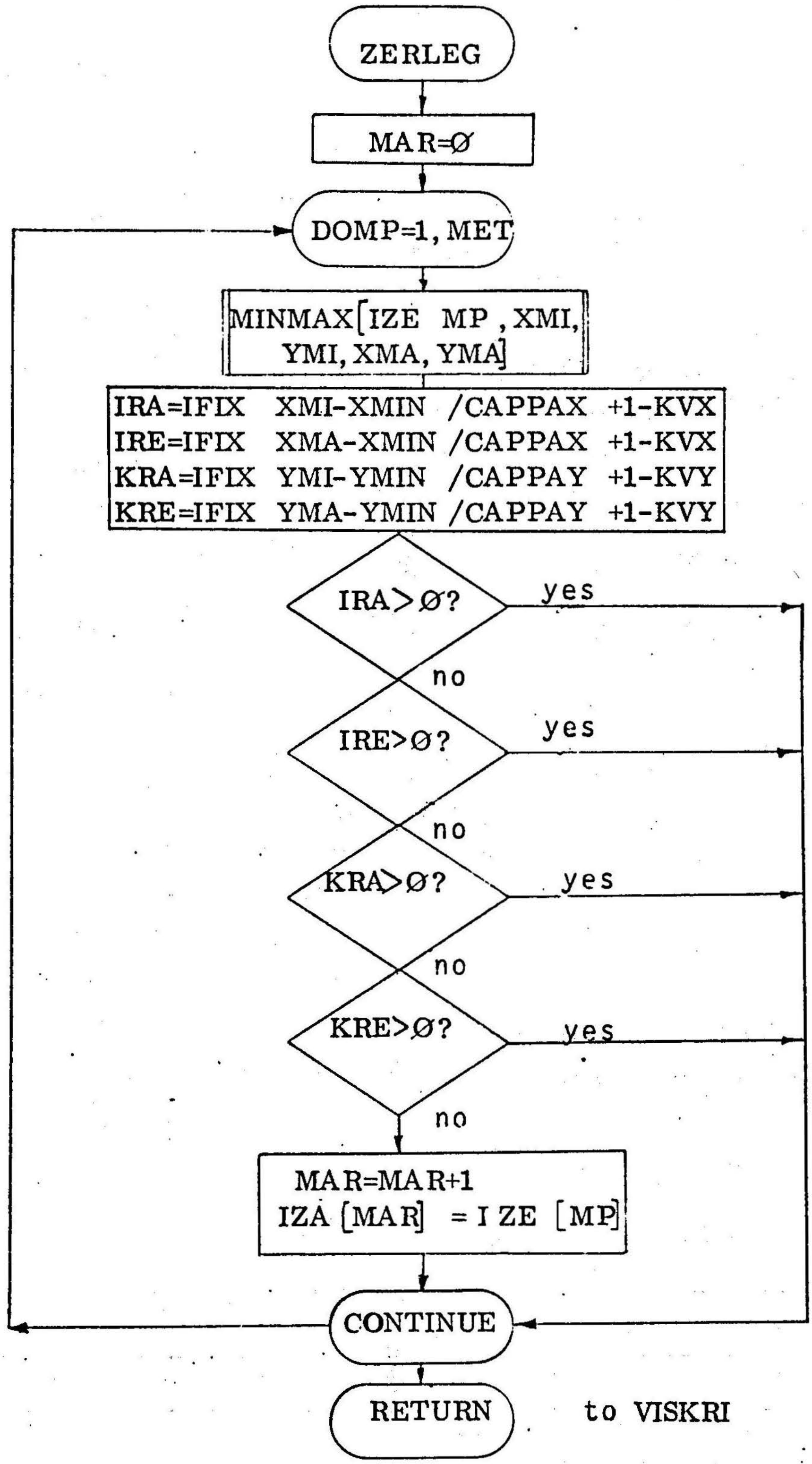
Name:	ZERLEG
Key-word:	Division of a screen patch
Language:	FORTRAN II
Call:	CALL ZERLEG (MET, IZE, MAR, IZA, CAPPAX, CAPPAY, KVX, KVY)
Parameters:	Input parameters: MET, IZE, CAPPAX, CAPPAY, KVX, KVY Output parameters:MAR, IZA
COMMON:	XMIN, YMIN
Subroutines:	MINMAX
Storage Location:	147

Out of a quantity of patches - the number of which is MET and the indices are stored in field IZE - those having point quantities in one quarter of the existing screen part are computed. The position of the screen quarter is given by the variables KVX, KVY, the appertaining distances between X- and Y-screen-lines are CAPPAX, CAPPAY. The indices of those patches, having point quantities in the screen part determined by KVX, KVY, are collected in array IZA Their number is allocated to the output parameter MAR.

In order to get the right beginning value for each call of ZERLEG, in the beginning MAR=Ø is set. For each patch, the index of which is in array IZE, its minimum XMI, YMI- and maximum XMA, YMA-values in X- and Y-direction are preserved by calling subroutine MINMAX. With these minimum- and maximum-values the variables IRA,IRE respecting screen-line-distance CAPPAX,the variables KRA,KRE respecting screen-line-distance CAPPAY are formed analogously to subroutine FLAVIS. IRA,KRA indicate the first screen part, IRE,KRE the last screen part containing point quantities of the patch.

If the first value IRA is greater than KVX or the last IRE less than KVX, the patch cannot have point quantities in screen part KVC, KVY.It is the same, if KRA greater KVY or KRE less KVY. In this case it is passed to the next patch. In all other cases the patch has point quantities in the fixed screen part. The output parameter MAR is increased by 1, and the index of the patch, which is in array IZE, is stored in array IZA.

The patches,lying in a screen patch IRY, IRX, were computed in subroutine INTER, where an indication array LIHIWI and an index list INDLI are used.


ZERLEG
MAR=Ø
DOMP=1, MET
MINMAX[IZE MP , XMI, YMI, XMA, YMA]
IRA=IFIX XMI-XMIN /CAPPAX +1-KVX
IRE=IFIX XMA-XMIN /CAPPAX +1-KVX
KRA=IFIX YMI-YMIN /CAPPAY +1-KVY
KRE=IFIX YMA-YMIN /CAPPAY +1-KVY
IRA>Ø?
yes
no
IRE>Ø?
yes
no
KRA>Ø?
yes
no
KRE>Ø?
yes
no
MAR=MAR+1
IZA [MAR] = I ZE [MP]
CONTINUE
RETURN
to VISKRI

3.11 SEARCH

Name: SEARCH

Key-word: Searching of patches containing the test point

Language: FORTRAN II

Call: CALL SEARCH (NIRS, IRS, NICA, NLSC, NMVC, NNRC, A, B, C, ISPRU)

Parameters: Input parameters: A, B, C, NIRS, IRS

Output parameters: NICA, NLSC, NMVC, NNRS, ISPRU

COMMON: EPS, ICA, KEMAX, LSC, MYC, NRC, XC, YC, ZC, IN

Subroutines: INFLAP, INFLA4, INFLA3

Storage Location: 237

Out of a number NIRS of patches, the indices of which are in field IRS, those containing the test point P(A,B,C) are to be searched.

First the counters NICA, NLSC, NMVC,and NNRC=Ø are set. The significance of the counters is shown in the following table (see also chap. 6):

Array of surface indices	1. Index	2. Index	3. Index	Number of surfaces in the array
ICA	N	N+KEMAX	N+KEMAX+1	NICA
LSC	N	N+1	N+KEMAX+1	NLSC
MVC	N+1	N+KEMAX+1	N+KEMAX	NMYC
NRC	N	N+1	N+KEMAX	NNRC

Moreover the branch-parameter ISPRU=1 is set.
With the call CALL INFLAP(A,B) in the subroutine package INFLAS the X-, Y-coordinates of the test point are transferred. In the further process of SEARCH only the indices of the corner points

of the respective patch are transferred to INFLAS.

Dependent on the DO-index of the DO-loop, these indices are computed in the loop. Then it is queried, whether all Z-coordinates of the quadrangle are smaller than the Z-coordinate C of the test point. If that is the case, the patch is irrelevant for the further testing, i.e. the patch cannot cover the test point and it is returned to the end of the loop.

If not all Z-coordinates of the quadrangle are smaller than C, it is computed by call CALL INFLA4(IR,IR1,IR3,IR2), whether the test point lies in the quadrangle. The result is in the COMMON-size IN. The address-allocation for IN is made in subroutine FLAVIS.

IN	Significance
-2	test point lies on a line
-1	test point lies in a corner point
∅	test point outside the quadrangle
+1	test point inside the quadrangle

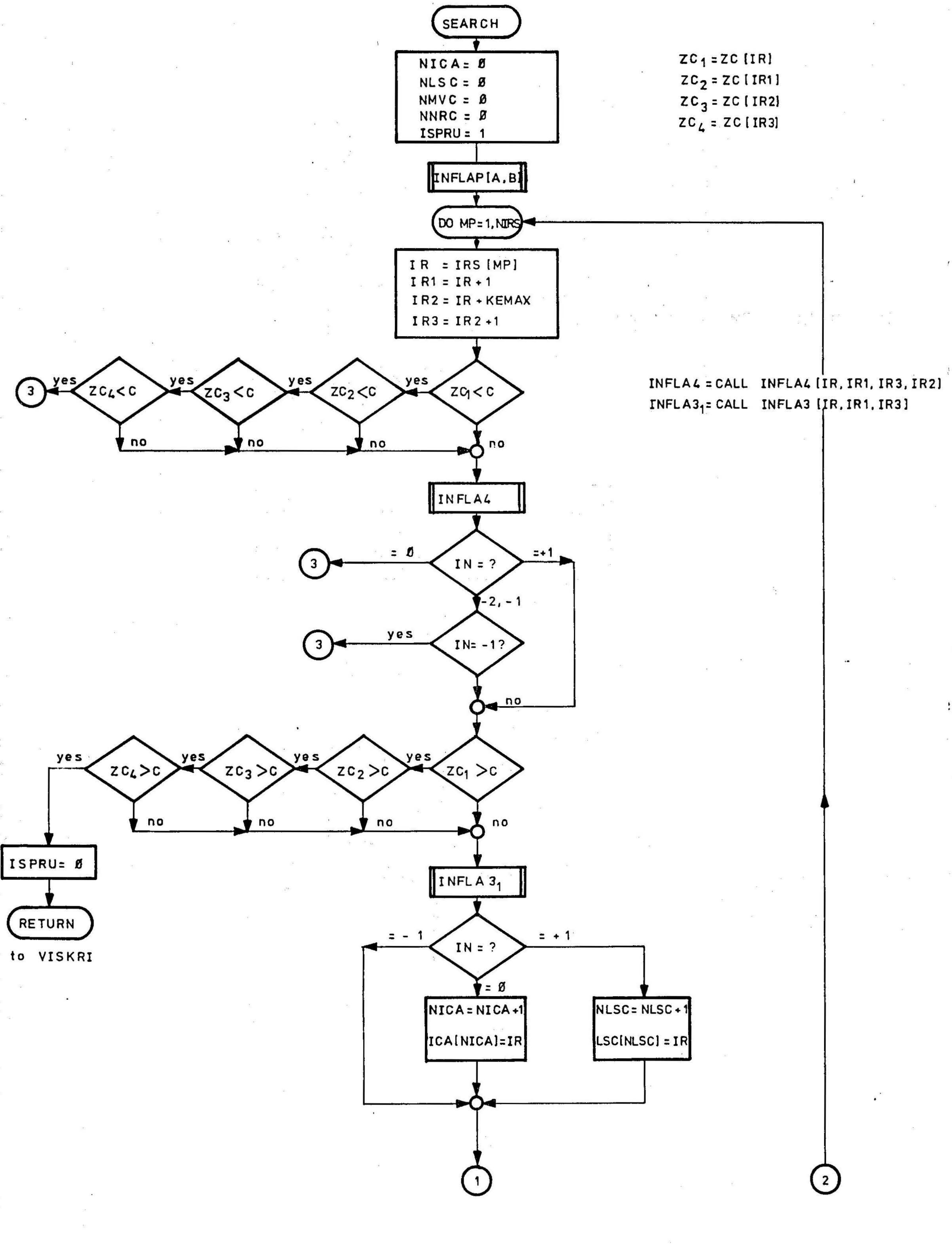
In the following branches IN is tested:

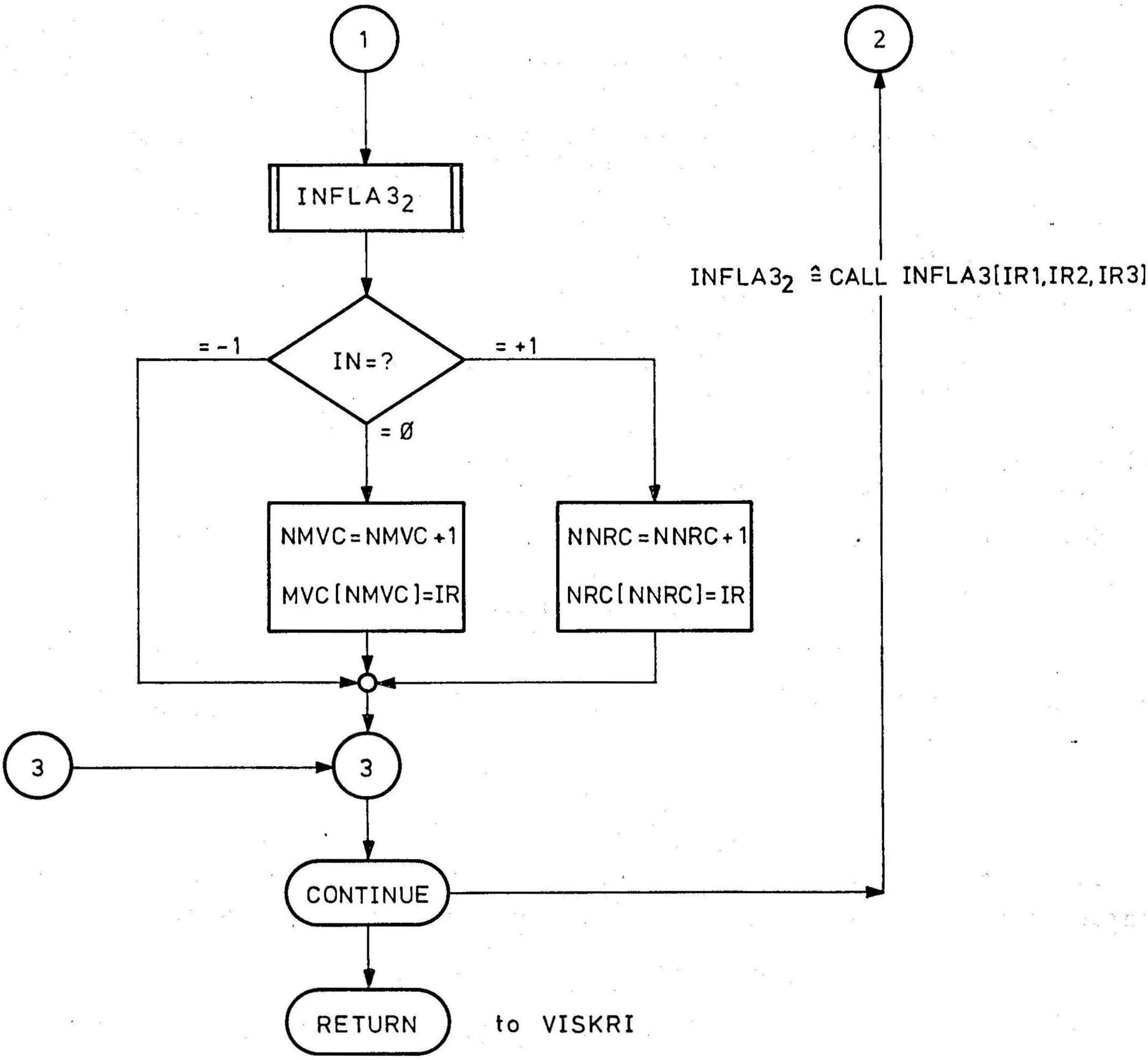
IN	further operations
=-1	branch off loop-end
=0	branch off loop-end
>∅ =-2	further testing of the Z-coordinates of the quadrangle

If IN=-2 or >∅ (test point lies in the quadrangle or on a boundary line), the Z-coordinates of the quadrangle are tested, whether all Z-coordinates of the quadrangle are greater than Z-coordinate C of the test point. If that is the case , the quadrangle anyway covers test point P and the branch parameter ISPRU is set to zero. Afterwards it is returned to VISKRI, where ISPRU=∅ leads to a return to GREVIS resp. VIDRAW with IVK=∅ (point invisible).

After the exclusion of all trivial cases, the quadrangle is divided in two different ways in triangles (see chap. 3.6) and by call CALL INFLA3(INDEX1, INDEX2, INDEX3) it is computed whether the test point is in a triangle.

When the loop is worked off, the numbers of the respective triangles have been allocated to the output parameters and the input sizes are unchanged. Then it is returned to VISKRI.


SEARCH
NICA = Ø
NLSC = Ø
NMVC = Ø
NNRC = Ø
ISPRU = 1
ZC1 = ZC [IR]
ZC2 = ZC [IR1]
ZC3 = ZC [IR2]
ZC4 = ZC [IR3]
INFLAP [A, B]
DO MP = 1, NIRS
IR = IRS [MP]
IR1 = IR + 1
IR2 = IR + KEMAX
IR3 = IR2 + 1
ZC1 < C
ZC2 < C
ZC3 < C
ZC4 < C
yes
no
3
INFLA4 = CALL INFLA4 [IR, IR1, IR3, IR2]
INFLA31 = CALL INFLA3 [IR, IR1, IR3]
INFLA4
IN = ?
= Ø
=+1
-2, -1
IN = -1?
ZC1 > C
ZC2 > C
ZC3 > C
ZC4 > C
ISPRU = Ø
RETURN
to VISKRI
INFLA 31
= - 1
= + 1
= Ø
NICA = NICA + 1
ICA [NICA] = IR
NLSC = NLSC + 1
LSC [NLSC] = IR
1
2


1
2
INFLA3₂
INFLA3₂ ≙ CALL INFLA3[IR1,IR2,IR3]
IN=?
= -1
= +1
= Ø
NMVC=NMVC+1
MVC[NMVC]=IR
NNRC=NNRC+1
NRC[NNRC]=IR
3
3
CONTINUE
RETURN
to VISKRI

3.12 INFLAS new

Name:	INFLAS
Key-word:	Computation, whether a point lies within a surface limited by vectors
Language:	META 920, has to be loaded with the FORTRAN-loader
Calls:	CALL INFLIN (XC, YC, IN) (taking over of the (field addresses and the address of the result parameters) CALL INFLAP (XP, YP) (taking over of the test point-coordinates) CALL INFLA3 (IND1, IND2, IND3) (INFLAS for a patch consisting of 3 corner points) CALL INFLA4 (IND1, IND2, IND3, IND4) (INFLAS for a patch consisting of 4 corner points)
Parameters:	Input parameters: XC,YC,XP,YP;IND1,IND2,IND3, IND4 Output parameter: IN
COMMON:	-
Subroutines:	-
Storage Location:	229

INFLAS has to compute as quickly as possible, whether a test point P(XP,YP) lies within a surface, which is spanned by points

P1 = P1 (XC(IND1), YC(IND1)) ,

P2 = P2 (XC(IND2), YC(IND 2)) ,

P3 = P3 (XC(IND3), YC (IND3))

and byP4 = P4 (XC(IND4), YC (IND4)) when having patches with 4 corner points.

For time-saving reasons, INFLAS is initialized one time in subroutine FLAVIS with the call CALL INFLIN(XC(1),YC(1),IN), i.e. the field addresses and the address of the result parameter IN are transferred.

For the same reasons the coordinates of test point P are transferred in subroutine SEARCH by call CALL INFLAP(XP,YP). In the loops, the indices of the corner points of a patch are then only transferred by CALL INFLA3(IND1,IND2,IND3) resp. CALL INFLA4 (IND1,IND2,IND3,IND4).

The subroutine yields for both, convexo and concave patches the true sentence. The result parameter IN gets for each case the following values:

IN	Significance	
-2	test point P lies on a line	of the patch
-1	test point P lies on a corner point	
Ø	test point P lies outside	
+1	test point P lies inside	

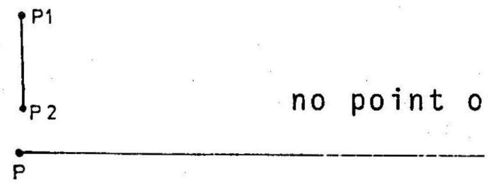
In subroutine INFLAS from a point P a straight line is put in one direction. It is tested by the boundary vectors of the surface whether they form a point of intersection with this straight line. The number of points of intersection available is counted. If the number is Ø or an even number, P is outside, if it is odd, P is inside the patch.

Thus we distinguish the following cases, which are generated by comparison of the signs of the two X-values:

Case 1:

X1 and X2 = Ø

1a) Y1 and Y2 have equal signs

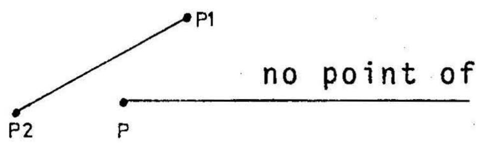


no point of intersection

Case 1b: Y1 and Y2 have different signs

Point P lies on a connection line

Case 2: One X-value = Ø, one negative



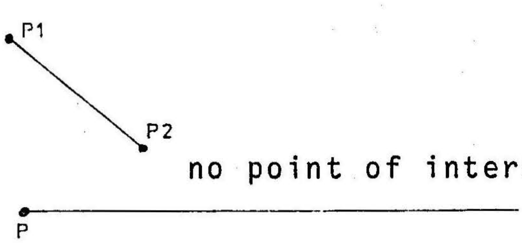
no point of intersection possible

Case 3:

One X-value = Ø, one positive

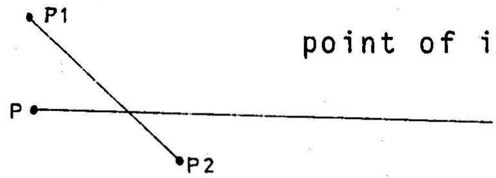
3a:

Y1 and Y2 have equal signs



no point of intersection possible

3b: Y1 and Y2 have different signs



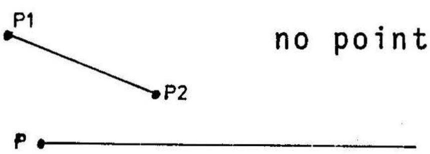
point of intersection is given

Case 4: Both X-values have positive signs;
point of intersection is given with different Y-signs; according to 3.

Case 5: Both X-values have negative signs;
according to 2 no point of intersection possible.

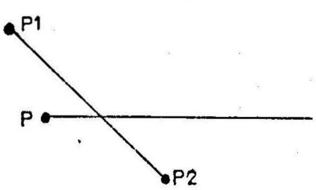
Case 6: X1 and X2 have different signs

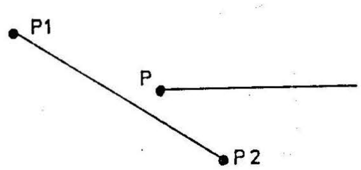
6a: Y1 and Y2 have equal signs



no point of intersection possible.

6b: Y1 and Y2 have different signs





One point of intersection possible.

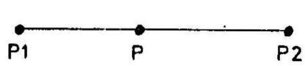
In order to ascertain,whether there is a point of intersection, the value YP is substituted into the straight-line-equation of the connection line.

If result value X=Ø, the point lies on the straight line.

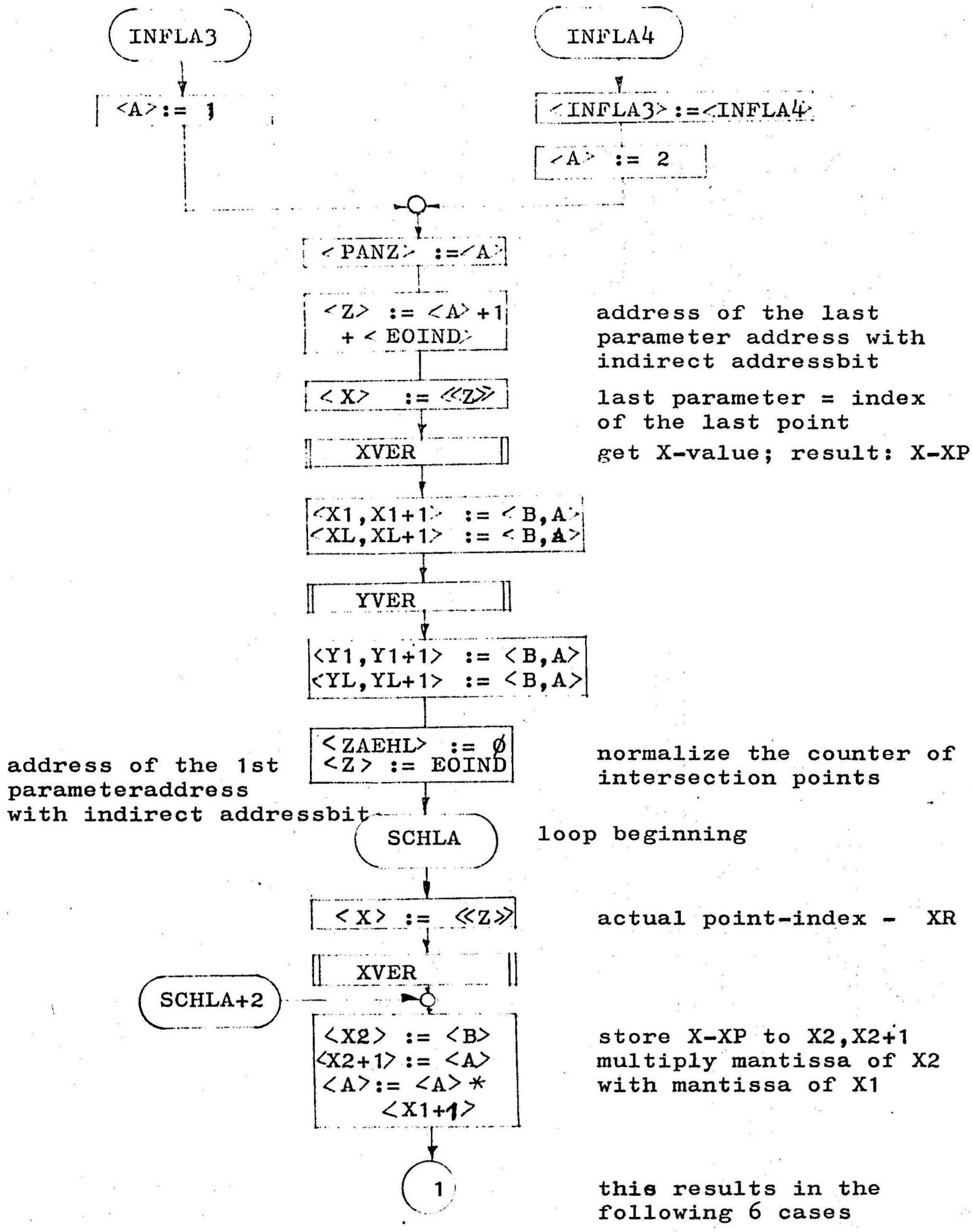
If the value is X > Ø, there is a point of intersection with the positive unlimited line.

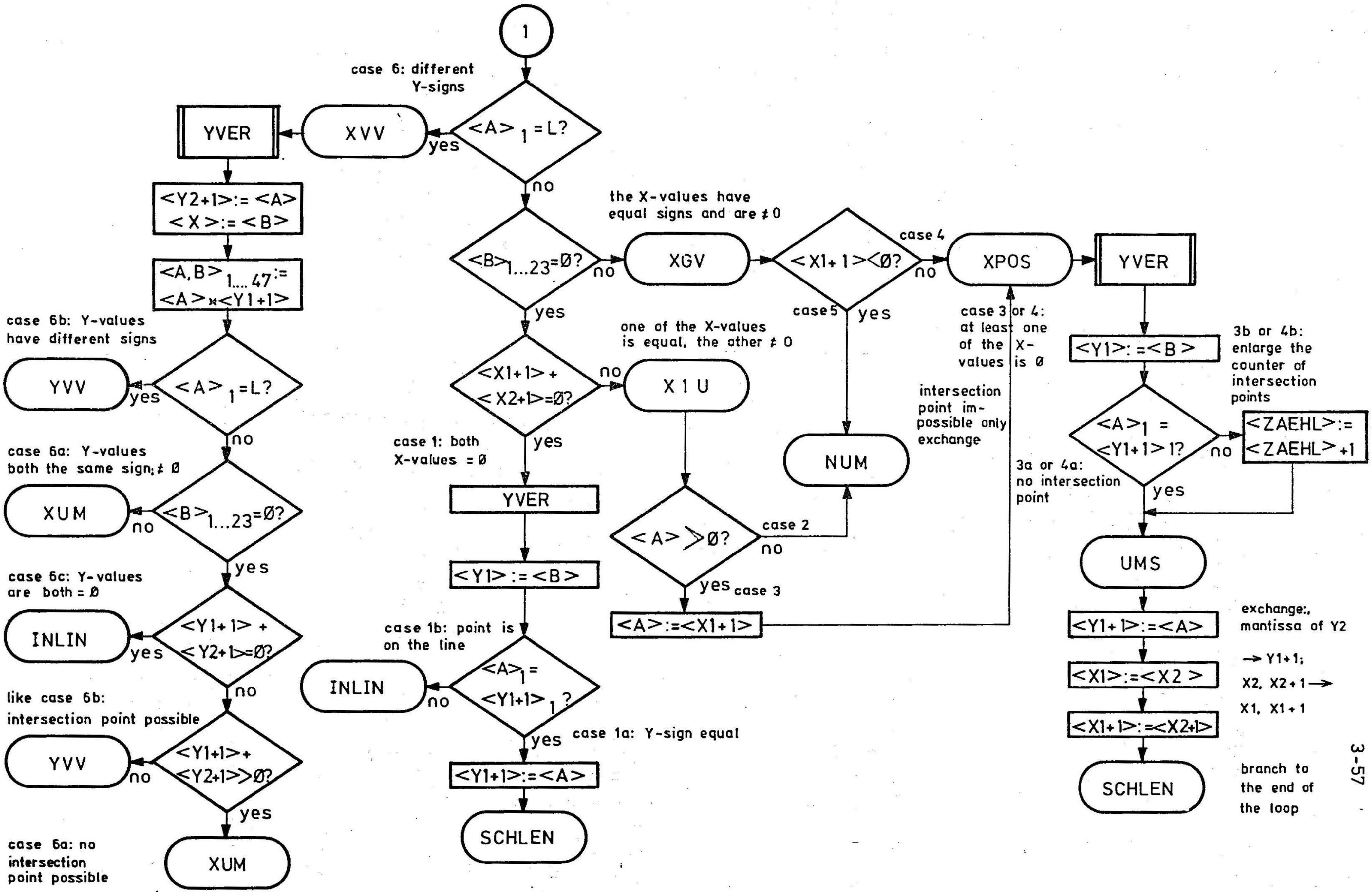
If X-value < Ø, there is no point of intersection.

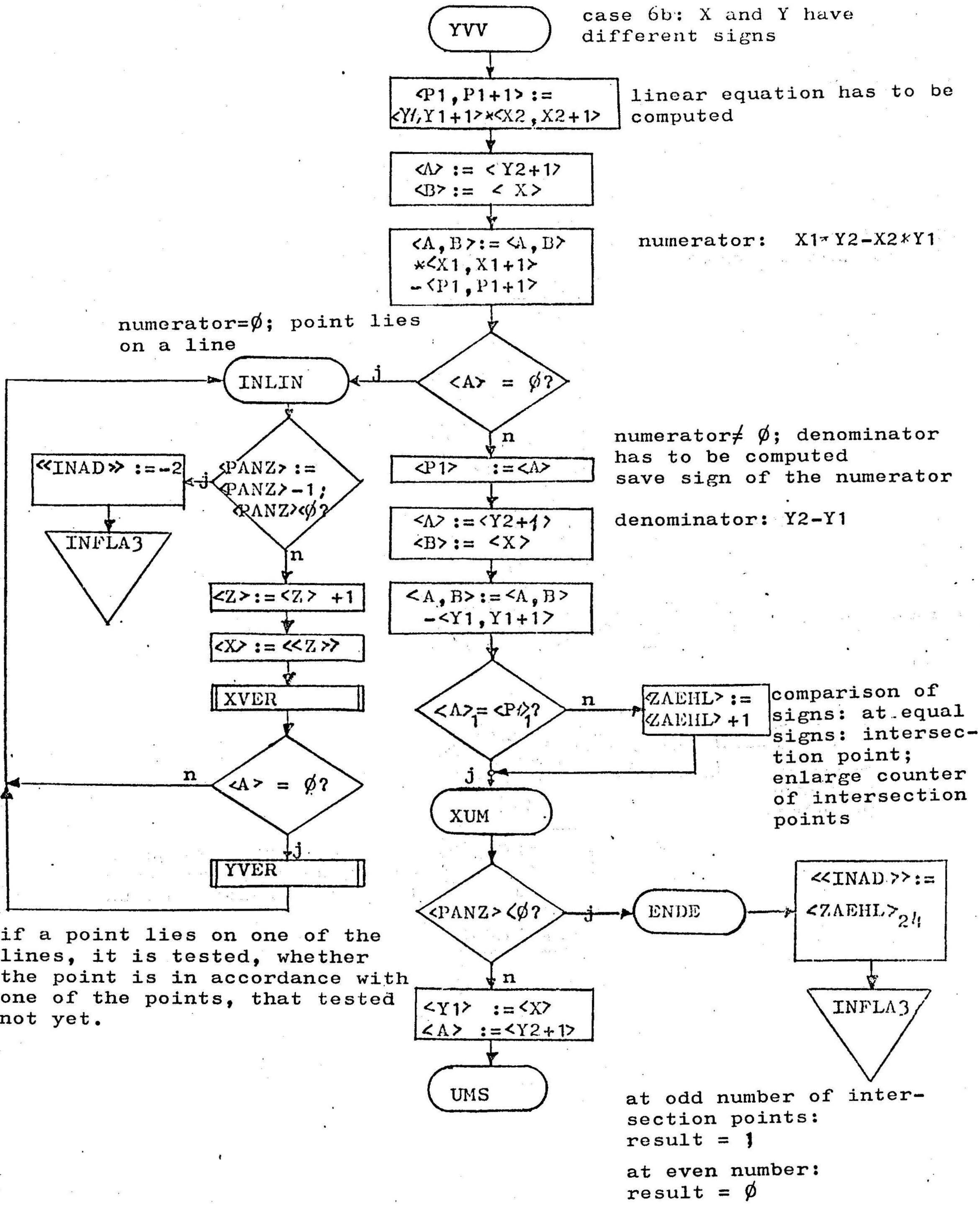
6c: Y1 and Y2 are = Ø



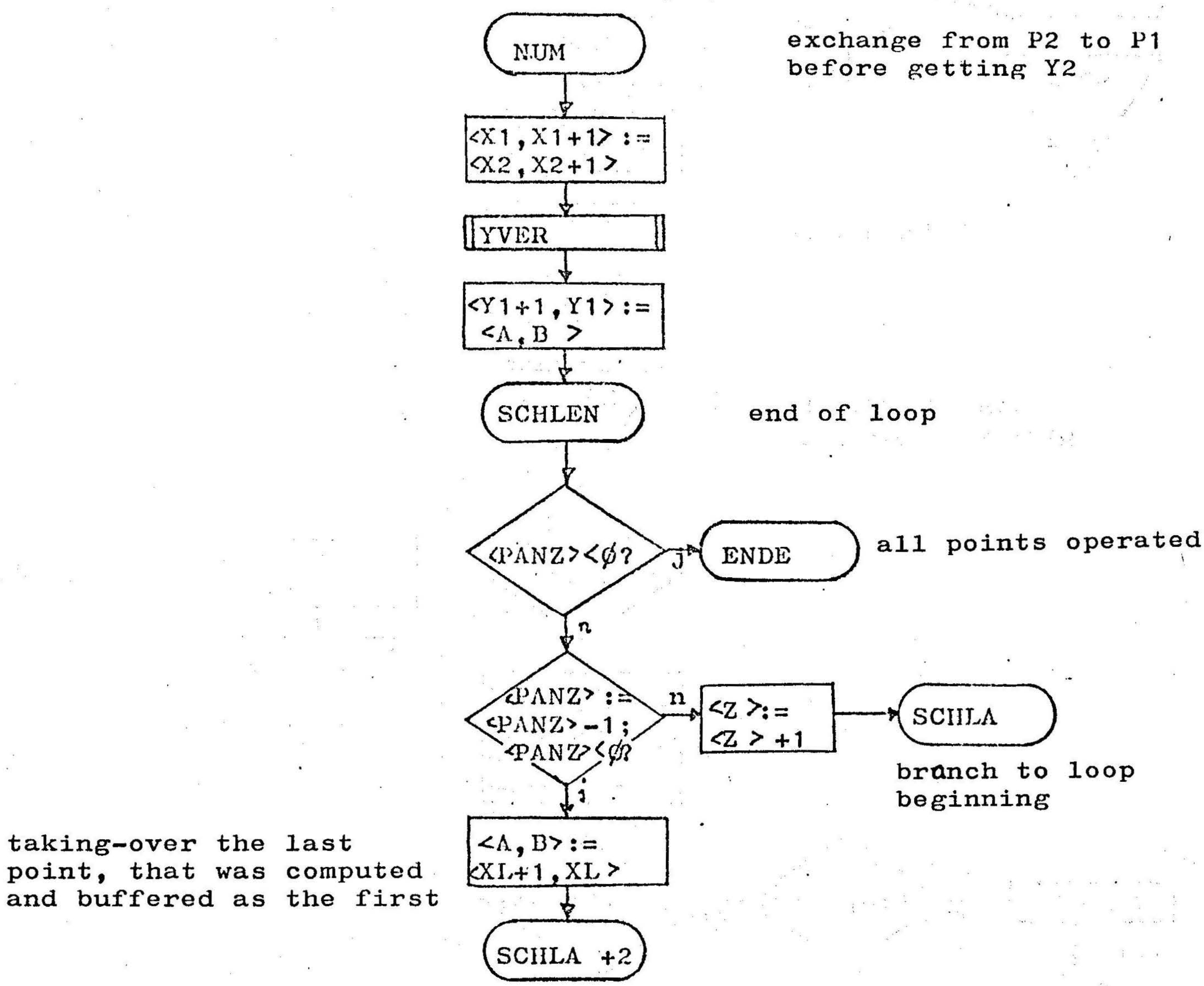
Point P lies on the connection line


INFLA3
<A>:= 1
INFLA4
<INFLA3>:=<INFLA4>
<A> := 2
<PANZ> :=<A>
<Z> := <A>+1 + <EOIND>
address of the last parameter address with indirect addressbit
<X> := <<Z>>
last parameter = index of the last point
XVER
get X-value; result: X-XP
<X1,X1+1> := <B,A>
<XL,XL+1> := <B,A>
YVER
<Y1,Y1+1> := <B,A>
<YL,YL+1> := <B,A>
<ZAEHL> := Ø
<Z> := EOIND
normalize the counter of intersection points
address of the 1st parameteraddress with indirect addressbit
SCHLA
loop beginning
<X> := <<Z>>
actual point-index - XR
XVER
SCHLA+2
<X2> := <B>
<X2+1> := <A>
<A>:= <A>* <X1+1>
store X-XP to X2,X2+1
multiply mantissa of X2 with mantissa of X1
1
this results in the following 6 cases

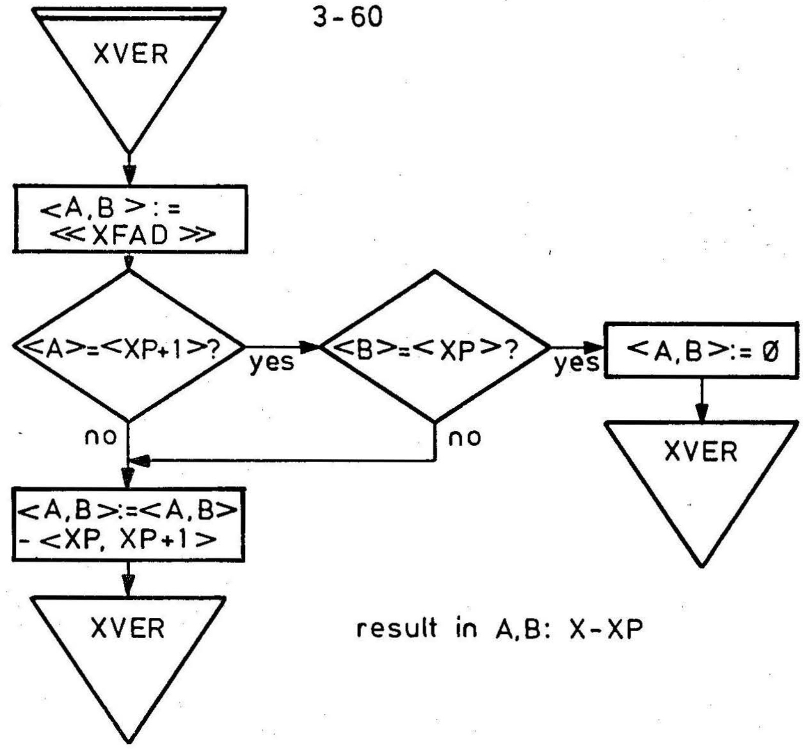

1
case 6: different Y-signs
<A> 1 = L?
yes
no
XVV
YVER
<Y2+1> := <A>
<X> := <B>
<A,B> 1....47 := <A> × <Y1+1>
case 6b: Y-values have different signs
YVV
<A> 1 = L?
case 6a: Y-values both the same sign; ≠ Ø
XUM
<B> 1...23 = Ø?
case 6c: Y-values are both = Ø
INLIN
<Y1+1> + <Y2+1> = Ø?
like case 6b: intersection point possible
YVV
<Y1+1> + <Y2+1> > Ø?
case 6a: no intersection point possible
XUM
<B> 1...23 = Ø?
the X-values have equal signs and are ≠ 0
XGV
<X1+1> < Ø?
case 4
case 5
XPOS
YVER
one of the X-values is equal, the other ≠ 0
<X1+1> + <X2+1> = Ø?
X1U
case 1: both X-values = Ø
YVER
<Y1> := <B>
case 1b: point is on the line
INLIN
<A> 1 = <Y1+1> 1 ?
case 1a: Y-sign equal
<Y1+1> := <A>
SCHLEN
<A> > Ø?
case 2
case 3
<A> := <X1+1>
NUM
intersection point impossible only exchange
case 3 or 4: at least one of the X-values is Ø
3a or 4a: no intersection point
<Y1> := <B>
3b or 4b: enlarge the counter of intersection points
<A> 1 = <Y1+1> 1?
<ZAEHL> := <ZAEHL> +1
UMS
exchange:, mantissa of Y2 → Y1+1;
X2, X2+1 → X1, X1+1
<Y1+1> := <A>
<X1> := <X2>
<X1+1> := <X2+1>
SCHLEN
branch to the end of the loop



if a point lies on one of the lines, it is tested, whether the point is in accordance with one of the points, that tested not yet.


NUM
exchange from P2 to P1 before getting Y2
<X1,X1+1> := <X2,X2+1>
YVER
<Y1+1,Y1> := <A,B >
SCHLEN
end of loop
<PANZ><Ø?
j
ENDE
all points operated
n
<PANZ> := <PANZ> -1; <PANZ><Ø?
n
<Z >:= <Z > +1
SCHLA
branch to loop beginning
j
<A,B>:= <XL+1,XL >
taking-over the last point, that was computed and buffered as the first
SCHLA +2

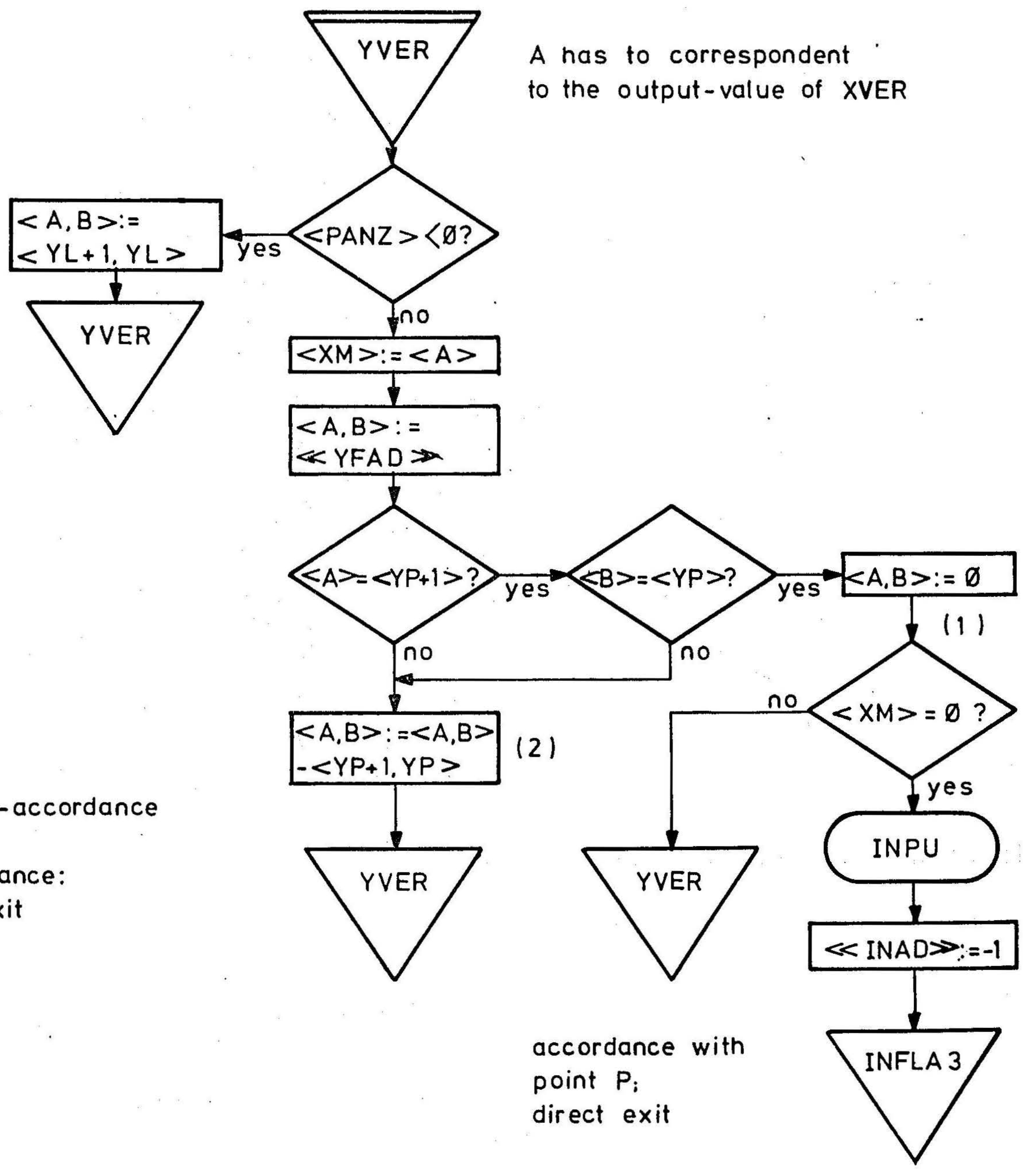
taking over of the X-coord. of the next point;
test regarding accordance with XP; subtraction X-XP



result in A,B: X-XP

A has to correspondent to the output-value of XVER

last point



(1) test for X-accordance

(2) no accordance: normal exit

accordance with point P; direct exit

4. Measurements of time

Measurements of time resulted in the following computing time:

$$T = \left(\sum_{IRY=1}^{11} \sum_{IRX=1}^{11} ML \cdot ITR \right) \cdot T_n \qquad (1)$$

there is

ML = Number of patches in the screen patch, which is defined by IRX and IRY.

ITR= Number of VISKRI-calls, the visibility testings of which refer to a point in the screen patch.

T_n = Computing time for a VISKRI-call, the visibility testing of which refers to a screen patch,that only contains one patch.

From (1) follows

(2) T = f(time-figure),i.e. the computing time is linearly proportional to a time-figure.

In the first approximation results

$$(3) \quad \overline{ML} = \left(\sum_{IRY=1}^{\sqrt{VN}} \sum_{IRX=1}^{\sqrt{VN}} ML \right) \cdot \frac{1}{N}$$

with $\overline{ML}$ = mean value of the patches per screen patch

and

$$(4) \quad \overline{ITR} = \left(\sum_{IRY=1}^{\sqrt{VN}} \sum_{IRX=1}^{\sqrt{VN}} ITR \right) \cdot \frac{1}{N}$$

with $\overline{ITR}$= mean value of the visibility testings per screen patch

Hence follows

$$(5) \quad \overline{ML} \approx \frac{1}{N} \quad ; \quad \overline{ITR} \approx \frac{1}{N}$$

and (6) time-figure $\approx \frac{1}{N}2$

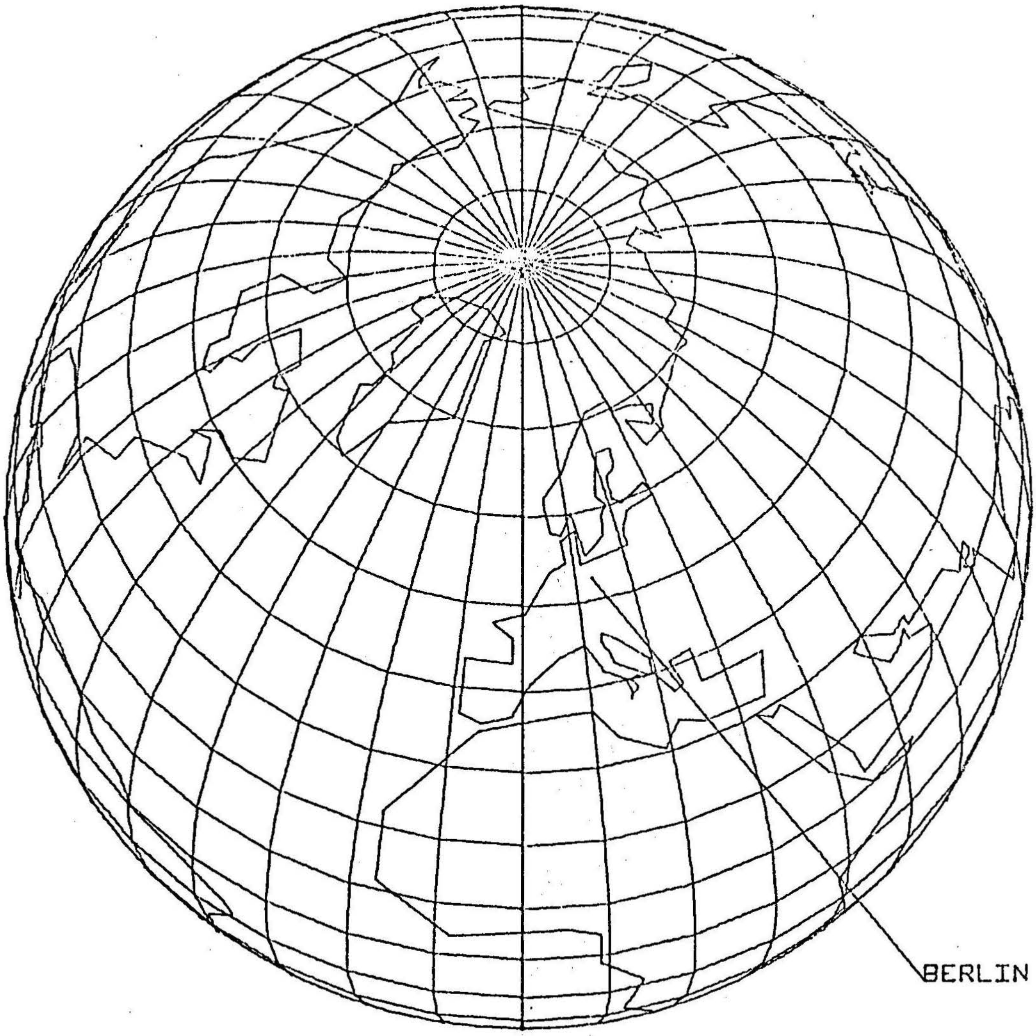
The computing time is inversely proportional to the square of the screen patch number or to the number of screen patches arising from a possible division.

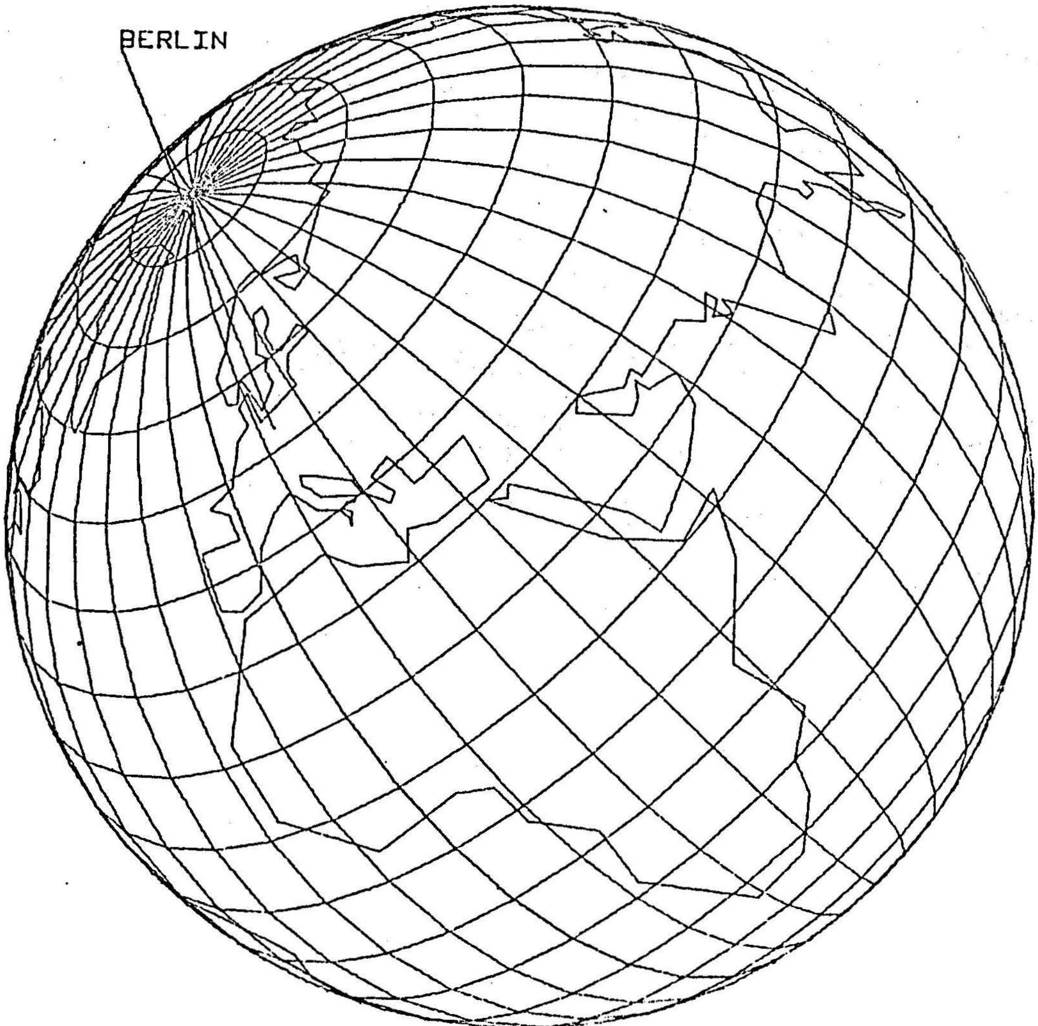
You cannot use this favourable dependence on smaller computers as for example C 90-40, for there is not enough storage space.

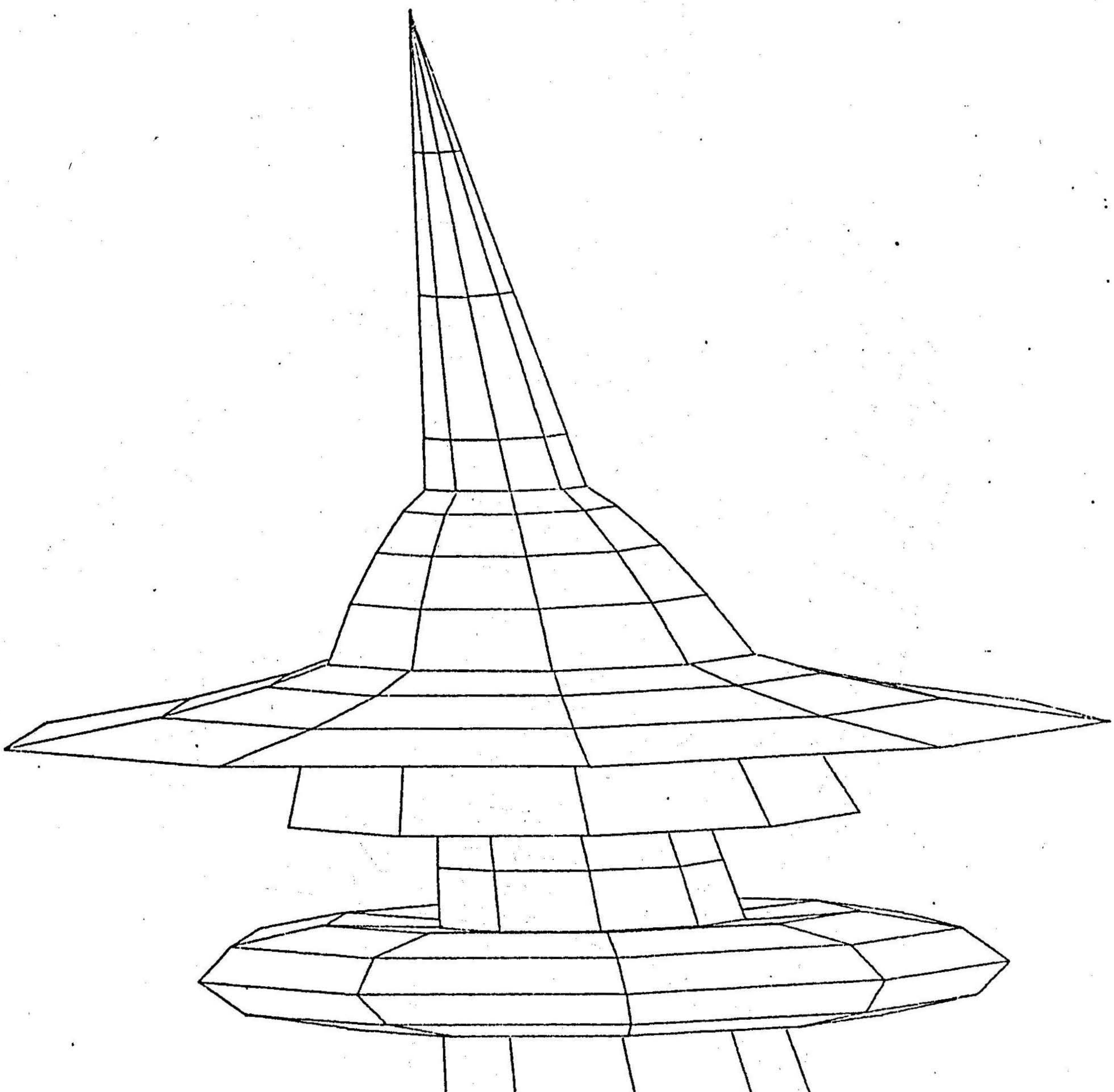
On large scale computers the use of this relation means a considerable time saving, and it is easily to realize by increasing the screen patch number resp. further division of the screen patches.

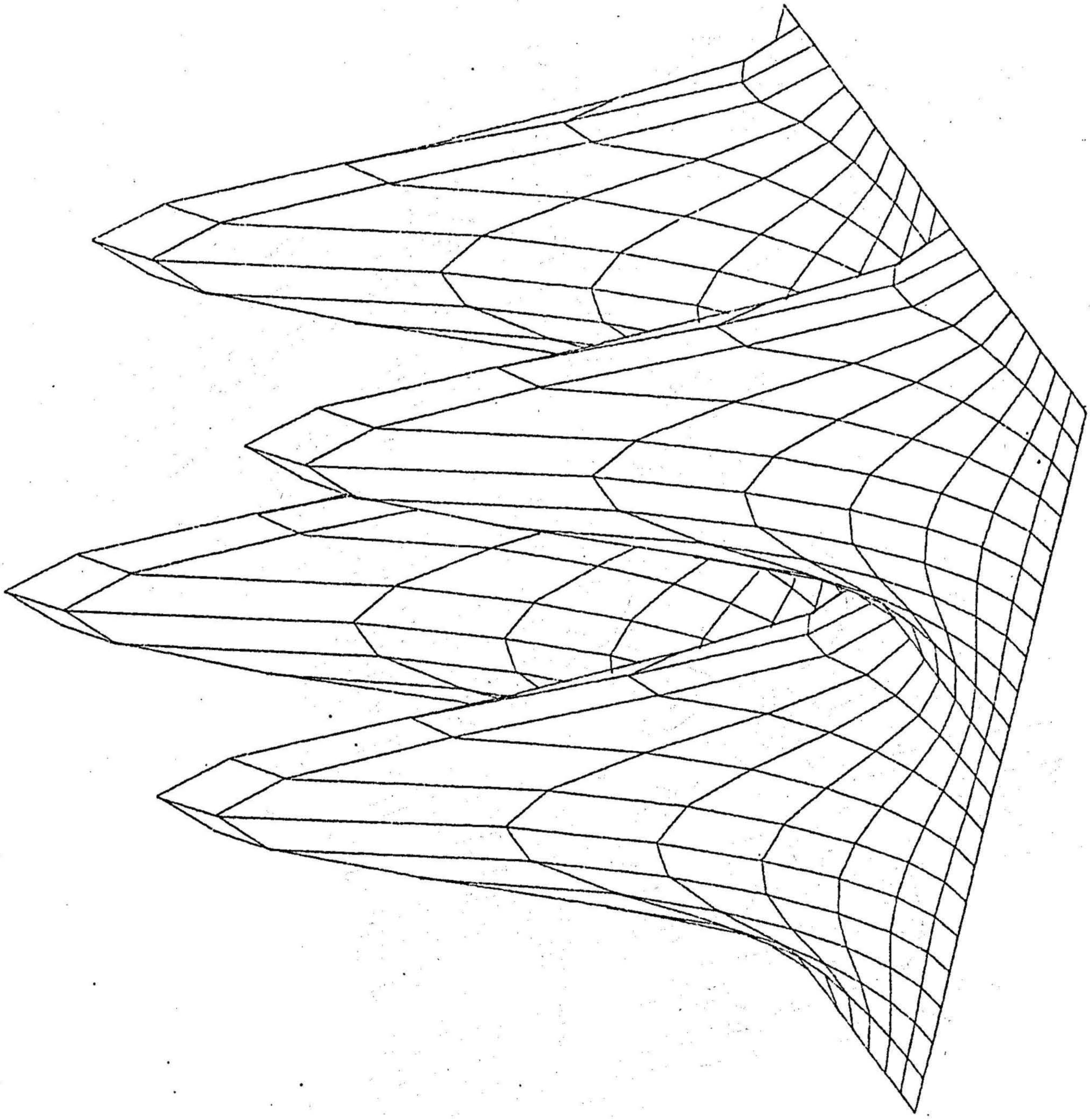
In its present form FLAVIS needs for 100 patches an average of 4 - 5 minutes.

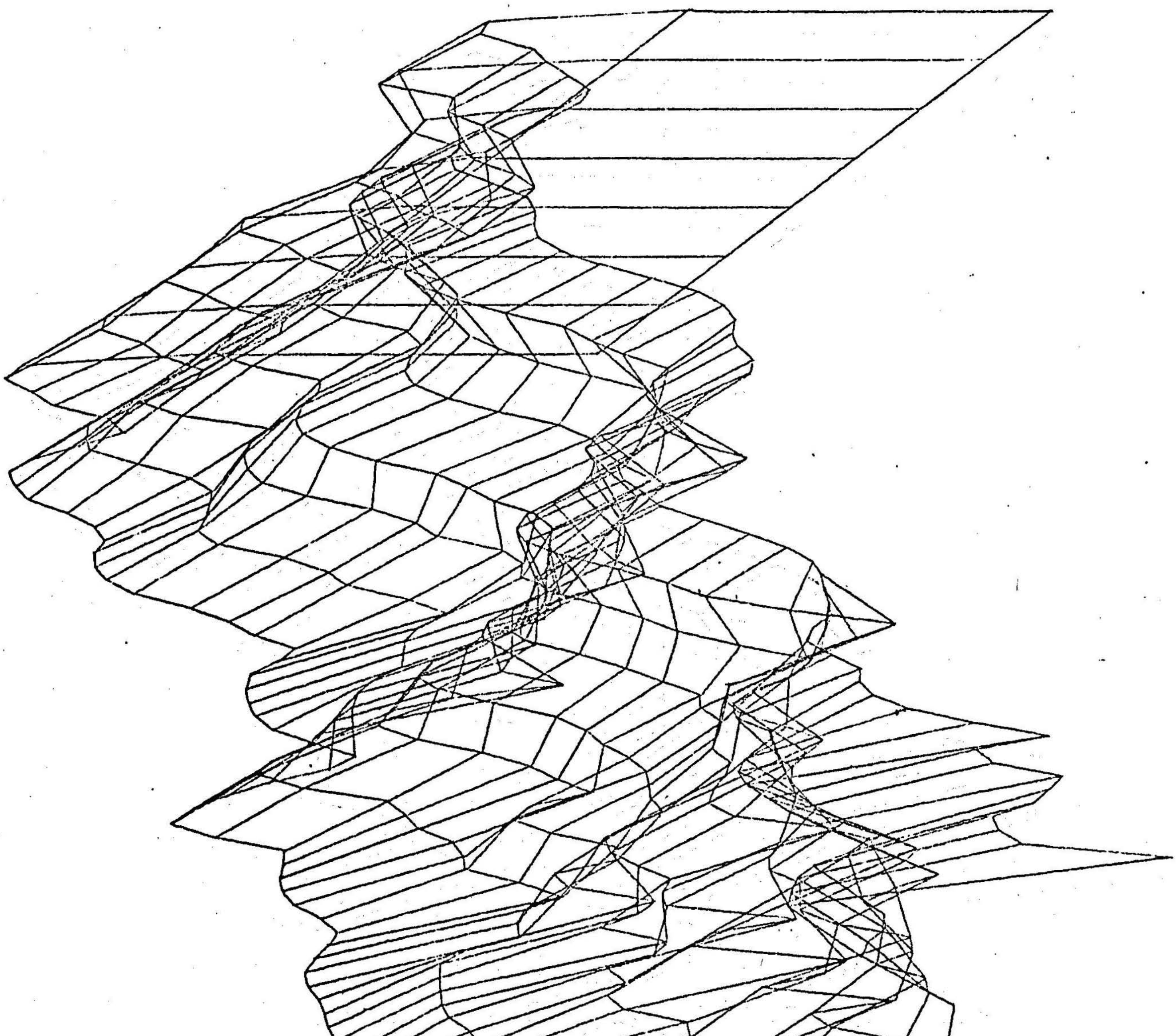
5. Picture examples for the results obtained by FLAVIS

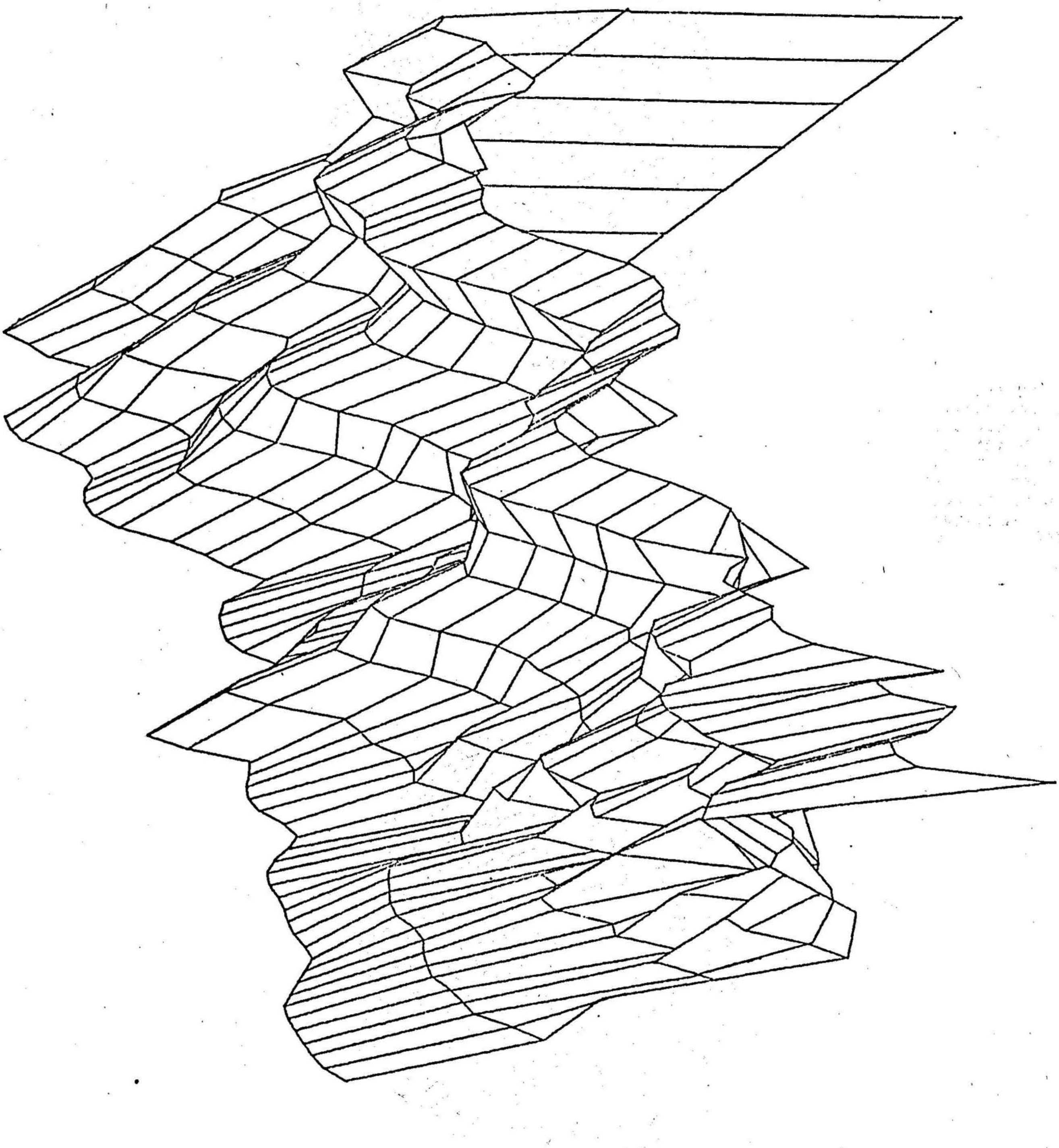

BERLIN

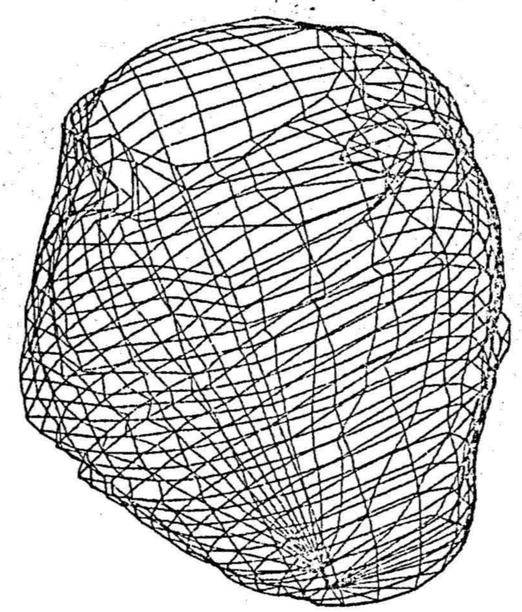

BERLIN



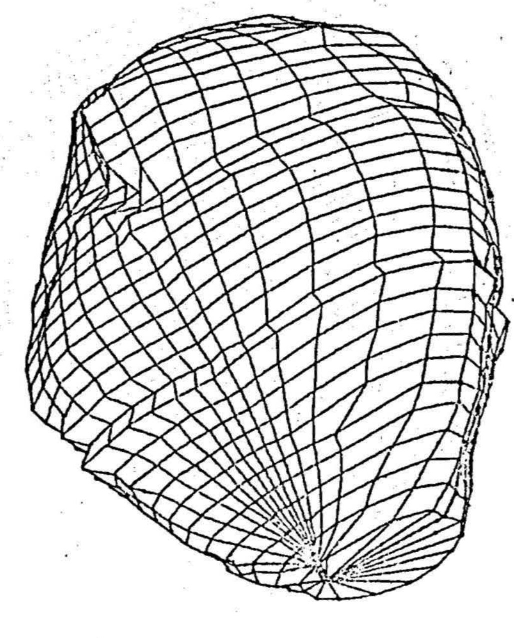




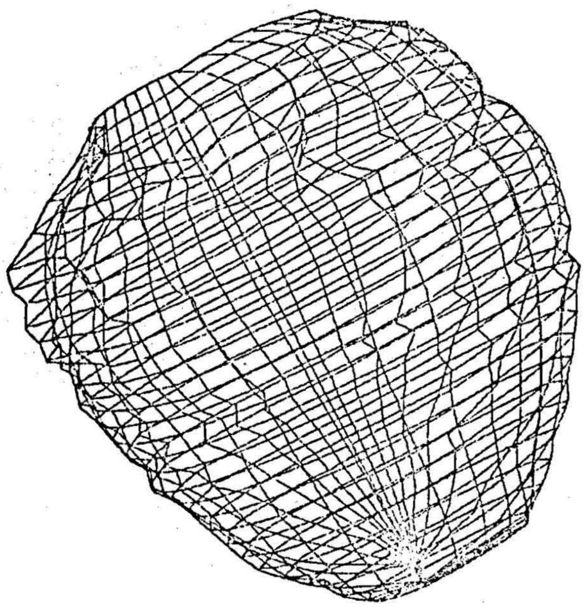




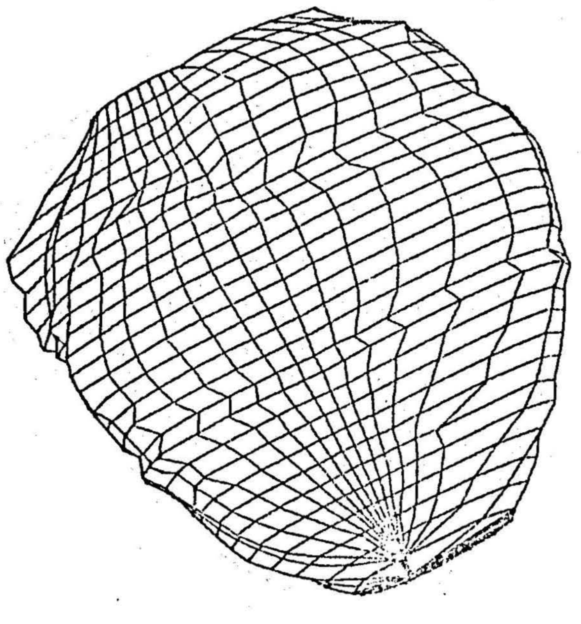
1.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 0 GRAD



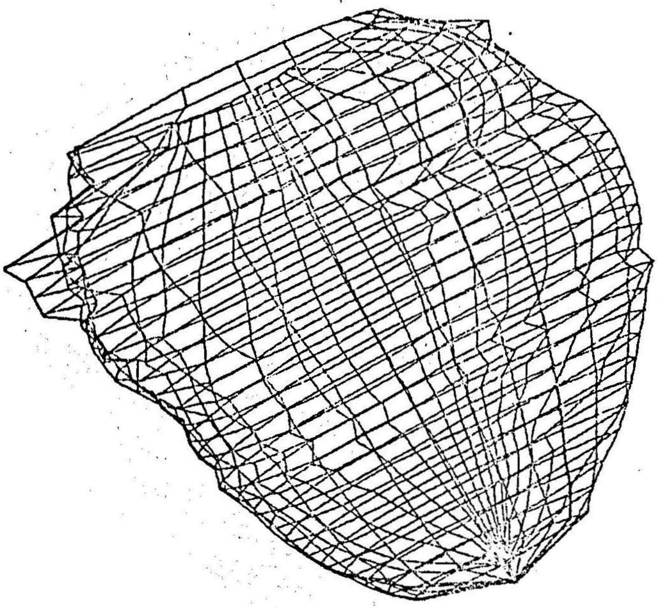
1.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 0 GRAD



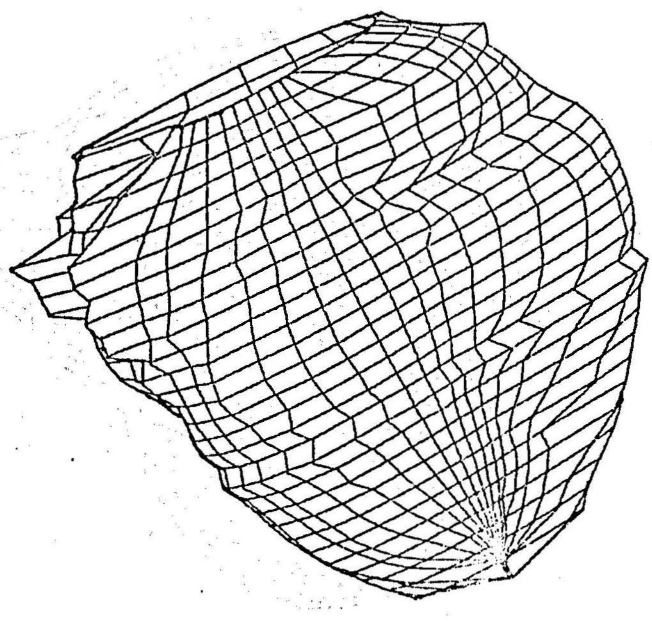
2.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 30 GRAD



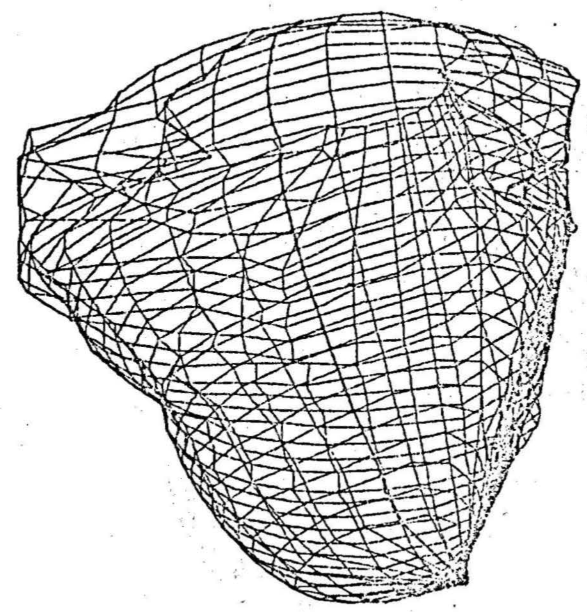
2.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 30 GRAD



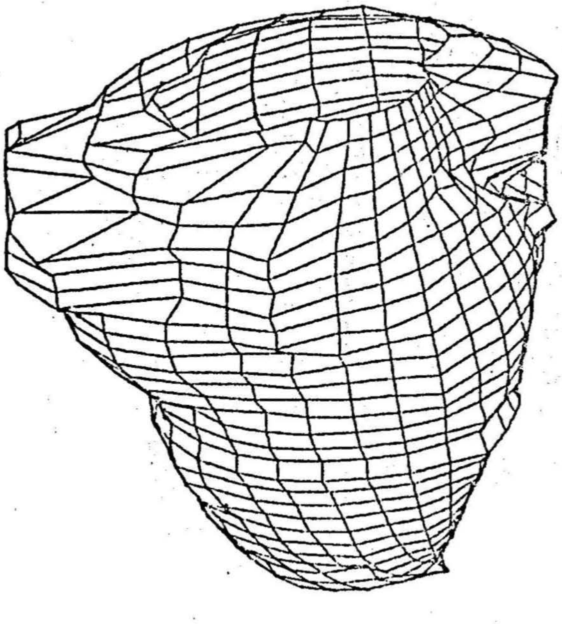
3.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 60 GRAD



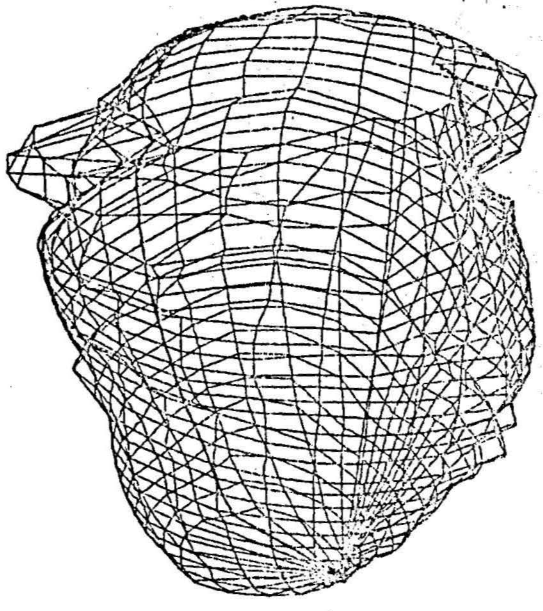
3.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 60 GRAD



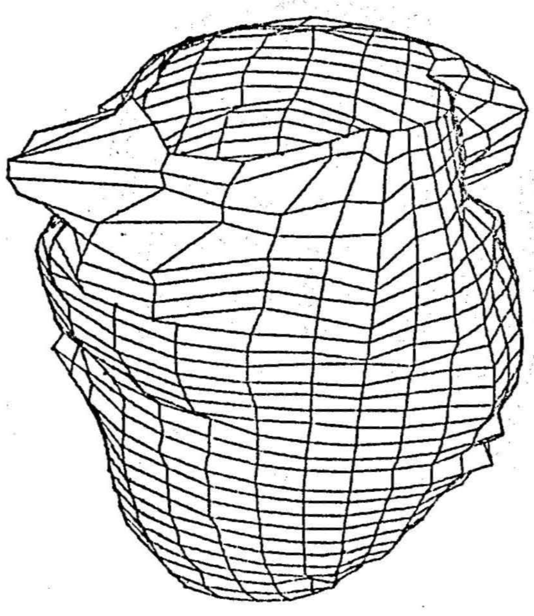
5.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 120 GRAD



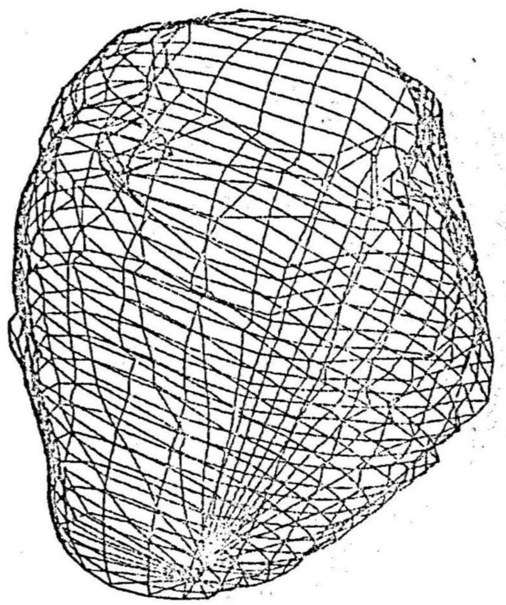
5.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 120 GRAD



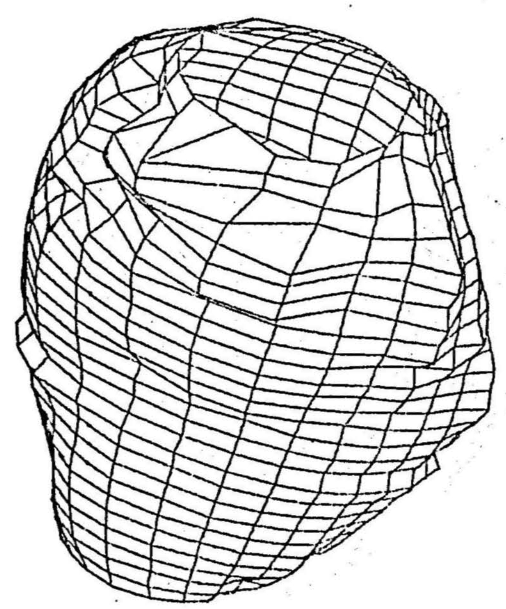
6.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 150 GRAD



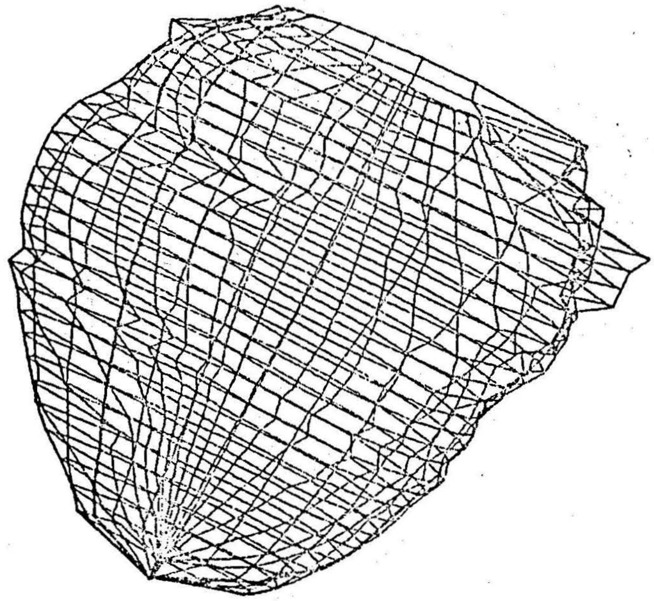
6.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 150 GRAD



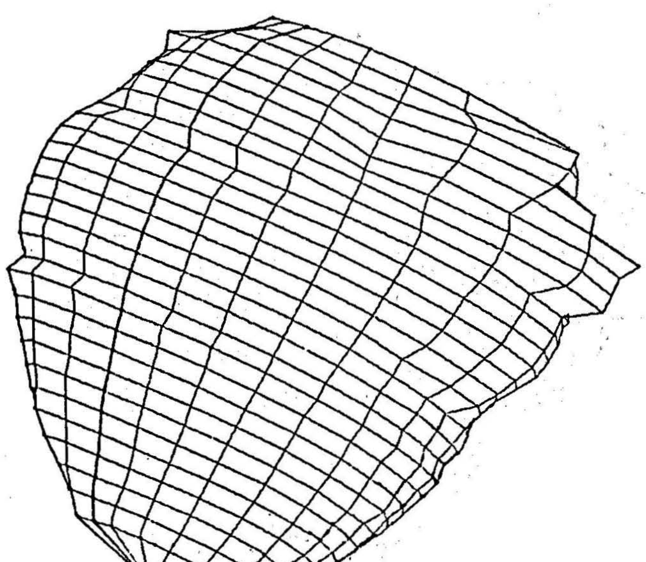
7•BILD:
ΔPHI=12.41 GRAD
ΔZ= 4•8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 180 GRAD



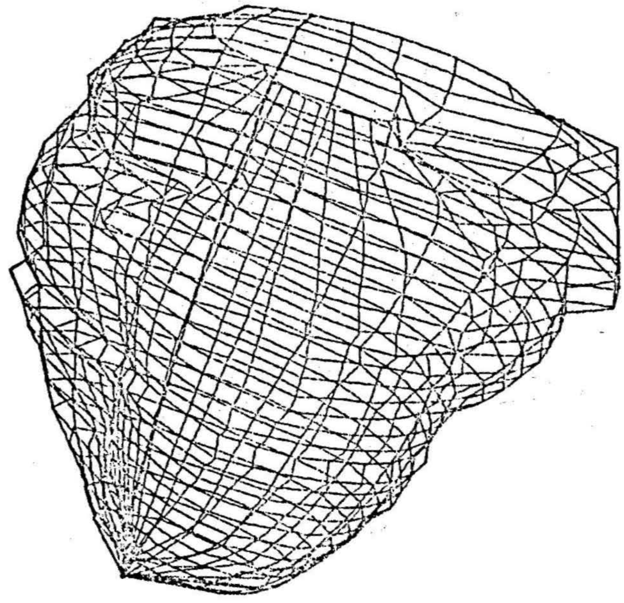
7•BILD:
ΔPHI=12.41 GRAD
ΔZ= 4•8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 180 GRAD



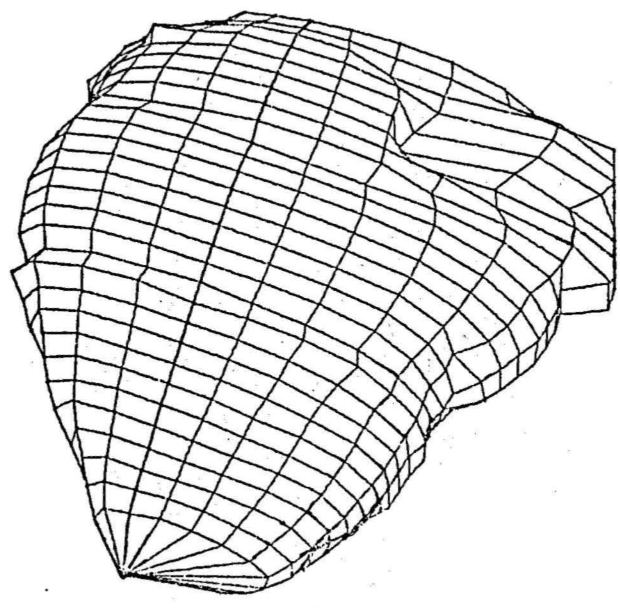
9•BILD:
ΔPHI=12•41 GRAD
ΔZ= 4•8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 240 GRAD



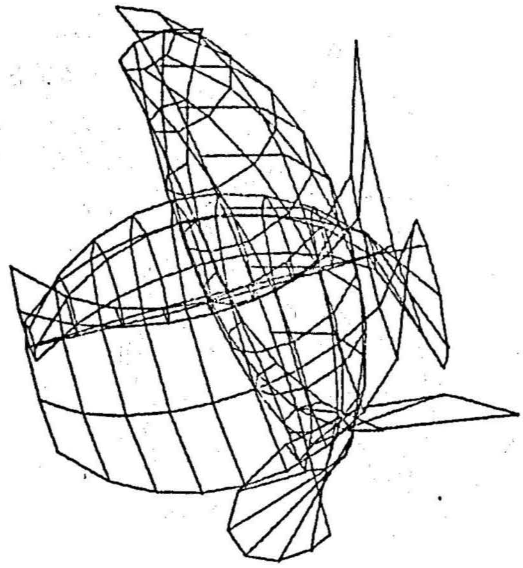
9•BILD:
ΔPHI=12•41 GRAD
ΔZ= 4•8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 240 GRAD

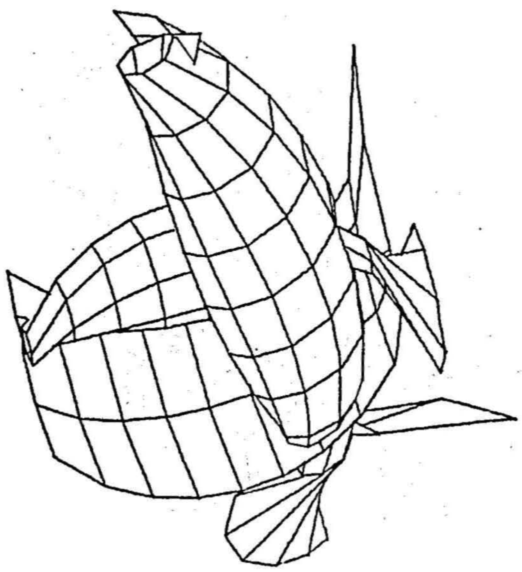


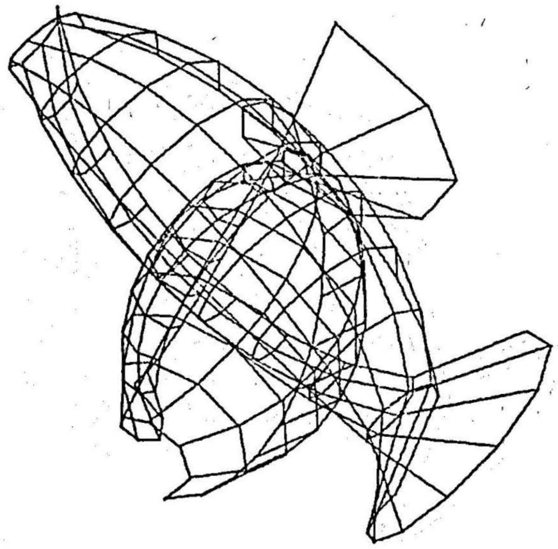
10.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 270 GRAD

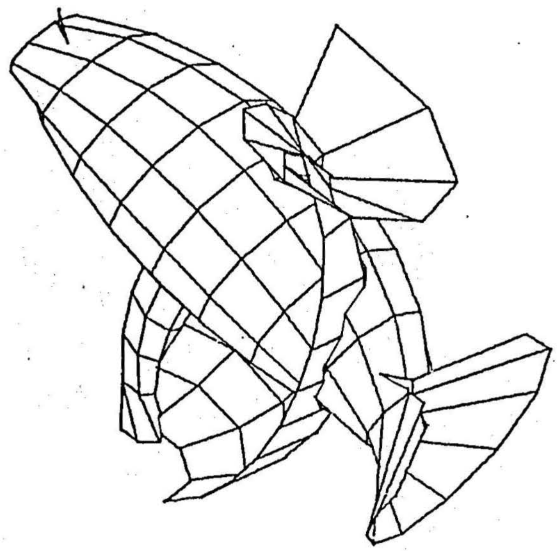


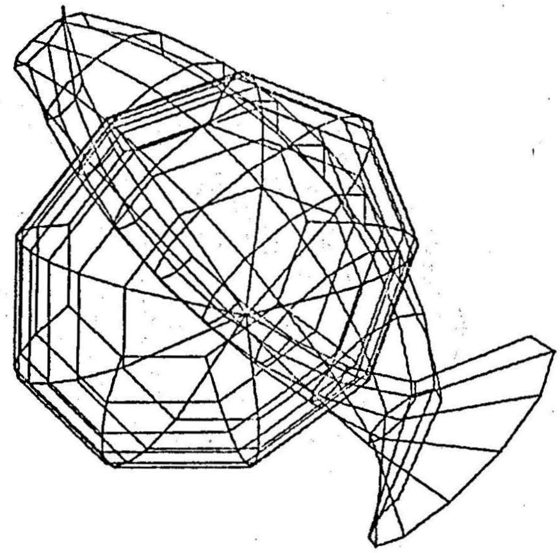
10.BILD:
ΔPHI=12.41 GRAD
ΔZ= 4.8 MM
STARTLAGE:
XW= 70 GRAD
YW= 0 GRAD
ZW= 20 GRAD
DREHUNG:
YW= 270 GRAD

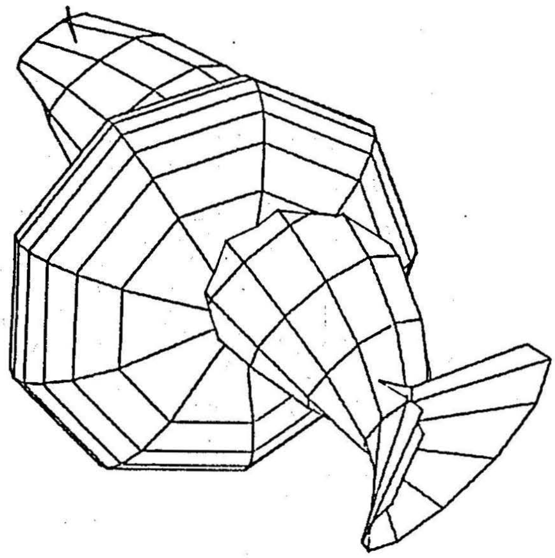


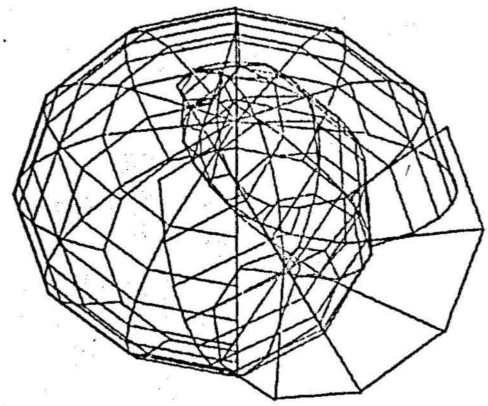


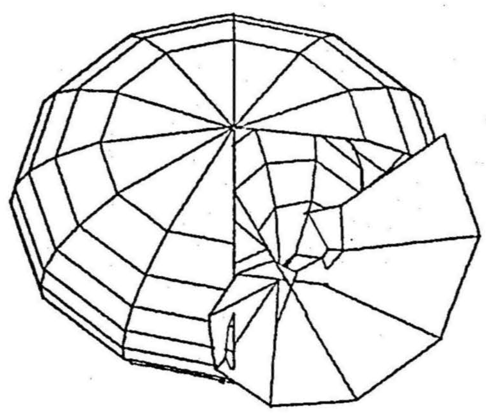


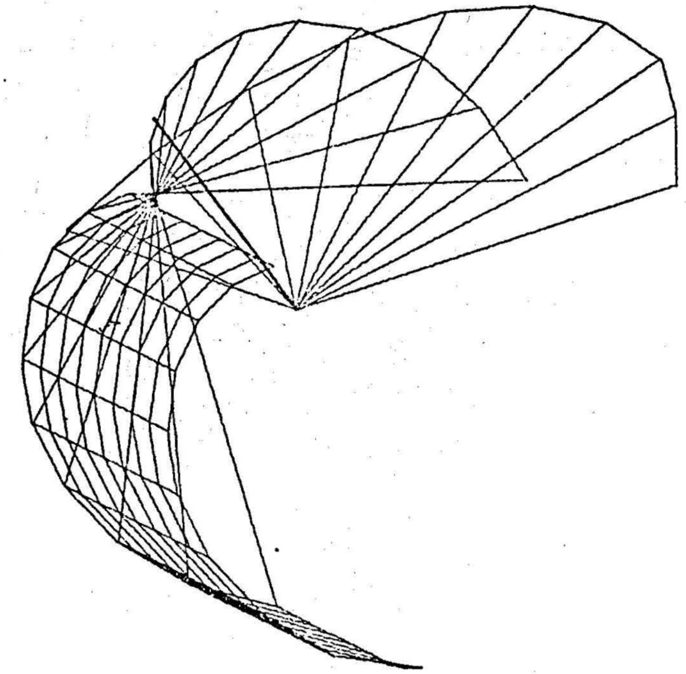


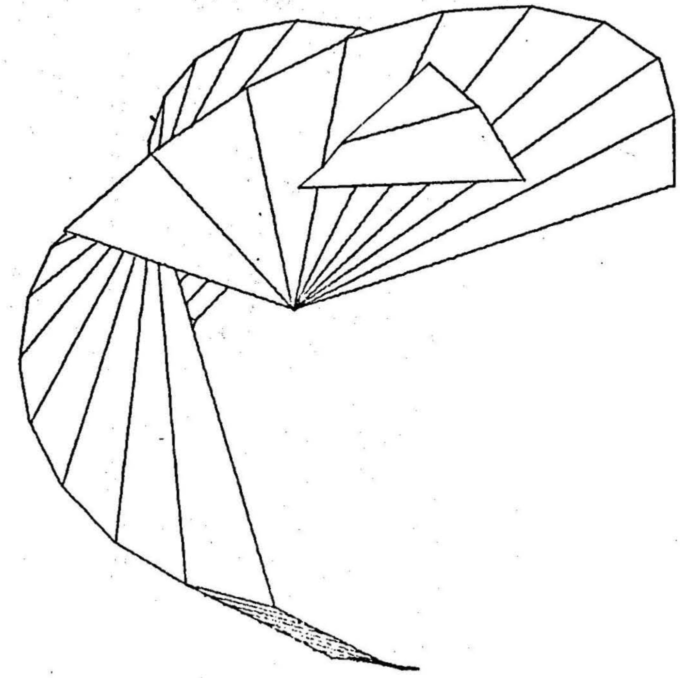


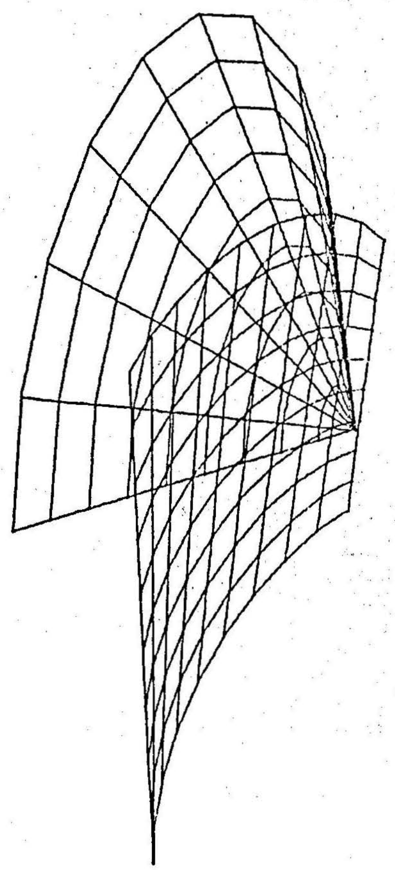


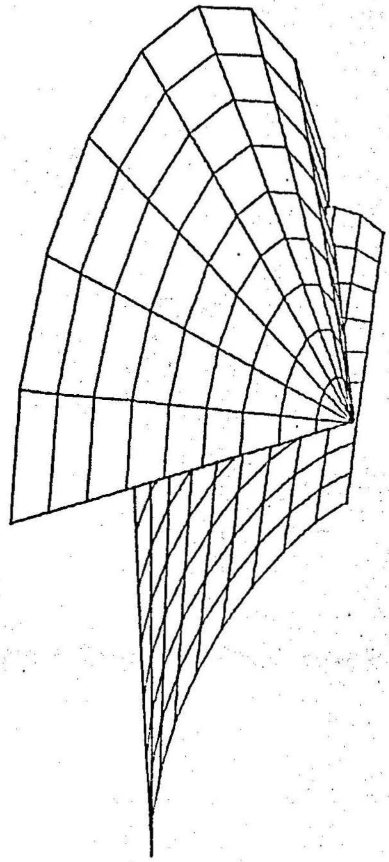


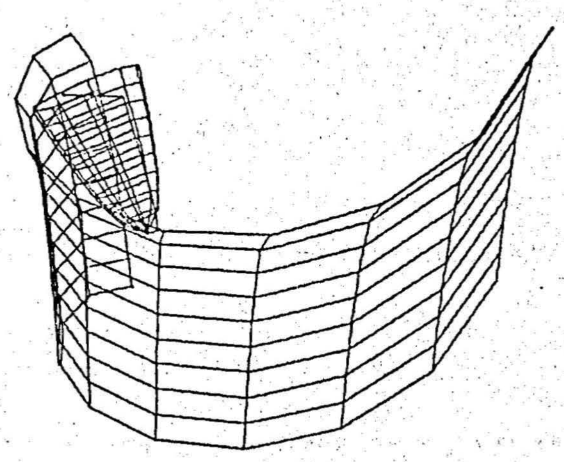


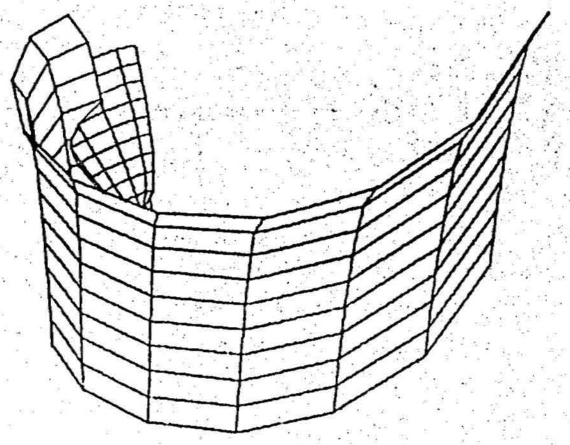












6. Listings

```
C1*********************************************************************C
C                                                                      C
C                                                                      C
C       FFFFF  L       AAA   V   V  I   SSS                            C
C       F      L      A   A  V   V  I  S   S                           C
C       F      L      A   A  V   V  I  S                               C
C       FFFF   L      AAAAA  V   V  I   SSS                            C
C       F      L      A   A  V   V  I      S                           C
C       F      L      A   A   V V   I  S   S                           C
C       F      LLLLL  A   A    V    I   SSS                            C
C                                                                      C
C                                                                      C
C                                                                      C
C**********************************************************************C
C                                                                      C
C                                                                      C
C     PROGRAM HIERARCHY OF FLAVIS                                      C
C                                                                      C
C                                                                      C
C**********************************************************************C
C                                                                      C
C                                                                      C
C                  **********                                          C
C     -------------*        *        /------/                          C
C     I            * FLAVIS *------->/VIDRAW/-----                     C
C     I       -----*        *        /------/    I                     C
C     I       I    **********           I        I                     C
C     I       I                         I        I                     C
C     I       I                         I        I                     C
C     V       V                         V        I                     C
C /------/ /------/                  /------/    I                     C
C /MINMAX/ /ASSIGN/                  /GREVIS/    I                     C
C /------/ /------/                  /------/    I                     C
C     ^       ^                         I        I                     C
C     I       I                         I        I                     C
C     I       I                         I        I                     C
C     I       I                         V        I                     C
C /------/ /------/                  /------/    I                     C
C /ZERLEG/ /INTER /<-----------------/VISKRI/<----                     C
C /------/ /------/                  /------/                          C
C     ^                               I I I                            C
C     I                               I I I        /------/            C
C     --------------------------------- I -------->/  DIF /            C
C                                       I          /ERROR /            C
C                                       I          /------/            C
C                                       V                              C
C                                    /------/                          C
C                                    /SEARCH/                          C
C                                    /------/                          C
C                                       I                              C
C                                       I                              C
C                                       I                              C
C                                       V                              C
C                                    /------/                          C
C                                    /INFLAS/                          C
C                                    /------/                          C
C                                                                      C
C                                                                      C
C**********************************************************************C
```

```
C2*****************************************************************C
C                                                                  C
C       EXPLANATIØN ØF THE SUBRØUTINES                             C
C                                                                  C
C                 'FØRTRAN 'ASSEMBL '                              C
C******************************************************************C
C                 '         '         '                            C
C       SUBR.     'USED     'USED     '  PURPØSE                   C
C                 'SUBR.    'SYSTEM-  '                            C
C                 '         'SUBR.    '                            C
C                 '         '         '                            C
C******************************************************************C
C                 '         '         '                            C
C       FLAVIS    'MINMAX   'PUTBIT   'TØ SETUP LISTS,TØ CØMPUTE CØNSTANTS   C
C                 'ASSIGN   'CLEAR    '                            C
C                 'VIDRAW   'SETUP    '                            C
C                 '         'GETANZ   '                            C
C                 '         '         '                            C
C       MINMAX    '  -      '  -      'MIN-,MAX-SEARCH IN SCREEN PATCH       C
C                 '         '         '                            C
C       ASSIGN    '  -      '  -      'TØ CØMPUTE BITFIELD FØR INDEX         C
C                 '         '         '                            C
C       VIDRAW    'VISKRI   'BLIND    'VISIBILITY, DRAWING THE VISIBLE PARTS C
C                 'GREVIS   'VECTR    'ØF THE PICTURE              C
C                 '         '         '                            C
C       GREVIS    'VISKRI   '  -      'TØ CØMPUTE THE LIMIT ØF VISIBILITY    C
C                 '         '         '                            C
C       VISKRI    'INTER    '  -      'VISIBILITY CRITERIØN        C
C                 'ZERLEG   '  -      '                            C
C                 'SEARCH   '  -      '                            C
C                 'DIF      '  -      '                            C
C                 '         '         '                            C
C       DIF       'ERRØR    '  -      'TØ CØMPUTE VISIB. ØF A TESTING PØINT  C
C       ERRØR     '  -      '  -      'TØ EXAMINE PRECISIØN ØF CØMPUTATIØN   C
C                 '         '         '                            C
C       INTER     'ASSIGN   'LØDBIT   'TØ CØMPUTE PATCH IN SCREEN PATCH      C
C                 '         '         '                            C
C       ZERLEG    'MINMAX   '  -      'TØ CØMPUTE NUMBER ØF PATCHES IN THE   C
C                 '         '         'DIVIDED SCREEN PATCH        C
C       SEARCH    '  -      'INFLAS   'SELECTIØN ØF PATCHES, CØNTAINING THE  C
C                 '         '         'TESTPØINT                   C
C******************************************************************C
C                                                                  C
C       EXPLANATIØN ØF THE SYSTEM-DEPENDING SUBRØUTINES            C
C                                                                  C
C******************************************************************C
C       SUBR.     'LANGU.   '  PURPØSE                             C
C                 '         '                                      C
C******************************************************************C
C       SETUP     'ASSEMBL 'DISPLAYFILE DEF., SCALATIØN            C
C       CLEAR     'ASSEMBL 'TØ ERASE DISPLAY PICTURE               C
C       BLIND     'ASSEMBL 'BLIND ADJUSTEMENT ØF PØINT             C
C       VECTR     'ASSEMBL 'TØ DRAW A VECTØR                       C
C       SHØW      'ASSEMBL 'TØ ØUTPUT DISPLAYFILE                  C
C       GETANZ    'ASSEMBL 'TØ CØMPUTE NUMBER ØF FILLED DISPLAYFILE-CELLS  C
C       PUTBIT    'ASSEMBL 'TØ FILLIN BITSAMPLES                   C
C       LØDBIT    'ASSEMBL 'ØUTPUT ØF BITSAMPLES                   C
C       INFLAS    'ASSEMBL 'INTERRØGATIØN,WHETHER TESTPØINT LIES IN PATCH  C
C                 '         '                                      C
C******************************************************************C
```

```
C                                                                          C
C3*************************************************************************C
C                                                                          C
C                                                                          C
C      SHORT SPECIFICATION OF THE FLAVIS-SUBROUTINE-PACKAGE                C
C      ====================================================                C
C                                                                          C
C                                                                          C
C**************************************************************************C
C                                                                          C
C                                                                          C
C                                                                          C
C      FIRST, IN THE FIRST  PART OF FLAVIS THE CONSTANTS ARE COMPUTED,     C
C      INDICATION- AND INDEX-LISTS ARE RESTORED.                           C
C      THAN,  IN A 2ND PART OF FLAVIS WITH THE SUBROUTINES MINMAX, ASSIGN
C      AND PUTBIT INDICATION- AND INDEX-LISTS ARE SET UP.                  C
C      IN A 3RD PART OF FLAVIS THE VISIBILITY STUDY IS STARTED AND         C
C      CONTROLLED. [SUBROUTINE VIDRAW]                                     C
C                                                                          C
C      VIDRAW CAUSES, IF NECESSARY, THE VISIBILITY STUDY OF THE KNOT       C
C      AND THAN TRANSFERS THE INITIAL- OR ENDPOINT OF A U- RESP. V-LINE-C
C      ELEMENT TO THE SUBROUTINE GREVIS. THE DISPLAYFILE THAN WILL BE      C
C      FILLED WITH THE RESULT OF GREVIS.                                   C
C                                                                          C
C      THE SUBROUTINE GREVIS TESTS THE LIMIT OF VISIBILITY WITH THE        C
C      SUBROUTINE VISKRI.                                                  C
C                                                                          C
C      THE SUBROUTINE VISKRI ASCERTAINS WITH THE PROGRAMS INTER, SEARCH C
C      ZERLEG AND DIF THE VISIBILITY OF THE TEST-POINT.                    C
C                                                                          C
C      THEREBY IT IS FOUND OUT WITH INTER, HOW MANY PATCHES ARE IN THE     C
C      SCREEN PATCH.                                                       C
C                                                                          C
C      WITH THE SUBROUTINE SEARCH IT IS ASCERTAINED WHICH PATCHES          C
C      INCLUDE THE RESPECTIVE TEST POINT [THIS IS DONE BY THE SUBROUTINEC
C      INFLAS:  IN FLAVIS WITH INFLIN THE ADDRESSES OF THE XC- AND YC-     C
C      LISTS AND THE ADDRESS OF THE RESULT PARAMETER ARE TRANSFERRED.      C
C      IN SEARCH FIRST WITH INFLAP, THE COO. OF THE TEST POINT ARE         C
C      TRANSFERRED. IN THE LOOP ONLY INFLA4 - WITH 4 INDICES OF THE        C
C      CORNER COO. OF A QUADRANGULAR PATCH RESP. INFLA3 WITH THE 3         C
C      INDICES OF A TRIANGULAR PATCH ARE CALLED].                          C
C      AFTER THAT THE PATCH IS PARTED IN TWO DIFFERENT WAYS INTO           C
C      TRIANGLES AND IT IS CALLED WHICH OF THEM INCLUDE THE TEST POINT     C
C      [WITH INFLA3]. THE RESULT IS STORED.                                C
C                                                                          C
C      IN DIF AND SEARCH TRIVIAL CASES ARE HANDLED PREVIOUSLY OR           C
C      EXCLUDED.                                                           C
C                                                                          C
C      BY THE SUBROUTINE ZERLEG - IF NESCESSARY - A DIVISION OF THE        C
C      SCREEN-PATCH IS MADE.                                               C
C                                                                          C
C      BY THE SUBROUTINE DIF IT IS ASCERTAINED WHETHER A TRIANGLE          C
C      COVERS THE TEST POINT OR NOT. POSSIBLE INACCURACIES IN COMPUTATION
C      ARE ELIMINATED BY THE SUBROUTINE ERROR.                             C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C                                                                          C
C**************************************************************************C
```

```
C4---------------------------------------------------------------------C
C                                                                      C
C                   LIST ØF THE CØMMØN-SIZES                           C
C                                                                      C
C       THE SUBR. ASSIGN AND ERRØR DØ NØT USE CØMMØN-SIZES.            C
C       THE SUBR. INFLAS IS AN ASSEMBLER-PRØGRAM AND USES THE CØMMØN-  C
C       SIZE IN.                                                       C
C       D = DEFINED ; U = USED ; EU = USED WITH EQUIVALENCE .          C
C                                                                      C
C----------------------------------------------------------------------C
C CØMMØN* HP*FLAVIS*VIDRAW*GREVIS*VISKRI*SEARCH*INTER*ZERLEG*MINMAX*DIF*
C----------------------------------------------------------------------C
C IN    *   *  D   *      *      *      *  D,U *     *      *      *   *
C XMIN  * D *  D,U *      *      *  U   *      *     *  U   *      *   *
C YMIN  * D *  D,U *      *      *  U   *      *     *  U   *      *   *
C IHQ   *   *      *      *      *  U   *      *  D  *      *      *   *
C INDLI *   *  D   *      *      *      *      *  U  *      *      *   *
C LIHIWI*   *  D,U *      *      *      *      *  U  *      *      *   *
C KEMAX * D *  U   *      *      *  U   *  U   *     *      *  U   *   *
C XT    * D *      *  U   *      *      *  EU  *     *      *  EU  * EU*
C YT    * D *      *  U   *      *      *  EU  *     *      *  EU  * EU*
C ZT    * D *      *  U   *      *      *      *     *      *      * EU*
C EPS   *   *  D   *  U   *  U   *      *  U   *     *      *      * U *
C DIVIS *   *  D   *      *  U   *      *      *     *      *      *   *
C III   *   *      *  U   *  D   *      *      *     *      *      *   *
C KZZ   *   *      *  U   *  D,U *      *      *     *      *      *   *
C ICA   *   *      *      *      *  U   *  D   *     *      *      *   *
C LSC   *   *      *      *      *  U   *  D   *     *      *      *   *
C MVC   *   *      *      *      *  U   *  D   *     *      *      *   *
C NRC   *   *      *      *      *  U   *  D   *     *      *      *   *
C I     *   *  D   *  U   *      *      *      *     *      *      *   *
C K     *   *  D   *  U   *      *      *      *     *      *      *   *
C L     *   *  D   *  U   *      *      *      *     *      *      *   *
C NPUF  *   *  D   *  U   *      *      *      *     *      *      *   *
C DELTAX*   *  D,U *      *      *  U   *      *     *      *      *   *
C DELTAY*   *  D,U *      *      *  U   *      *     *      *      *   *
C KWQ   *   *      *      *      *  D,U *      *     *      *      *   *
C LFQ   *   *      *      *      *  D,U *      *     *      *      *   *
C MBQ   *   *      *      *      *  D,U *      *     *      *      *   *
C IØPT1 *   *  D   *      *      *  U   *      *     *      *      *   *
C IØPT2 *   *  D   *      *      *  U   *      *     *      *      *   *
C IØPT3 *   *  D   *      *      *  U   *      *     *      *      *   *
C SIGMAX*   *  D   *      *      *  U   *      *     *      *      *   *
C SIGMAY*   *  D   *      *      *  U   *      *     *      *      *   *
C TEILX *   *  D   *      *      *  U   *      *     *      *      *   *
C TEILY *   *  D   *      *      *  U   *      *     *      *      *   *
C THETAX*   *  D   *      *      *  U   *      *     *      *      *   *
C THETAY*   *  D   *      *      *  U   *      *     *      *      *   *
C IE    * D *  U   *      *      *      *      *     *      *      *   *
C IEMAX * D *  U   *      *      *      *      *     *      *      *   *
C IQ    *   *  D,U *      *      *      *      *     *      *      *   *
C KE    * D *  U   *      *      *      *      *     *      *      *   *
C LEMAX * D *  U   *      *      *      *      *     *      *      *   *
C NANZAL* U *  D   *      *      *      *      *     *      *      *   *
C XMAX  * D *  D,U *      *      *      *      *     *      *      *   *
C YMAX  * D *  D,U *      *      *      *      *     *      *      *   *
C XC    * D *      *      *      *      *  U   *     *      *  U   * U *
C YC    * D *      *      *      *      *  U   *     *      *  U   * U *
C ZC    * D *      *      *      *      *      *     *      *      * U *
C----------------------------------------------------------------------C
```

```
C5*6*************************************************************C
C                                                                C
C                                                                C
C           SIGNIFICANCE OF THE COMMON-SIZES                     C
C                                                                C
C                                                                C
C****************************************************************C
C                                                                C
C****************************************************************C
C                                                                C
C                                                                C
C DELTAX: DISTANCE OF TWO X-SCREEN LINES                         C
C                                                                C
C DELTAY: DISTANCE OF TWO Y-SCREEN LINES                         C
C                                                                C
C DIVIS : MEASURE FOR THE ABSOLUTE SIZE OF ALL PATCHES IN THE X-,Y-PLANE
C                                                                C
C EPS   : EPSILON = 0.00001                                      C
C                                                                C
C I     : U-INDEX OF A KNOT                                      C
C                                                                C
C ICA   : LIST OF THE TRIANGLES GOING THROUGH THE POINTS WITH THE C
C         INDICES IQ,IQ+KEMAX,IQ+KEMAX+1 AND CONTAINING A POINT THAT C
C         IS TO BE STUDIED FOR VISIBILITY.                       C
C                                                                C
C IE    : NUMBER OF U-LINES                                      C
C                                                                C
C IEMAX : MAXIMUM NUMBER OF U-LINES                              C
C                                                                C
C IHQ   : LIST OF THE IQ-INDICES OF PATCHES, HAVING POINT QUANTITIES C
C         IN ONE SCREEN PATCH                                    C
C                                                                C
C III   : SHOWS, WHETHER 2 POINTS HAVE A SHORTER DISTANCE THAN EPS C
C                                                                C
C IKL   : FREE FOR EXTENSIONS                                    C
C                                                                C
C IN    : SHOWS, WHETHER A POINT LIES IN A PATCH                 C
C                                                                C
C INDLI : INDEX-LIST OF PATCHES                                  C
C                                                                C
C IM    : FREE FOR EXTENSIONS                                    C
C                                                                C
C IOPT1 : LIMIT FOR NUMBER OF PATCHES AT THE 1ST DIVISION OF THE C
C         SCREEN PATCH                                           C
C IOPT2 : LIMIT FOR NUMBER OF PATCHES AT THE 2ND DIVISION OF THE C
C         SCREEN PATCH                                           C
C IOPT3 : LIMIT FOR NUMBER OF PATCHES AT THE 3RD DIVISION OF THE C
C         SCREEN PATCH                                           C
C                                                                C
C IQ    : INDEX, COMBINING I-,K-,L-INDEX                         C
C                                                                C
C K     : V-INDEX OF A KNOT                                      C
C                                                                C
C KE    : NUMBER OF V-LINES                                      C
C                                                                C
C KEMAX : MAXIMUM NUMBER OF V-LINES                              C
C                                                                C
C KWQ   : LIST OF IQ-INDICES OF PATCHES HAVING POINT QUANTITIES IN C
C         A 1/4   OF THE SCREEN PATCH                            C
C                                                                C
```

```
C KZZ    : LOOP-COUNTER FOR LIMIT OF VISIBILITY                          C
C                                                                        C
C L      : INDEX OF SURFACE RELATIONSHIP WHEN THERE ARE SEVERAL SURFACESC
C                                                                        C
C LEMAX  : MAXIMUM NUMBER OF SURFACES                                     C
C                                                                        C
C LFQ    : LIST OF IQ-INDICES OF PATCHES, HAVING POINT QUANTITIES IN      C
C          1/16 OF THE SCREEN PATCH                                      C
C                                                                        C
C LIHIWI: INDICATION LIST FOR THE ALLOCATION OF PATCHES TO THE SCREEN    C
C                                                                        C
C LSC    : LIST OF TRIANGLES, GOING THROUGH THE POINTS WITH THE           C
C          INDICES IQ,IQ+1,IQ+KEMAX+1 AND CONTAINING A POINT THAT IS TO C
C          BE STUDIED FOR VISIBILITY                                     C
C                                                                        C
C MBQ    : LIST OF IQ-INDICES OF PATCHES, HAVING POINT QUANTITIES IN      C
C          1/64 OF THE SCREEN PATCH                                      C
C                                                                        C
C MVC    : LIST OF TRIANGLES, GOING THROUGH THE POINTS WITH THE           C
C          INDICES IQ+1,IQ+KEMAX,IQ+KEMAX+1 AND CONTAINING A POINT THAT C
C          IS TO BE STUDIED FOR VISIBILITY                               C
C                                                                        C
C NANZAL: NUMBER OF DISPLAY-FILE CELLS, WHICH ARE ALREADY FILLED         C
C                                                                        C
C NPUF   : DISPLAY-FILE                                                   C
C                                                                        C
C NRC    : LIST OF TRIANGLES, GOING THROUGH THE POINTS WITH THE           C
C          INDICES IQ,IQ+1,IQ+KEMAX AND CONTAINING A POINT THAT IS TO    C
C          BE STUDIED FOR VISIBILITY                                     C
C                                                                        C
C SIGMAX: ONE QUARTER OF THE DISTANCE BETWEEN 2 X-SCREEN LINES           C
C SIGMAY: ONE QUARTER OF THE DISTANCE BETWEEN 2 Y-SCREEN LINES           C
C                                                                        C
C TEILX : HALF DISTANCE BETWEEN 2 X-SCREEN LINES                         C
C TEILY : HALF DISTANCE BETWEEN 2 Y-SCREEN LINES                         C
C                                                                        C
C THETAX: ONE EIGHTH OF THE DISTANCE BETWEEN 2 X-SCREEN LINES            C
C THETAY: ONE EIGHTH OF THE DISTANCE BETWEEN 2 Y-SCREEN LINES            C
C                                                                        C
C XC     : =XT                                                           C
C                                                                        C
C XMAX   : MAXIMUM X-COORDINATE OF THE DISPLAY                           C
C                                                                        C
C XMIN   : MINIMUM X-COORDINATE OF THE DISPLAY                           C
C                                                                        C
C XT     : X-COORDINATE OF KNOTS                                         C
C                                                                        C
C YC     : =YT                                                           C
C                                                                        C
C YMAX   : MAXIMUM Y-COORDINATE OF THE DISPLAY                           C
C                                                                        C
C YMIN   : MINIMUM Y-COORDINATE OF THE DISPLAY                           C
C                                                                        C
C YT     : Y-COORDINATE OF KNOTS                                         C
C                                                                        C
C ZC     : =ZT                                                           C
C                                                                        C
C ZT     : Z-COORDINATE OF KNOTS                                         C
C                                                                        C
C**********************************************************************C
```

```
C7********************************************************************C
C                                                                     C
C      SIGNIFICANCE ØF THE LISTS ØF PATCH-INDICES                     C
C      ======================================                         C
C                                                                     C
C                                                                     C
C      IN LIHIWI[IRY,IRX,1] APPEARS THE NUMBER ØF PATCHES IN THE      C
C      SCREEN PATCH IRY,IRX.                                          C
C      IN LIHIWI[IRY,IRX,2] APPEARS THE RESPECTIVE START VALUE [INDEX] C
C      FØR THE INDLI-LIST.                                            C
C      IN INDLI[INDEX] APPEAR THE INDICES ØF THE PATCHES.             C
C                                                                     C
C*********************************************************************C
C                '         '                  '        '              C
C      LIST ØF   'SCREEN  'SCREENSIZE         'LIST-   'NUMBER ØF PATCHES C
C      PATCH     'PATCH   '                   'SIZE    'IN THE LIST   C
C      INDICES   '        '  ΔX   '   ΔY      '        '              C
C*********************************************************************C
C                '        '       '           '        '              C
C      IHQ       '1/1     'DELTAX'DELTAY      ' 200    'NIHQ          C
C                '        '       '           '        '              C
C      KWQ       '1/4     'TEILX 'TEILY       ' 100    'NKWQ          C
C                '        '       '           '        '              C
C      LFQ       '1/16    'SIGMAX'SIGMAY      '  50    'NLFQ          C
C                '        '       '           '        '              C
C      MBQ       '1/64    'THETAX'THETAY      '  40    'NMBQ          C
C                '        '       '           '        '              C
C*********************************************************************C
C                                                                     C
C      THERE IS:                                                      C
C      DELTAX = ABS[XMAX - XMIN]/11.                                  C
C      DELTAY = ABS[YMAX - YMIN]/11.                                  C
C      TEILX=DELTAX/2.                                                C
C      TEILY=DELTAY/2.                                                C
C      SIGMAX=DELTAX/4.                                               C
C      SIGMAY=DELTAY/4.                                               C
C      THETAX=DELTAX/8.                                               C
C      THETAY=DELTAY/8.                                               C
C                                                                     C
C*********************************************************************C
C                                                                     C
C      LISTS ØF THE DIVIDED PATCHES:                                  C
C                                                                     C
C*********************************************************************C
C                                                                     C
C      LIST ØF   ' INDICES ØF THE TRIANGLES       'NUMBER ØF   'LIST-SIZE C
C      PATCH     '                                'PATCHES IN  '      C
C      INDICES   ' 1.INDEX '2.INDEX    '3.INDEX   'THE LIST    '      C
C*********************************************************************C
C                '          '           '          '            '     C
C                '          '           '          '            '     C
C      ICA       'N         'N+KEMAX    'N+KEMAX+1 'NICA        '25   C
C                '          '           '          '            '     C
C      LSC       'N         'N+1        'N+KEMAX+1 'NLSC        '25   C
C                '          '           '          '            '     C
C      MVC       'N+1       'N+KEMAX+1  'N+KEMAX   'NMVC        '25   C
C                '          '           '          '            '     C
C      NRC       'N         'N+1        'N+KEMAX   'NNRC        '25   C
C                '          '           '          '            '     C
C*********************************************************************C
```

```
C8*******************************************************************C
C                                                                   C
C                                                                   C
C       SPECIFICATION OF DIMENSIONING:                              C
C       ==============================                              C
C                                                                   C
C                                                                   C
C*******************************************************************C
C                                                                   C
C                                                                   C
C                                                                   C
C       IN CASE MORE THAN 1000 KNOTS ARE TO BE OPERATED OR THE NUMBER OF C
C       KNOTS PER SURFACE SHOULD BE OTHER THAN 100, THE DIMENSIONING OF C
C       FOLLOWING SIZES HAS TO BE CHANGED:                          C
C                                                                   C
C                                                                   C
C       XT,YT,ZT,  XC,YC,ZC,  IE,KE,  INDLI,NPUF.                   C
C                                                                   C
C       YOU HAVE TO TAKE NOTICE OF THE FOLLOWING RULES:             C
C                                                                   C
C       IF                                                          C
C       XT[ID,KD,LD],      YT[ID,KD,LD]   AND   ZT[ID,KD,LD]        C
C                                                                   C
C       WITH ID = MAX. NUMBER OF KNOTS IN U-DIRECTION               C
C       WITH KD = MAX. NUMBER OF KNOTS IN V-DIRECTION               C
C            LD = MAX. NUMBER OF SURFACES                           C
C                                                                   C
C       THEN                                                        C
C                                                                   C
C       XC[ID*KD*LD], YC[ID*KD*LD]  AND  ZC[ID*KD*LD]               C
C                                                                   C
C       MOREOVER                                                    C
C                                                                   C
C       INDLI[ID*KD*LD + SECURITY-DISTANCE ]                        C
C                                                                   C
C       NPUF[ 2*[ID*KD*LD + SECURITY-DISTANCE]  ] HAVE TO BE SETUP. C
C                                                                   C
C       IN THE MAIN PROGRAM DEPENDENT ON THE DIMENSIONING THE FOLLOWING C
C       SIZES HAVE TO BE DEFINED:                                   C
C                                                                   C
C       IEMAX = ID                                                  C
C       KEMAX = KD                                                  C
C       LEMAX = LD                                                  C
C                                                                   C
C       ALL THE OTHER ARRAYS CAN REMAIN FIRMELY DIMENSIONED.        C
C                                                                   C
C                                                                   C
C*******************************************************************C
C                                                                   C
C                                                                   C
C       INDEPENDENT ON THE DIMENSIONING PRIOR TO A FLAVIS-CALL THE  C
C       FOLLOWING SIZES HAVE TO BE DEFINED IN THE MAIN PROGRAM:     C
C                                                                   C
C       XMIN,XMAX,YMIN,YMAX;                                        C
C       XT,YT,ZT   BZW. XC,YC,ZC                                    C
C       IE,KE,LEMAX.                                                C
C                                                                   C
C                                                                   C
C                                                                   C
C*******************************************************************C
```

```
C9*****************************************************************C
C                                                                  C
C                                                                  C
C       NECESSARY CHANGES FØR THE RUN ØF FLAVIS ØN ØTHER MACHINES  C
C       =========================================================  C
C       THAN C90/40:                                               C
C       ============                                               C
C                                                                  C
C                                                                  C
C******************************************************************C
C                                                                  C
C                                                                  C
C       IN FLAVIS:                                                 C
C       37  CALL ASSIGN[....             NØT APPLICABLE            C
C       38  CALL PUTBIT[...              NØT APPLICABLE            C
C       THEREFØRE                                                  C
C       37  INDLI[IANZ]=IQ                                         C
C                                                                  C
C       THE STATEMENTS                                             C
C       53  CALL CLEAR                                             C
C       54  CALL SETUP[...                                         C
C       67  CALL GETANZ[...                                        C
C       68  CALL CLEAR                                             C
C       HAVE TØ BE REPLACED FØR RESPECTIVE ØTHER SYSTEM-DEPENDENT  C
C       DISPLAY-PRØGRAMS ØR SIMILAR.                               C
C                                                                  C
C                                                                  C
C       THE SUBRØUTINE ASSIGN IS NØT APPLICABLE.                   C
C                                                                  C
C                                                                  C
C       IN VIDRAW:                                                 C
C       THE STATEMENTS                                             C
C       25  CALL BLIND[...                                         C
C       26  CALL VECTR[...                                         C
C       27  CALL SHØW[O.]                                          C
C       HAVE TØ BE SUBSTITUTED FØR RESPECTIVE ØTHER SYSTEM-DEPENDENT C
C       DISPLAY-PRØGRAMS ØR SIMILAR.                               C
C                                                                  C
C                                                                  C
C       IN INTER:                                                  C
C       8 CALL ASSIGN[...                NØT APPLICABLE            C
C       9 IHQ[NIHQ]=....                 NØT APPLICABLE            C
C       THEREFØRE                                                  C
C       8 IHQ[NIHQ]=INDLI[IANZ]                                    C
C                                                                  C
C                                                                  C
C       THE INFLAS SUBRØUTINE-PACKAGE [INFLIN,INFLAP,INFAL3,INFLA4] C
C       HAS TØ BE NEWLY WRITTEN FØR EACH MACHINE IN ASSEMBLER.     C
C                                                                  C
C                                                                  C
C       THE DIMENSIØNING INSTRUCTIØN FØR INDLI IS:                 C
C                                                                  C
C       INDLI[  4*[ID*KD*LD + SECURITY DISTANCE] ]                 C
C                                                                  C
C                                                                  C
C                                                                  C
C                                                                  C
C                                                                  C
C                                                                  C
C******************************************************************C
```

```
C**********************************************************************C
C                                                                      C
C                                                                      C
C       SUBROUTINE FLAVIS                                              C
C                                                                      C
C                                                                      C
C**********************************************************************C
C                                                                      C
C       IN THE 1ST PART OF THE SUBROUTINE FLAVIS THE CONSTANTS ARE     C
C       COMPUTED.                                                      C
C       IN THE 2ND PART THE INDICATION ARRAYS AND THE PATCH INDEX LIST C
C       ARE BUILT UP.                                                  C
C                                                                      C
C       IN THE 3RD PART THE VISIBLE PICTURE ELEMENTS ARE DRAWN.        C
C                                                                      C
C       CALL:     CALL FLAVIS                                          C
C       INPUTPARAMETERS:        -                                      C
C       OUTPUTPARAMETERS:       -                                      C
C       COMMON:                 DELTAX,DELTAY,DIVIS,EPS,I,IE,IEMAX,IN,INDLI,C
C                               IOPT1,IOPT2,IOPT3,IQ,K,KE,KEMAX,L,LEMAX,   C
C                               LIHIWI,NANZAL,NPUF,SIGMAX,SIGMAY,TEILX,TEILY,
C                               THETAX,THETAY,XMAX,XMIN,YMAX,YMIN          C
C       SUBROUTINES:            MINMAX,ASSIGN,VIDRAW                       C
C                                                                      C
C**********************************************************************C
        SUBROUTINE FLAVIS
C***********************************************************************
C
       DIMENSION XT[10,10,10],YT[10,10,10],ZT[10,10,10],INDLI[1024]
       DIMENSION XC[1000],YC[1000],ZC[1000],IE[10],KE[10]
       DIMENSION NPUF[2048]
       DIMENSION ICA[25],LSC[25],MVC[25],NRC[25]
       DIMENSION LIHIWI[11,11,2],IHQ[200],KWQ[100],MBQ[40],LFQ[50]
C***********************************************************************
       COMMON IN,IM,IKL
       COMMON XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT,ZT,EPS,DIVIS,III,KZZ
       COMMON ICA,LSC,MVC,NRC,I,K,L,NPUF,DELTAX,DELTAY,KWQ,LFQ,MBQ
       COMMON IOPT1,IOPT2,IOPT3,SIGMAX,SIGMAY,TEILX,TEILY,THETAX,THETAY
       COMMON IE,IEMAX,IQ,KE,LEMAX,NANZAL,XMAX,YMAX
C***********************************************************************
       EQUIVALENCE [XT,XC],[YT,YC],[ZT,ZC]
C***********************************************************************
C
C      RESET LISTS:
C
C******RESET LIHIWI-LIST:
C
       DO 1000 LDO=1,2
       DO 1000 KDO=1,11
       DO 1000 IDO=1,11
  1000 LIHIWI[IDO,KDO,LDO]=0
C
C******RESET INDLI-LIST:
C
       DO 2000 IDO=1,1024
C
  2000 INDLI[IDO]=0
C
C
C
```

```
C**********************************************************************
C**********************************************************************
C
C      CØMPUTE CØNSTANTS FØR NEW SCALATIØN
C
C**********************************************************************
     1 DIFFX=ABS[XMAX-XMIN]
     2 DIFFY=ABS[YMAX-YMIN]
C**********************************************************************
C
C      NEW SCALATIØN:
C
C**********************************************************************
     3 XMIN=XMIN-DIFFX/500.
     4 XMAX=XMAX+DIFFX/500.
     5 YMIN=YMIN-DIFFY/500.
     6 YMAX=YMAX+DIFFY/500.
C**********************************************************************
C
C      CØMPUTE SCREEN DISTANCES
C
C**********************************************************************
C      1          SCREEN UNIT
     7 DELTAX=DIFFX/11.
     8 DELTAY=DIFFY/11.
C      1/4        SCREEN UNIT
     9 TEILX=DELTAX/2.
    10 TEILY=DELTAY/2.
C      1/16       SCREEN UNIT
    11 SIGMAX=DELTAX/4.
    12 SIGMAY=DELTAY/4.
C      1/64       SCREEN UNIT
    13 THETAX=DELTAX/8.
    14 THETAY=DELTAY/8.
C**********************************************************************
C
C      CØMPUTE ØTHER CØNSTANTS:
C
C**********************************************************************
    15 DIVIS=[DIFFX+DIFFY]/700.
    16 IØPT1=8
    17 IØPT2=8
    18 IØPT3=16
    19 EPS=0.00001
C**********************************************************************
C
C      BUILD UP INDICATIØN LISTS
C
C**********************************************************************
    20 DØ 52 ISPR=1,2
    21 ISPRU=ISPR-1
C
C      SET BRANCH PARAMETER
C      ISPRU=0 : BUILD UP INDICATIØN ARRAYS [ 1ST RUN ]
C      ISPRU=1 : CØMPLETE INDICATIØN ARRAYS [ 2ND RUN ]
C
    22 IQ=0
    23   DØ 43 L=1,LEMAX
    24     DØ 41 I=1,IE[L]-1
    25       DØ 39 K=1,KE[L]-1
```

```
      26        IQ=IQ+1
C*****************************************************************
C     COMPUTATION OF THE MINIMUM AND MAXIMUM X- RESP. Y-VALUES FOR
C     EACH PATCH:
C
      27        CALL MINMAX[IQ,XMI,YMI,XMA,YMA]
C
C     XMI,YMI,XMA,YMA FORM A TRIANGLE, CONTAINING THE PATCH.THE
C     VARIABLES IRA,IRE; KRA,KRE ARE COMPUTED OUT OF IT, THEY INDICATE
C     OVER WHICH SCREEN PATCHES THE PATCH IS EXTENDED.
C
C
      28        IRA=IFIX[[XMI-XMIN]/DELTAX]+1
      29        IRE=IFIX[[XMA-XMIN]/DELTAX]+1
      30        KRA=IFIX[[YMI-YMIN]/DELTAY]+1
      31        KRE=IFIX[[YMA-YMIN]/DELTAY]+1
C
C
C     COMPUTE THE NUMBER OF PATCHES PER SCREEN PATCH
C     [IRX,IRY INDICATE THE SCREEN PATCH]:
C
      32          DO 39 IRY=KRA,KRE
      33            DO 39 IRX=IRA,IRE
C
C     COMPUTE OVER WHICH SCREEN PATCHES THE PATCH IS EXTENDED AND STORE
C     IN LIHIWI[IRY,IRX,1]
C
      34            LIHIWI[IRY,IRX,1]=LIHIWI[IRY,IRX,1]+1
C
C     BRANCH FOR THE 1ST RESP. 2ND RUN:
C
      35            IF[ISPRU] 36,39,36
C
C     COMPUTATION OF THE PATCH COUNTER IANZ:
C
      36            IANZ=LIHIWI[IRY,IRX,2]-1+LIHIWI[IRY,IRX,1]
C
C     COMPUTE THE VALUES FOR THE PROGRAMS PUTBIT AND LODBIT:
C
      37            CALL ASSIGN[IANZ,IZQ,IMIN,IMAX]
C
C     WITH THE PROGRAM PUTBIT[INDLI[IZQ],IMIN,IMAX,IQ] THE PATCH INDEX
C     IQ IS STORED IN THE CELL INDLI[IZQ] BETWEEN THE BITS IMIN AND
C     IMAX.
C
      38            CALL PUTBIT[INDLI[IZQ],IMIN,IMAX,IQ]
C
      39        CONTINUE
C
C     SKIP THE MEMORY LOCATIONS, WHICH DO NOT CORRESPOND TO THE KNOTS:
C
C
      40      IQ=IQ+KEMAX+1-KE[L]
      41      CONTINUE
C
C
C     SKIP THE MEMORY LOCATIONS, WHICH DO NOT CORRESPOND TO THE KNOTS:
C
      42    IQ=IQ+KEMAX*[IEMAX+1-IE[L]]
      43    CONTINUE
```

```
C
C*******************************************************************
C
C      CØMPLETE THE INDICATIØN LISTS:
C
C*******************************************************************
C
C      BRUNCH FØR THE 1ST RESP. 2ND RUN:
C
   44 IF[ISPRU] 53,45,53
   45 KHV=0
C
C
   46   DØ 52 IRY=1,11
   47     DØ 52 IRX=1,11
C
C      SUM THE NUMBER ØF PATCHES IN ALL SCREEN PATCHES AND STØRE TØ
C      LIHIWI[IRY,IRX,2] AND RESET LIHIWI[IRY,IRX,1] FØR THE 2ND RUN:
C
   48         IF[LIHIWI[IRY,IRX,1]] 49,52,49
   49         LIHIWI[IRY,IRX,2]=KHV+1
   50         KHV=KHV+LIHIWI[IRY,IRX,1]
   51         LIHIWI[IRY,IRX,1]=0
   52   CØNTINUE
C*******************************************************************
C
C
C      DRAWING ØF THE VISIBLE PICTURE ELEMENTS:
C
C*******************************************************************
C      CLEAR DISPLAY:
   53 CALL CLEAR
C
C      INITIALIZATIØN ØF THE INFLAS SUBRØUTINE-PACKAGE, THAT MEANS
C      ADDRESS-TRANSFER ØF THE XC-,YC-ARRAYS AND THE IN-PARAMETER:
C
        CALL INFLIN[XC[1],YC[1],IN]
C
C      DEFINITIØN ØF DISPLAY-FILE AND SCALATIØN ØF THE DISPLAY:
C
   54 CALL SETUP[NPUF[1],XMIN,XMAX,YMIN,YMAX]
C
C*****DRAW U-LINES WITH VISIBILITY:
C
   55   DØ 60 L=1,LEMAX
   56     DØ 60 K=1,IE[L]
   57       DØ 60 I=2,KE[L]
   58         I1=I-1
              CALL VIDRAW[I1,K,I]
   60   CØNTINUE
C
C*****DRAW V-LINES WITH VISIBILITY:
C
   61   DØ 66 L=1,LEMAX
   62     DØ 66 I=1,KE[L]
   63       DØ 66 K=2,IE[L]
   64         K1=K-1
              CALL VIDRAW[I,K1,K]
   66   CØNTINUE
C
```

```
C
C       ASCERTAIN THE NUMBER OF DISPLAY-FILE CELLS ALREADY FILLED:
     67 CALL GETANZ[NANZAL]
C
C       CLEAR DISPLAY:
     68 CALL CLEAR
C
C       RETURN TO THE MAIN PROGRAM:
  69    RETURN
C*************************************************************************
        END

OMMON ALLOCATION

77777 IN        77776 IM        77775 IKL       77773 XMIN
77771 YMIN      77461 IHQ       75461 INDLI     75077 LIHIWI
75076 KEMAX     71156 XT        65236 YT        61316 ZT
61314 EPS       61312 DIVIS     61311 III       61310 KZZ
61257 ICA       61226 LSC       61175 MVC       61144 NRC
61143 I         61142 K         61141 L         55141 NPUF
55137 DELTAX    55135 DELTAY    54771 KWQ       54707 LFQ
54637 MBQ       54636 IOPT1     54635 IOPT2     54634 IOPT3
54632 SIGMAX    54630 SIGMAY    54626 TEILX     54624 TEILY
54622 THETAX    54620 THETAY    54606 IE        54605 IEMAX
54604 IQ        54572 KE        54571 LEMAX     54570 NANZAL
54566 XMAX      54564 YMAX      71156 XC        65236 YC
61316 ZC

ROGRAM ALLOCATION

00022 LDO       00023 KDO       00024 IDO       00025 ISPR
00026 ISPRU     00027 IRA       00030 IRE       00031 KRA
00032 KRE       00033 IRY       00034 IRX       00035 IANZ
00036 IZQ       00037 IMIN      00040 IMAX      00041 KHV
00042 I1        00043 K1        00044 FLAVIS    00046 DIFFX
00050 DIFFY     00052 XMI       00054 YMI       00056 XMA
00060 YMA

UBPROGRAMS REQUIRED

ABS       MINMAX    IFIX      ASSIGN    PUTBIT    CLEAR
INFLIN    SETUP     VIDRAW    GETANZ
E END
```

```
C*********************************************************************C
C                                                                     C
C      SUBRØUTINE MINMAX                                              C
C                                                                     C
C                                                                     C
C*********************************************************************C
C                                                                     C
C      THE SUBRØUTINE MINMAX CØMPUTES THE SMALLEST AND LARGEST X-,Y-  C
C      CØØRDINATE ØF THE PATCH FIXED BY IMM.                          C
C                                                                     C
C                                                                     C
C      CALL:     CALL MINMAX[IMM,XMI,YMI,XMA,YMA]                     C
C      INPUTPARAMETERS:       IMM                                     C
C      ØUTPUTPARAMETERS:      XMI,YMI,XMA,YMA                         C
C      CØMMØN:                KEMAX,XC,YC                             C
C      SUBRØUTINES:           -                                       C
C                                                                     C
C*********************************************************************C
      SUBRØUTINE MINMAX[IMM,XMI,YMI,XMA,YMA]
C**********************************************************************
      DIMENSIØN XT[10,10,10],YT[10,10,10],XC[1000],YC[1000],INDLI[1024]
      DIMENSIØN IHQ[200],LIHIWI[11,11,2]
C**********************************************************************
      CØMMØN IN,IM,IKL,XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT
C**********************************************************************
      EQUIVALENCE [XT,XC],[YT,YC]
C**********************************************************************
    1 KM=IMM
    2 KMM=IMM+KEMAX-2
C******STARTING VALUE [INDEX = IMM] :
C
    3     XMI=XC[KM]
    4     YMI=YC[KM]
    5     XMA=XMI
    6     YMA=YMI
C
C******SEARCH FØR MINIMUM AND MAXIMUM:
C
C*****N=1: INDEX = IMM+1
C*****N=2: INDEX = IMM+KEMAX
C*****N=3: INDEX = IMM+KEMAX+1
C
C
    7 DØ 18 N=1,3
    8 KM=KM+N
    9   IF[XC[KM]-XMI] 10,11,11
   10     XMI=XC[KM]
   11   IF[YC[KM]-YMI] 12,13,13
   12     YMI=YC[KM]
   13   IF[XC[KM]-XMA] 15,15,14
   14     XMA=XC[KM]
   15   IF[YC[KM]-YMA] 17,17,16
   16     YMA=YC[KM]
   17 KM=KMM
   18 CØNTINUE
C******RETURN TØ THE SUBRØUTINE ZERLEG RESP. FLAVIS
   23 RETURN
C**********************************************************************
      END
```

ØMMØN ALLØCATIØN

77777 IN	77776 IM	77775 IKL	77773 XMIN
77771 YMIN	77461 IHQ	75461 INDLI	75077 LIHIWI
75076 KEMAX	71156 XT	65236 YT	71156 XC
65236 YC			

RØGRAM ALLØCATIØN

00016 MINMAX	00017 KM	DUMMY IMM	00020 KMM
00021 N	DUMMY XMI	DUMMY YMI	DUMMY XMA
DUMMY YMA			

HE END

```
C****************************************************************************
C
C
C       SUBRØUTINE ASSIGN
C
C
C****************************************************************************
C
C       THE SUBRØUTINE ASSIGN CØMPUTES FØR THE INDEX ØF A PATCH A BIT-
C       GRØUP DEFINED BY IMIN AND IMAX IN A WØRD, DEFINED BY IANZ.
C
C       CALL:     CALL ASSIGN[IANZ,IZQ,IMIN,IMAX]
C       INPUTPARAMETERS:         IANZ
C       ØUTPUTPARAMETERS:        IZQ,IMIN,IMAX
C       CØMMØN:                  -
C       SUBRØUTINES:             -
C
C****************************************************************************
        SUBRØUTINE ASSIGN[IANZ,IZQ,IMIN,IMAX]
C****************************************************************************
C******FIX THE WØRD
C
    1   IZQ=[IANZ+1]/2
C
C
C
C*****FIX A BIT GRØUP
C
    2   IELEMT=MØD[IANZ,2]
    3   IF[IELEMT] 7,4,7
C
    4   IMIN=12
    5   IMAX=22
C
C
C*****RETURN TØ INTER RESP. FLAVIS
    6   RETURN
C
    7   IMIN=1
    8   IMAX=11
C
C
C
C*****RETURN TØ INTER RESP. FLAVIS
    9   RETURN
C****************************************************************************
        END

RØGRAM ALLØCATIØN

 DUMMY IZQ       DUMMY IANZ      00011 IELEMT    DUMMY IMIN
 DUMMY IMAX      00012 ASSIGN

UBPRØGRAMS REQUIRED

 MØD
HE END
```

```
C*******************************************************************C
C                                                                   C
C                                                                   C
C      SUBRØUTINE VIDRAW                                            C
C                                                                   C
C                                                                   C
C*******************************************************************C
C                                                                   C
C      THE SUBRØUTINE VIDRAW TESTS THE KNØT P[XD1,YD1,ZD1] FØR VISIBILI-C
C      TY. THEN THE LIMIT ØF VISIBILITY IS TESTED WITH THE SUBRØUTINE C
C      GREVIS IN U- ØR V-DIRECTIØN. THE DISPLAY-FILE IS ACCØRDINGLY   C
C      FILLED AND THE PICTURE IS SHØWN ØN THE DISPLAY.                C
C                                                                   C
C      CALL:    CALL VIDRAW[I1,K1,IKANF]                            C
C      INPUTPARAMETERS:       I1,K1,IKANF                           C
C      ØUTPUTPARAMETERS:      -                                     C
C      CØMMØN:                EPS,I,III,K,KZZ,L,NPUF,XT,YT,ZT       C
C      SUBRØUTINES:           VISKRI,GREVIS                         C
C                                                                   C
C*******************************************************************C
      SUBRØUTINE VIDRAW[I1,K1,IKANF]
C********************************************************************
      DIMENSIØN XT[10,10,10],YT[10,10,10],ZT[10,10,10],INDLI[1024]
      DIMENSIØN NPUF[2048]
      DIMENSIØN ICA[25],LSC[25],MVC[25],NRC[25],LIHIWI[11,11,2],IHQ[200]
C********************************************************************
      CØMMØN IN,IM,IKL
      CØMMØN XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT,ZT,EPS,DIVIS,III,KZZ
      CØMMØN ICA,LSC,MVC,NRC,I,K,L,NPUF
C********************************************************************
C
C******IF I1 = I-1 AND K1 = K, THE U-LINES ARE TESTED FØR VISIBILITY.
C
C******IF I1 = I AND K1 = K-1, THE V-LINES ARE TESTED FØR VISIBILITY.
C
C******INITIAL PØINT:
    1  XD1=XT[I1,K1,L]
    2  YD1=YT[I1,K1,L]
    3  ZD1=ZT[I1,K1,L]
C*****END PØINT:
    4     XD2=XT[I,K,L]
    5     YD2=YT[I,K,L]
    6     ZD2=ZT[I,K,L]
    7     XYD2=XD2+YD2
C*****THE 1ST PØINT IS ØNLY TESTED FØR VISIBILITY AT THE BEGINNING
C*****ØF A U- RESP. V-LINE.
    8 IF[IKANF-2] 9,13,9
    9 IF[IVK] 13,15,21
C********************************************************************
C*****FIRST THE KNØT P[XD1,YD1,ZD1] IS TESTED FØR VISIBILITY.
C
C******IVK INDICATES, WHETHER THE KNØT IS VISIBLE ØR NØT:
C
C*****IVK=1: VISIBLE ; IVK=0: INVISIBLE.
C
   13 CALL VISKRI[XD1,YD1,ZD1,IVK]
   14 IF[IVK] 15,15,21
C
C********************************************************************
```

```
C***********************************************************************
C
C******FIRST POINT INVISIBLE:
C*****THE SUBROUTINE GREVIS COMPUTES THE LIMIT OF VISIBILITY.
C*****RETURN FROM GREVIS, IF VISIBLE POINT WAS FOUND.
C
   15 CALL GREVIS[-1,IVK,XD1,YD1,ZD1,XD2,YD2,ZD2,XSTR,YSTR,ZSTR]
C
C*****IF III > 0, THEN RETURN TO SUBROUTINE FLAVIS
C
   16 IF[III] 34,17,34
C
   17   XD1=XSTR
   18   YD1=YSTR
   19   ZD1=ZSTR
C******INTERROGATION, WHETHER THE END POINT IS REACHED:
   20 IF[ABS[XSTR+YSTR-XYD2]-EPS] 34,34,21
C
C
C***********************************************************************
C
C******FIRST POINT VISIBLE:
C*****THE SUBROUTINE GREVIS COMPUTES THE LIMIT OF VISIBILITY.
C*****RETURN FROM GREVIS, IF INVISIBLE POINT WAS FOUND.
C
   21 CALL GREVIS[0,IVK,XD1,YD1,ZD1,XD2,YD2,ZD2,XSTR,YSTR,ZSTR]
C
C*****IF III > 0, THEN RETURN TO  SUBROUTINE FLAVIS
C
C
   22 IF[III] 34,23,34
C
   23 IF[KZZ-1] 30,24,24
C
C*****INTERROGATION, WHETHER INITIAL POINT AND LAST POINT FOUND ARE
C        IN THE EPS-CRITERION.
   24 IF[ABS[XD1+YD1-XSTR1-YSTR1]-EPS] 26,26,25
C******FILLING OF THE DISPLAY-FILE:
C
   25         CALL BLIND[XD1,YD1]
   26         CALL VECTR[XSTR,YSTR]
   27         CALL SHOW[0.]
C
C***********************************************************************
C
   28 XSTR1=XSTR
   29 YSTR1=YSTR
C
   30   XD1=XSTR
   31   YD1=YSTR
   32   ZD1=ZSTR
C*****INTERROGATION, WHETHER THE END POINT IS REACHED:
   33 IF[ABS[XSTR+YSTR-XYD2]-EPS] 34,34,15
C
C***********************************************************************
C*****RETURN TO FLAVIS
C
   34 RETURN
      END
```

```
COMMON ALLOCATION

  77777 IN        77776 IM        77775 IKL       77773 XMIN
  77771 YMIN      77461 IHG       75461 INDLI     75077 LIHIWI
  75076 KEMAX     71156 XT        65236 YT        61316 ZT
  61314 EPS       61312 DIVIS     61311 III       61310 KZZ
  61257 ICA       61226 LSC       61175 MVC       61144 NRC
  61143 I         61142 K         61141 L         55141 NPUF

PROGRAM ALLOCATION

  DUMMY I1        DUMMY K1        DUMMY IKANF     00016 IVK
  00017 VIDRAW    00021 XD1       00023 YD1       00025 ZD1
  00027 XD2       00031 YD2       00033 ZD2       00035 XYD2
  00037 XSTR      00041 YSTR      00043 ZSTR      00045 XSTR1
  00047 YSTR1

SUBPROGRAMS REQUIRED

  VISKRI     GREVIS     ABS      BLIND      VECTR      SHOW

THE END
```

```
C***********************************************************************C
C                                                                       C
C                                                                       C
C      SUBRØUTINE GREVIS                                                C
C                                                                       C
C                                                                       C
C***********************************************************************C
C                                                                       C
C      THE SUBRØUTINE GREVIS CØMPUTES THE LIMIT ØF THE VISIBILITY.      C
C      THERE ARE:                                                       C
C      XD1,YD1,ZD1 = X-,Y-,Z-CØØRDINATES ØF THE 1ST PØINT               C
C      XD2,YD2,ZD2 = X-,Y-,Z-CØØRDINATES ØF THE 2ND PØINT               C
C      XSTR,YSTR,ZSTR = X-,Y-,Z-CØØRDINATES ØF THE LIMIT ØF VISIBILITY  C
C                                                                       C
C      CALL:CALL GREVIS[IWAHL,IVK,XD1,YD1,ZD1,XD2,YD2,ZD2,XSTR,YSTR,ZSTR]
C      INPUTPARAMETERS:      IWAHL,XD1,YD1,ZD1,XD2,YD2,ZD2              C
C      ØUTPUTPARAMETERS:     XSTR,YSTR,ZSTR,IVK                         C
C      CØMMØN:               DIVIS,EPS,III,KZZ                          C
C      SUBRØUTINES:          VISKRI                                     C
C                                                                       C
C***********************************************************************C
      SUBRØUTINE GREVIS[IWAHL,IVK,XD1,YD1,ZD1,XD2,YD2,ZD2,XSTR,YSTR,ZSTR
     1]
C************************************************************************
      DIMENSIØN XT[10,10,10],YT[10,10,10],ZT[10,10,10],INDLI[1024]
      DIMENSIØN LIHIWI[11,11,2],IHQ[200]
C************************************************************************
      CØMMØN IN,IM,IKL
      CØMMØN XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT,ZT,EPS,DIVIS,III,KZZ
C************************************************************************
    1 III=0
    2 KZZ=0
C******CØØ.-DIFFERENCE ØF THE 2 PØINTS:
    3 XD=XD2-XD1
    4 YD=YD2-YD1
    5 ZD=ZD2-ZD1
    6 XABS=ABS[XD]
    7 YABS=ABS[YD]
    8 BETRAG=XABS+YABS
    9 IF[BETRAG-EPS] 10,10,12
C
C*****III=1 MEANS THAT THE TWØ KNØTS ARE WITHIN AN EPS-CRITERIØN, I.E.
C*****XD=XD2-XD1=YD=YD2-YD1=0
   10 III=1
      IVK=-1
C
C
C******RETURN TØ SUBRØUTINE VIDRAW
   11 RETURN
C************************************************************************
C
   12 MBET=IFIX[BETRAG/DIVIS]
C
C*****IF MBET < 1, MBET = 1 IS SET.
C
   13 IF[MBET-1] 14,15,15
   14 MBET=1
C
   15 BET=FLØAT[MBET]
```

```
C******BRANCH, WHETHER X-COORD. IS VARIED AND Y- AND Z-COORD. ARE
C*****COMPUTED OR WHETHER Y-COORD. IS VARIED AND X- AND Z-COORD.
C*****ARE COMPUTED:
   17      IF[XABS-YABS] 18,18,30
C
C*****************************************************************C
C     IVK CAN ONLY BE 0 [TEST POINT INVISIBLE] OR 1 [TEST POINT VISIBLE]
C                                                                 C
C     IWAHL = -1: 1ST POINT INVISIBLE, RETURN, IF VISIBLE POINT WAS   C
C     FOUND:  IVK+IWAHL = 0 .                                     C
C                                                                 C
C     IWAHL =  0: 1ST POINT   VISIBLE, RETURN, IF INVISIBLE POINT WAS C
C     FOUND:  IVK+IWAHL = 0 .                                     C
C*****************************************************************C
C******Y-COORD. IS VARIED AND X- AND Z-COORD. ARE COMPUTED
C     XABS=ABS[XD2-XD1]=0, I.E. XD2-XD1=XD=0
C
   18      XYD=XD/YD
           ZYD=ZD/YD
           YBET=YD/BET
           YSTR=YD1
   19      YSTR=YSTR+YBET
   20      XSTR=[YSTR-YD1]*XYD+XD1
   21      ZSTR=[YSTR-YD1]*ZYD+ZD1
   22 CALL VISKRI[XSTR,YSTR,ZSTR,IVK]
C******UP TO MBET LOOPS
C*****RETURN TO SUBROUTINE VIDRAW, IF THE 1ST POINT IS VISIBLE AND
C     AN INVISIBLE POINT WAS FOUND OR IF THE 1ST POINT WAS INVISIBLE
C     AND A VISIBLE POINT WAS FOUND.
   23      IF[IVK+IWAHL] 26,44,26
   26      KZZ=KZZ+1
   27      IF[KZZ-MBET] 19,44,44
C
C******************************************************************
C******X-COORD. IS VARIED AND Y- AND Z-COORD. ARE COMPUTED
C     YABS=ABS[YD2-YD1]=0, I.E. YD2-YD1=YD=0
C
   30      YXD=YD/XD
           ZXD=ZD/XD
           XBET=XD/BET
           XSTR=XD1
   31      XSTR=XSTR+XBET
   32      YSTR=[XSTR-XD1]*YXD+YD1
   33      ZSTR=[XSTR-XD1]*ZXD+ZD1
   35 CALL VISKRI[XSTR,YSTR,ZSTR,IVK]
C******UP TO MBET LOOPS
C*****RETURN TO SUBROUTINE VIDRAW, IF THE 1ST POINT IS VISIBLE AND
C     AN INVISIBLE POINT WAS FOUND OR IF THE 1ST POINT WAS INVISIBLE
C     AND A VISIBLE POINT WAS FOUND.
   36      IF[IVK+IWAHL] 39,44,39
   39      KZZ=KZZ+1
   40      IF[KZZ-MBET] 31,44,44
C
C******************************************************************
C
C******RETURN TO SUBROUTINE VIDRAW
C
   44 RETURN
C*****************************************************************C
      END
```

```
CØMMØN ALLØCATIØN

  77777 IN           77776 IM          77775 IKL          77773 XMIN
  77771 YMIN         77461 IHQ         75461 INDLI        75077 LIHIWI
  75076 KEMAX        71156 XT          65236 YT           61316 ZT
  61314 EPS          61312 DIVIS       61311 III          61310 KZZ

PRØGRAM ALLØCATIØN

  DUMMY IVK          00034 MBET        DUMMY IWAHL        00035 GREVIS
  00037 XD           DUMMY XD2         DUMMY XD1          00041 YD
  DUMMY YD2          DUMMY YD1         00043 ZD           DUMMY ZD2
  DUMMY ZD1          00045 XABS        00047 YABS         00051 BETRAG
  00053 BET          00055 XYD         00057 ZYD          00061 YBET
  DUMMY YSTR         DUMMY XSTR        DUMMY ZSTR         00063 YXD
  00065 ZXD          00067 XBET

SUBPRØGRAMS REQUIRED

  ABS          IFIX         FLØAT       VISKRI
THE END
```

```
C*********************************************************************C
C                                                                     C
C                                                                     C
C     SUBRØUTINE VISKRI                                               C
C                                                                     C
C                                                                     C
C*********************************************************************C
C                                                                     C
C      THE SUBRØUTINE VISKRI TESTS WHETHER THE TEST PØINT P[A,B,C]    C
C      IS VISIBLE ØR INVISIBLE. A,B,C ARE THE X-,Y-,Z-CØØRDINATES     C
C      ØF THE TESTPØINT.                                              C
C                                                                     C
C     CALL:     CALL VISKRI[A,B,C,IVK]                                C
C     INPUTPARAMETERS:       A,B,C                                    C
C     ØUTPUTPARAMETERS:      IVK                                      C
C     CØMMØN:                DELTAX,DELTAY,ICA,IHQ,IØPT1,IØPT2,IØPT3, C
C                            KEMAX,KWQ,LFQ,LSC,MBQ,MVC,NRC,SIGMAX,SIGMAY,C
C                            TEILX,TEILY,THETAX,THETAY,XMIN,YMIN      C
C     SUBRØUTINES:           INTER,ZERLEG,SEARCH,DIF                  C
C                                                                     C
C*********************************************************************C
      SUBRØUTINE VISKRI[A,B,C,IVK]
C**********************************************************************
      DIMENSIØN XT[10,10,10],YT[10,10,10],ZT[10,10,10],INDLI[1024]
      DIMENSIØN NPUF[2048]
      DIMENSIØN ICA[25],LSC[25],MVC[25],NRC[25]
      DIMENSIØN LIHIWI[11,11,2],IHQ[200],KWQ[100],MBQ[40],LFQ[50]
C**********************************************************************
      CØMMØN IN,IM,IKL
      CØMMØN XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT,ZT,EPS,DIVIS,III,KZZ
      CØMMØN ICA,LSC,MVC,NRC,I,K,L,NPUF,DELTAX,DELTAY,KWQ,LFQ,MBQ
      CØMMØN IØPT1,IØPT2,IØPT3,SIGMAX,SIGMAY,TEILX,TEILY,THETAX,THETAY
C**********************************************************************
      AX=A-XMIN
      BY=B-YMIN
C*****CØMPUTE THE SCREEN PATCH
    1 IRX=IFIX[AX/DELTAX]+1
    2 IRY=IFIX[BY/DELTAY]+1
C******CØMPUTE 1/4  ØF THE SCREEN PATCH
    3 KDX=IFIX[AX/TEILX]+1
    4 KDY=IFIX[BY/TEILY]+1
C******CØMPUTE 1/16 ØF THE SCREEN PATCH
    5 LPX=IFIX[AX/SIGMAX]+1
    6 LPY=IFIX[BY/SIGMAY]+1
C******CØMPUTE 1/64 ØF THE SCREEN PATCH
    7 MGX=IFIX[AX/THETAX]+1
    8 MGY=IFIX[BY/THETAY]+1
C
C******INTERRØGATIØN, WHETHER THE NEW TEST PØINT HAS MØVED ØUT ØF THE
C      PREVIØUS TEST REGIØN:
C
    9      IF[IRX-IRXA] 17,10,17
   10      IF[IRY-IRYA] 17,11,17
   11      IF[KDX-KDXA] 18,12,18
   12      IF[KDY-KDYA] 18,13,18
   13      IF[LPX-LPXA] 20,14,20
   14      IF[LPY-LPYA] 20,15,20
   15      IF[MGX-MGXA] 22,16,22
   16      IF[MGY-MGYA] 22,43,22
```

```
C
C*****CØMPUTE THE PATCH IN THE SCREEN PATCH FØUND:
C
   17 CALL INTER[IRY,IRX,NIHQ]
   18      IF[NIHQ-IØPT1] 25,25,19
C
C*****DIVISIØN ØF THE SCREEN PATCH TØ 1/4  :
C
   19 CALL ZERLEG[NIHQ,IHQ,NKWQ,KWQ,TEILX,TEILY,KDX,KDY]
   20      IF[NKWQ-IØPT2] 29,29,21
C
C*****DIVISIØN ØF THE SCREEN PATCH TØ 1/16 :
C
   21 CALL ZERLEG[NKWQ,KWQ,NLFQ,LFQ,SIGMAX,SIGMAY,LPX,LPY]
   22      IF[NLFQ-IØPT3] 35,35,23
C
C*****DIVISIØN ØF THE SCREEN PATCH TØ 1/64 :
C
   23 CALL ZERLEG[NLFQ,LFQ,NMBQ,MBQ,THETAX,THETAY,MGX,MGY]
   24      GØ TØ 43
C
C*****CHØØSE THE PATCHES, CØNTAINING THE PØINT THAT HAS TØ BE TESTED.
C
C
C*****ØUT ØF A CØMPLETE SCREEN PATCH:
C
   25 CALL SEARCH[NIHQ,IHQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
   26 GØ TØ 50
C
C*****ØUT ØF A 1/4 ØF THE SCREEN PATCH:
C
   29 CALL SEARCH[NKWQ,KWQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
   30 GØ TØ 48
C
C*****ØUT ØF A 1/16 ØF THE SCREEN PATCH:
C
   35 CALL SEARCH[NLFQ,LFQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
   36 GØ TØ 46
C
C*****ØUT ØF A 1/64 ØF THE SCREEN PATCH:
C
   43 CALL SEARCH[NMBQ,MBQ,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
C
   44      MGXA=MGX
   45      MGYA=MGY
   46      LPXA=LPX
   47      LPYA=LPY
   48      KDXA=KDX
   49      KDYA=KDY
   50      IRXA=IRX
   51      IRYA=IRY
C
C*****ISPRU = -1: NØ CØVERING PØSSIBLE
C*****ISPRU =  0: TØTAL CØVERING
C*****ISPRU =  1: MØRE DETAILED TESTING NECESSARY
C
C
      IF[ISPRU]    94,511,513
  511 IVK=0
      RETURN
```

```
C*****EXCLUDE CØNCAVE PATCHES:
C
 513   IF[SENSESWITCH 2] 514,52
 514   IVK=1
       GØTØ 53
C
C
  52       IVK=2
C**************************************************************************
C*****INTERRØGATIØN, WHETHER ADDITIØNAL PATCHES CØVER THE PØINT P[A,B,C]
C      THAT HAS TØ BE TESTED:
  53       IF[NICA] 63,63,54
C*****THE TRIANGLES WITH THE NUMBER NICA, THE INDICES ØF WHICH ARE
C      STØRED IN THE ARRAY ICA, ARE TESTED WHETHER THEY CØVER THE
C      TESTPØINT. A TRIANGLE CØVERS P[A,B,C], IF DIFF < 0.
C
C
  54 DØ 62 MP=1,NICA
  55 I1=ICA[MP]
  56 I2=I1+KEMAX
  57 I3=I2+1
  58 CALL DIF[I1,I2,I3,DIFF,A,B,C]
  59       IF[DIFF] 60,62,62
C*****PØINT INVISIBLE
  60       IVK=IVK-1
C*****IF IVK=0, THEN RETURN TØ SUBRØUTINE GREVIS RESP. VIDRAW
  61       IF[IVK] 95,95,62
  62 CØNTINUE
C
C
C
C
C**************************************************************************
C*****INTERRØGATIØN, WHETHER ADDITIØNAL PATCHES CØVER THE PØINT P[A,B,C]
C      THAT HAS TØ BE TESTED:
  63       IF[NLSC] 73,73,64
C*****THE TRIANGLES WITH THE NUMBER NLSC, THE INDICES ØF WHICH ARE
C      STØRED IN THE ARRAY LSC, ARE TESTED WHETHER THEY CØVER THE
C      TESTPØINT. A TRIANGLE CØVERS P[A,B,C], IF DIFF < 0.
C
C
  64 DØ 72 MP=1,NLSC
  65 I1=LSC[MP]
  66 I2=I1+1
  67 I3=I2+KEMAX
  68 CALL DIF[I1,I2,I3,DIFF,A,B,C]
  69       IF[DIFF] 70,72,72
C*****PØINT INVISIBLE
  70       IVK=IVK-1
C*****IF IVK=0, THEN RETURN TØ SUBRØUTINE GREVIS RESP. VIDRAW
  71       IF[IVK] 95,95,72
  72 CØNTINUE
C
C
C
C
C
C
C
C
```

```
C***********************************************************************
C*****INTERROGATIθN, WHETHER ADDITIθNAL PATCHES CθVER THE PθINT P[A,B,C]
C       THAT HAS Tθ BE TESTED:
    73       IF[NMVC] 84,84,74
C*****THE TRIANGLES WITH THE NUMBER NMVC, THE INDICES θF WHICH ARE
C       STθRED IN THE ARRAY MVC, ARE TESTED WHETHER THEY CθVER THE
C       TESTPθINT. A TRIANGLE CθVERS P[A,B,C], IF DIFF < 0.
C
C
    74 Dθ 83 MP=1,NMVC
    75 I0=MVC[Mp]
    76 I1=I0+1
    77 I2=I0+KEMAX
    78 I3=I2+1
    79 CALL DIF[I1,I2,I3,DIFF,A,B,C]
    80       IF[DIFF] 81,83,83
C*****PθINT INVISIBLE
    81       IVK=IVK-1
C*****IF IVK=0, THEN RETURN Tθ SUBRθUTINE GREVIS RESP. VIDRAW
    82       IF[IVK] 95,95,83
    83 CθNTINUE
C
C
C
C
C
C
C
C
C***********************************************************************
C*****INTERRθGATIθN, WHETHER ADDITIθNAL PATCHES CθVER THE PθINT P[A,B,C]
C       THAT HAS Tθ BE TESTED:
    84       IF[NNRC] 94,94,85
C*****THE TRIANGLES WITH THE NUMBER NNRC, THE INDICES θF WHICH ARE
C       STθRED IN THE ARRAY NRC, ARE TESTED WHETHER THEY CθVER THE
C       TESTPθINT. A TRIANGLE CθVERS P[A,B,C], IF DIFF < 0.
C
C
    85 Dθ 93 MP=1,NNRC
    86 I1=NRC[MP]
    87 I2=I1+1
    88 I3=I1+KEMAX
    89 CALL DIF[I1,I2,I3,DIFF,A,B,C]
    90       IF[DIFF] 91,93,93
C*****PθINT INVISIBLE
C*****IF IVK=0, THEN RETURN Tθ SUBRθUTINE GREVIS RESP. VIDRAW
    91       IVK=IVK-1
    92       IF[IVK] 95,95,93
    93 CθNTINUE
C
C
C***********************************************************************
C
C*****PθINT VISIBLE
    94       IVK=1
C
C*****RETURN Tθ SUBRθUTINE GREVIS RESP. VIDRAW
    95 RETURN
C**********************************************************************C
       END
```

COMMON ALLOCATION

77777 IN	77776 IM	77775 IKL	77773 XMIN
77771 YMIN	77461 IHQ	75461 INDLI	75077 LIHIWI
75076 KEMAX	71156 XT	65236 YT	61316 ZT
61314 EPS	61312 DIVIS	61311 III	61310 KZZ
61257 ICA	61226 LSC	61175 MVC	61144 NRC
61143 I	61142 K	61141 L	55141 NPUF
55137 DELTAX	55135 DELTAY	54771 KWQ	54707 LFQ
54637 MBQ	54636 IOPT1	54635 IOPT2	54634 IOPT3
54632 SIGMAX	54630 SIGMAY	54626 TEILX	54624 TEILY
54622 THETAX	54620 THETAY		

PROGRAM ALLOCATION

00017 IRX	00020 IRY	00021 KDX	00022 KDY
00023 LPX	00024 LPY	00025 MGX	00026 MGY
00027 IRXA	00030 IRYA	00031 KDXA	00032 KDYA
00033 LPXA	00034 LPYA	00035 MGXA	00036 MGYA
00037 NIHQ	00040 NKWQ	00041 NLFQ	00042 NMBQ
00043 NICA	00044 NLSC	00045 NMVC	00046 NNRC
00047 ISPRU	DUMMY IVK	00050 MP	00051 I1
00052 I2	00053 I3	00054 IO	00055 VISKRI
00057 AX	DUMMY A	00061 BY	DUMMY B
DUMMY C	00063 DIFF		

SUBPROGRAMS REQUIRED

IFIX INTER ZERLEG SEARCH DIF

THE END

```
C**********************************************************************
C
C
C     SUBROUTINE DIF
C
C
C**********************************************************************
C
C     THE SUBROUTINE DIF COMPUTES OF THE EQUATION OF PLANES, THAT IS
C     SPANNED BY THE TRIANGLE XF1,YF1,ZF1; XF2,YF2,ZF2; XF3,YF3,ZF3,
C     THE Z-COORDINATE ZV AND COMPARES IT WITH C, THE Z-COORDINATE OF
C     THE TEST POINT. THE TEST POINT P[A,B,C] IS VISIBLE, IF OF THE
C     Z-COORDINATES OF THE TRIANGLE CONTAINING THE TEST POINT, ONLY
C     ONE IS LARGER THAN C. BEFORE THIS COMPUTATION IS MADE, THE
C     TRIVIAL CASES:
C     1] ALL Z-COO. OF THE TRIANGLE ARE LARGER THAN C [POINT INVISIBLE]
C     2] ALL Z-COO. OF THE TRIANGLE ARE SMALLER THAN C [POINT VISIBLE]
C     ARE TESTED AND EVALUATED.
C
C     CALL:    CALL DIF[I1,I2,I3,DIFF,A,B,C]
C     INPUTPARAMETERS:      I1,I2,I3,A,B,C
C     OUTPUTPARAMETERS:     DIFF
C     COMMON:               EPS,XC,YC,ZC
C     SUBROUTINES:          ERROR
C
C**********************************************************************
      SUBROUTINE DIF [I1,I2,I3,DIFF,A,B,C]
C**********************************************************************
      DIMENSION XT[10,10,10],YT[10,10,10],ZT[10,10,10],INDLI[1024]
       DIMENSION XC[1000],YC[1000],ZC[1000],IHQ[200],LIHIWI[11,11,2]
C**********************************************************************
      COMMON IN,IM,IKL
      COMMON XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT,ZT,EPS
C**********************************************************************
      EQUIVALENCE [XT,XC],[YT,YC],[ZT,ZC]
C**********************************************************************
C
C*****INTERROGATION, WHETHER ALL ZC<C OR ALL ZC>C:
C
C
C*****THE TESTPOINT IS VISIBLE, IF ALL ZC<C.
C
C*****THE TESTPOINT IS INVISIBLE, IF ALL ZC>C.
C
      IF[C-ZC[I1]+EPS] 4,1,1
    1 IF[C-ZC[I2]+EPS] 101,2,2
    2 IF[C-ZC[I3]+EPS] 101,3,3
C
C
C*****TEST POINT IS VISIBLE.
C
    3 DIFF=+10.
C
C*****RETURN TO VISKRI.
C
      RETURN
C
C
C**********************************************************************
```

```
    4  IF[C-ZC[I2]+EPS] 5,101,101
    5  IF[C-ZC[I3]+EPS] 6,101,101
C
C*****TEST POINT IS INVISIBLE.
C
    6  DIFF=-10.
C
C*****RETURN TO VISKRI.
       RETURN
C
C***********************************************************************
C
C*****TEST POINT HAS TO BE TESTED MORE PRECISELY:
C
C
C*****STORE CORNER COORDINATES:
C
  101  XF1=XC[I1]
  102  YF1=YC[I1]
  103  ZF1=ZC[I1]
C
C***********************************************************************
C
C      TO AVOID ROUNDING ERRORS, IN EACH COORDINATE DIFFERENCE IS
C      INTERROGATED, WHETHER IT HAS ALREADY REACHED THE SIZE OF EPS.
C      IF THAT IS THE CASE, THE DIFFERENCE IS SET TO 0.
C      [ SUBROUTINE ERROR ]
C
C***********************************************************************
C
C*****COMPUTATION OF INTERMEDIATE RESULTS:
C
       XX1=XC[I2]-XF1
C
         CALL ERROR[XX1,EPS]
C
       XX2=XC[I3]-XF1
C
         CALL ERROR[XX2,EPS]
C
       YY1=YC[I2]-YF1
C
         CALL ERROR[YY1,EPS]
C
       YY2=YC[I3]-YF1
C
         CALL ERROR[YY2,EPS]
C
       ZZ1=ZC[I2]-ZF1
C
         CALL ERROR[ZZ1,EPS]
C
       ZZ2=ZC[I3]-ZF1
C
         CALL ERROR[ZZ2,EPS]
C
C
C
C
C
```

```
C*********************************************************************
C
C       THE SIZE DIVID GIVES INFØRMATIØN ABØUT THE PØSITIØN ØF THE 3     C
C       CØRNERS P1[XF1,YF1,ZF1], P2[XF2,YF2,ZF2], P3[XF3,YF3,ZF3]       C
C       ØF THE TRIANGLE.                                                 C
C                                                                        C
C       DIVID = 0 MEANS:                                                 C
C                       1. P1 = P2                                       C
C       ØR              2. P1 = P3                                       C
C       ØR              3. P1,P2, AND P3  LIE ØN A LINE.                 C
C                                                                        C
C       IF ØNE ØF THE 3 CASES ØCCURED, THE EQUATIØN ØF PLANES IS NØ      C
C       LØNGER UNIQUELY DEFINED [PLANE IS DEGENERATED INTØ A LINE].      C
C       IN THIS CASE DIFF >0 [VISIBLE] IS SET.                           C
C                                                                        C
C*********************************************************************C
C
C
        DIVID=XX1*YY2-XX2*YY1
C
C
C
C*****INTERRØGATIØN, WHETHER DIVID = 0:
C
C
        IF[ABS[DIVID]-EPS] 3,3,116
C
C
C
C
C
C
C
C*****SØLUTIØN ØF THE EQUATIØN ØF PLANES:
C
  116   ZV=ZF1+[[B-YF1]*[XX1*ZZ2-XX2*ZZ1]-[A-XF1]*[YY1*ZZ2-YY2*ZZ1]]
     1      /DIVID
C
C
C
C
C
C
C
C*****A PLANE CØVERS THE TEST PØINT P[A,B,C], IF DIFF<0.
C
  117   DIFF=C-ZV+EPS
C
C
C
C
C
C
C
C*****RETURN TØ THE SUBRØUTINE VISKRI
C
  118   RETURN
C*********************************************************************C
        END
```

ΘMMΘN ALLΘCATIΘN

77777 IN	77776 IM	77775 IKL	77773 XMIN
77771 YMIN	77461 IHG	75461 INDLI	75077 LIHIWI
75076 KEMAX	71156 XT	65236 YT	61316 ZT
61314 EPS	71156 XC	65236 YC	61316 ZC

RΘGRAM ALLΘCATIΘN

DUMMY I1	DUMMY I2	DUMMY I3	00036 DIF
DUMMY C	DUMMY DIFF	00040 XF1	00042 YF1
00044 ZF1	00046 XX1	00050 XX2	00052 YY1
00054 YY2	00056 ZZ1	00060 ZZ2	00062 DIVID
00064 ZV	DUMMY B	DUMMY A	

UBPRΘGRAMS REQUIRED

ERRΘR ABS

HE END

```
C*************************************************************************
C
C
C       SUBROUTINE ERROR
C
C
C*************************************************************************
C
C       THE SUBROUTINE ERROR COMPUTES, WHETHER A SIZE VAL IS ABSOLUTELY
C       SMALLER THAN EPS. IF THAT IS THE CASE VAL =0 IS SET.
C
C
C       CALL:     CALL ERROR[VAL,EPS]
C       INPUTPARAMETERS:      VAL,EPS
C       OUTPUTPARAMETERS:     VAL
C       COMMON:               -
C       SUBROUTINES:          -
C
C*************************************************************************
        SUBROUTINE ERROR[VAL,EPS]
C*************************************************************************
C
        IF[ABS[VAL]-EPS] 1,1,2
      1 VAL=0.
C
C
      2 RETURN
C*****RETURN TO DIF
C*************************************************************************
        END
```

```
ROGRAM ALLOCATION

 00005 ERROR      DUMMY VAL        DUMMY EPS

UBPROGRAMS REQUIRED

 ABS
HE END
```

```
C*********************************************************************
C
C
C      SUBROUTINE INTER
C
C
C*********************************************************************
C
C      THE SUBROUTINE INTER COMPUTES THE PATCHES IN A SCREEN PATCH,
C      THAT IS FIXED BY THE INPUTPARAMETERS IRX,IRY. THE OUTPUTPARA-
C      METER NIHQ INDICATES THE NUMBER OF PATCHES SEARCHED, THE INDICES
C      OF WHICH ARE COLLECTED IN THE ARRAY IHQ.
C
C
C      CALL:    CALL INTER[IRY,IRX,NIHQ]
C      INPUTPARAMETERS:       IRY,IRX
C      OUTPUTPARAMETERS:      NIHQ
C      COMMON:                LIHIWI,INDLI,IHQ
C      SUBROUTINES:           ASSIGN
C
C*********************************************************************
       SUBROUTINE INTER[IRY,IRX,NIHQ]
C*********************************************************************
       DIMENSION IHQ[200],INDLI[1024],LIHIWI[11,11,2]
C*********************************************************************
       COMMON IN,IM,IKL,XMIN,YMIN,IHQ,INDLI,LIHIWI
C*********************************************************************
C
    1  NIHQ=0
C
C*****INDICATION ARRAY:
C
    2  LI1=LIHIWI[IRY,IRX,1]
    3  IF[LI1] 4,11,4
C
C*****START VALUE FOR EACH SCREEN PATCH:
C
    4  LI2=LIHIWI[IRY,IRX,2]
C
    5  DO 10 IL=1,LI1
C*****COMPUTATION OF THE PATCH-COUNTER IANZ:
C
    6  IANZ=LI2+NIHQ
    7  NIHQ=NIHQ+1
C
    8  CALL ASSIGN[IANZ,IZQ,IMIN,IMAX]
C
C*****LODBIT LOADS INTO THE CELL IHQ[NIHQ] THE INDEX, WHICH IS IN THE
C      CELL INDLI[IZQ] OF THE INDEX ARRAY BETWEEN BITS IMIN AND IMAX
C
    9  IHQ[NIHQ]=LODBIT[INDLI[IZQ],IMIN,IMAX]
C
   10  CONTINUE
C
C
C*****RETURN TO THE SUBROUTINE VISKRI
   11  RETURN
C*********************************************************************
       END
```

```
COMMON ALLOCATION

 77777 IN          77776 IM          77775 IKL         77773 XMIN
 77771 YMIN        77461 IHQ         75461 INDLI       75077 LIHIWI

PROGRAM ALLOCATION

 00012 INTER       DUMMY NIHQ        00013 LI1         DUMMY IRY
 DUMMY IRX         00014 LI2         00015 IL          00016 IANZ
 00017 IZQ         00020 IMIN        00021 IMAX

SUBPROGRAMS REQUIRED

 ASSIGN     LODBIT
THE END
```

```
C***********************************************************************C
C                                                                       C
C                                                                       C
C     SUBRØUTINE ZERLEG                                                 C
C                                                                       C
C                                                                       C
C***********************************************************************C
C                                                                       C
C     THE SUBRØUTINE ZERLEG CØMPUTES THE NUMBER MAR AND THE INDICES     C
C     ICA ØF THE PATCHES, LYING IN THE SCREEN PART THAT IS CREATED BY   C
C     THE DIVISIØN. MET IS THE NUMBER AND IZE ARE THE INDICES ØF THE    C
C     PATCHES, LYING IN THE DIVIDED SCREEN PATCH. CAPPAX,CAPPAY INDICATE
C     THE BREADTHS AND KVX,KVY INDICATE THE PØSITIØN ØF THE SCREEN      C
C     PART CREATED BY THE DIVISIØN.                                     C
C                                                                       C
C                                                                       C
C     CALL:     CALL ZERLEG[MET,IZE,MAR,IZA,CAPPAX,CAPPAY,KVX,KVY]      C
C     INPUTPARAMETERS:      MET,IZE,CAPPAX,CAPPAY,KVX,KVY               C
C     ØUTPUTPARAMETERS:     MAR,IZA                                     C
C     CØMMØN:               XMIN,YMIN                                   C
C     SUBRØUTINES:          MINMAX                                      C
C                                                                       C
C***********************************************************************C
      SUBRØUTINE ZERLEG[MET,IZE,MAR,IZA,CAPPAX,CAPPAY,KVX,KVY]
C***********************************************************************
      DIMENSIØN IZE[1],IZA[1]
C***********************************************************************
      CØMMØN IN,IM,IKL,XMIN,YMIN
C***********************************************************************
    1 MAR=0
    2 DØ 14 MP=1,MET
C*****CØMPUTATIØN ØF THE MINIMUM AND MAXIMUM VALUES ØF EACH PATCH IN
C     X- RESP. Y-DIRECTIØN:
C
    3   CALL MINMAX[IZE[MP],XMI,YMI,XMA,YMA]
    4   IRA=IFIX[[XMI-XMIN]/CAPPAX]+1-KVX
    5   IRE=IFIX[[XMA-XMIN]/CAPPAX]+1-KVX
    6   KRA=IFIX[[YMI-YMIN]/CAPPAY]+1-KVY
    7   KRE=IFIX[[YMA-YMIN]/CAPPAY]+1-KVY
C*****IF  IRA>KVX ØR  IRE<KVX THE PATCH CAN HAVE NØ PØINT QUANTITIES
C     IN THE SCREEN PART KVX, KVY.
C
    8   IF[IRA] 9,9,14
    9   IF[IRE] 14,10,10
C*****IF  KRA>KVX ØR  KRE<KVX THE PATCH CAN HAVE NØ PØINT QUANTITIES
C     IN THE SCREEN  PART KVX, KVY.
C
   10   IF[KRA] 11,11,14
   11   IF[KRE] 14,12,12
C*****NEXT PATCH:
C
   12   MAR=MAR+1
C*****INDEX ØF THE PATCH ØUT ØF THE ARRAY IZE --> IZA:
   13   IZA[MAR]=IZE[MP]
   14 CØNTINUE
C*****RETURN TØ THE SUBRØUTINE VISKRI
   15 RETURN
C***********************************************************************
      END
```

CØMMØN ALLØCATIØN

77777 IN	77776 IM	77775 IKL	77773 XMIN
77771 YMIN			

PRØGRAM ALLØCATIØN

DUMMY IZE	DUMMY IZA	DUMMY MAR	00024 MP
DUMMY MET	00025 IRA	DUMMY KVX	00026 IRE
00027 KRA	DUMMY KVY	00030 KRE	00031 ZERLEG
00033 XMI	00035 YMI	00037 XMA	00041 YMA
DUMMY CAPPAX	DUMMY CAPPAY		

SUBPRØGRAMS REQUIRED

MINMAX   IFIX

THE END

```
C*********************************************************************C
C                                                                     C
C                                                                     C
C      SUBROUTINE SEARCH                                              C
C                                                                     C
C                                                                     C
C*********************************************************************C
C                                                                     C
C      THE SUBROUTINE SEARCH SELECTS OF A NUMBER OF NIRS PATCHES THE  C
C      INDICES OF WHICH ARE IN THE ARRAY IRS, THOSE, CONTAINING THE   C
C      POINT WITH THE COORDINATES A,B.                                C
C                                                                     C
C      FIRST WITH THE SUBROUTINE INFLAP[A,B] THE X-, Y-COORDINATES OF C
C      THE TEST POINT ARE DELIVERED. [IN FLAVIS BY THE SUBROUTINE     C
C      INFLIN[XC[1],YC[1],IN] THE BEGINNING ADDRESSES OF THE XC-, YC- C
C      ARRAYSAND THE RESULT PARAMETER WERE DELIVERED.]                C
C      THEREFORE BY INFLA4[INDEX1,INDEX2,INDEX3,INDEX4] AND           C
C      INFLA3[INDEX1,INDEX2,INDEX3] ONLY THE INDICES OF THE PATCH COR- C
C      NERS ARE DELIVERED.                                            C
C                                                                     C
C      BEFORE INFLA4 IS CALLED, THE TRIVIAL CASE, THAT THE PATCH LIES C
C      UNIQUELY BEHIND P IS EXCLUDED. PRIOR TO THE ACTUAL DIVISION INTO C
C      2 TRIANGLES, THE TRIVIAL CASE, THAT THE PATCH LIES UNIQUELY IN C
C      FRONT OF P, IS TESTED. IF THAT IS THE CASE, ISPRU IS SET TO ZERO C
C      AND IT IS RETURNED TO VISKRI. ISPRU=0 MEANS IN VISKRI THAT THE C
C      TEST POINT IS INVISBLE.                                        C
C                                                                     C
C                                                                     C
C      BY THESE PRE-TESTINGS UNNECESSARY INFLA4- RESP. INFLA3-CALLS ARE C
C      AVOIDED.                                                       C
C                                                                     C
C      CALL:      CALL SEARCH[NIRS,IRS,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]  C
C      INPUTPARAMETERS:     A,B,C,NIRS,IRS                            C
C      OUTPUTPARAMETERS:    NICA,NLSC,NMVC,NNRC,ISPRU                 C
C      COMMON:              EPS,ICA,KEMAX,LSC,MVC,NRC,XC,YC,ZC,IN     C
C      SUBROUTINES:         INFLAS [INFLAP, INFLA4, INFLA3]           C
C                                                                     C
C*********************************************************************C
      SUBROUTINE SEARCH[NIRS,IRS,NICA,NLSC,NMVC,NNRC,A,B,C,ISPRU]
C**********************************************************************
      DIMENSION XT[10,10,10],YT[10,10,10],ZT[10,10,10],INDLI[1024]
      DIMENSION XC[1000],YC[1000],ZC[1000]
      DIMENSION ICA[25],LSC[25],MVC[25],NRC[25],IHQ[200],LIHIWI[11,11,2]
      DIMENSION IRS[1]
C**********************************************************************
      COMMON IN,IM,IKL
      COMMON XMIN,YMIN,IHQ,INDLI,LIHIWI,KEMAX,XT,YT,ZT,EPS,DIVIS,III,KZZ
      COMMON ICA,LSC,MVC,NRC
C**********************************************************************
      EQUIVALENCE [XT,XC],[YT,YC],[ZT,ZC]
C**********************************************************************
C*****SET VARIABLES TO ZERO, THAT INDICATE NUMBER OF TRIANGLES
C      IN WHICH THE POINT TO BE TESTED FOR VISIBILITY IS SITUATED:
C
    1 NICA=0
    2 NLSC=0
    3 NMVC=0
    4 NNRC=0
    5 ISPRU=1
```

```
C*****DELIVER X-, Y-COORDINATES OF THE TEST POINT:
C
    6  CALL INFLAP[A,B]
C
C*****4 CORNERS OF A PATCH WITH THE INDICES IRS[MP], IRS[MP]+1,
C      IRS[MP]+KEMAX+1 AND IRS[MP]+KEMAX:
C
    7  DO 39 MP=1,NIRS
C
C*****COMPUTE INDICES:
C
    8  IR=IRS[MP]
    9  IR1=IR+1
   10  IR2=IR+KEMAX
   11  IR3=IR2+1
C
C*****INTERROGATION, WHETHER ALL Z-COORDINATES OF THE QUADRANGLE ARE
C      SMALLER THAN THE Z-COORDINATE OF THE TEST POINT. IF THAT IS THE
C      CASE, THE PATCH IS IRRELEVANT FOR FURTHER TESTING:
C
   12  IF[ZC[IR]-C+EPS] 13,13,16
   13  IF[ZC[IR1]-C+EPS] 14,14,16
   14  IF[ZC[IR3]-C+EPS] 15,15,16
   15  IF[ZC[IR2]-C+EPS] 39,39,16
C
C
C*****INTERROGATION, WHETHER THE TEST POINT P[A,B,C] LIES IN THE PATCH:
C
   16  CALL INFLA4[IR,IR1,IR3,IR2]
C
C*****EXCLUDE, THAT P LIES IN A PATCH CORNER:
C
C
   17  IF[IN] 18,39,19
   18  IF[IN+1] 19,39,39
C
C*****INTERROGATION, WHETHER ALL Z-COORDINATES OF THE QUADRANGLE ARE
C      LARGER THAN THE Z-COORDINATE OF THE TEST POINT. IF THAT IS THE
C      CASE, P IS INVISIBLE AND VIA VISKRI IT IS BRUNCHED TO GREVIS.
C
C
   19  IF[C-ZC[IR]+EPS] 20,25,25
   20  IF[C-ZC[IR1]+EPS] 21,25,25
   21  IF[C-ZC[IR2]+EPS] 22,25,25
   22  IF[C-ZC[IR3]+EPS] 23,25,25
C
   23  ISPRU=0
C
C*****RETURN TO VISKRI
C
   24  RETURN
C
C
C
C
C
C
C
C
C*********************************************************************
```

```
C*****DIVISION OF THE PATCH INTO 2 TRIANGLES WITH THE INDICES
C        LSC[MP], LSC[MP]+1, LSC[MP]+KEMAX+1 AND ICA[MP], ICA[MP]+KEMAX,
C        ICA[MP]+KEMAX+1.
C
C*****INTERROGATION, WHETHER POINT P LIES IN ONE OF THE 2 TRIANGLES:
C
   25   CALL INFLA3[IR,IR1,IR3]
C
C*****EXCLUDE THAT THE POINT LIES AT THE EDGE OF A TRIANGLE:
C
   26   IF[IN] 32,27,30
C
C*****FILL ARRAYS:
C
   27   NICA=NICA+1
   28   ICA[NICA]=IR
C
   29   GOTO 32
C
   30   NLSC=NLSC+1
   31   LSC[NLSC]=IR
C
C
C
C***********************************************************************
C
C*****DIVISION OF THE PATCH INTO 2 TRIANGLES WITH THE INDICES
C        NRC[MP], NRC[MP]+1, NRC[MP]+KEMAX AND MVC[MP]+1, MVC[MP]+KEMAX,
C        MVC[MP]+KEMAX+1.
C
C*****INTERROGATION, WHETHER POINT P LIES IN ONE OF THE 2 TRIANGLES:
C
   32   CALL INFLA3[IR1,IR2,IR3]
C
C*****EXCLUDE THAT THE POINT LIES AT THE EDGE OF A TRIANGLE:
C
   33   IF[IN] 39,37,34
C
C*****FILL ARRAYS:
C
   34   NMVC=NMVC+1
   35   MVC[NMVC]=IR
C
   36   GOTO 39
C
   37   NNRC=NNRC+1
   38   NRC[NNRC]=IR
C
   39   CONTINUE
C
C
C
C*****RETURN TO VISKRI
C
   40   RETURN
C
C***********************************************************************
        END
```

COMMON ALLOCATION

77777 IN	77776 IM	77775 IKL	77773 XMIN
77771 YMIN	77461 IHG	75461 INDLI	75077 LIHIWI
75076 KEMAX	71156 XT	65236 YT	61316 ZT
61314 EPS	61312 DIVIS	61311 III	61310 KZZ
61257 ICA	61226 LSC	61175 MVC	61144 NRC
71156 XC	65236 YC	61316 ZC	

PROGRAM ALLOCATION

DUMMY IRS	DUMMY NICA	DUMMY NLSC	DUMMY NMVC
DUMMY NNRC	DUMMY ISPRU	00033 MP	DUMMY NIRS
00034 IR	00035 IR1	00036 IR2	00037 IR3
00040 SEARCH	DUMMY A	DUMMY B	DUMMY C

SUBPROGRAMS REQUIRED

INFLAP INFLA4 INFLA3

THE END

```
                                   PAGE
                              ***************************************************************
                              *
                              *
                              *    I N F L A S
                              *
                              *
                              ***************************************************************
                              *
                              *    A. SEYFERTH
                              *
                              ***************************************************************
                              *
                              *    INFLIN (XF,YF,IN)
                              *
                              *    INDICATIØN ØF THE FILE-BEGINNING ADDRESS AND THE RESULT CELL
                              *
                              ***************************************************************
                              *
                              *
00000                                RØRG     0
                              *
        00000074              EØIND  EQU      074
        00000563              EØADR  EQU      371
                              *
                              *      DEFINITIØN ØF THE PRØGRAMMED ØPERATØRS:
                              *
                              LDP    ØPD      012500000
                              FLS    ØPD      013500000
                              FLM    ØPD      014100000
```

```
                                  PAGE
                           *******************************************************************
                           *
00000   0 00 00000         INFLIN PZE                   ENTRY
                           *
00001   0 71 0 00563              LDX     E0ADR
00002   2 76 0 00000              LDA     0,2           BEGINNING-ADDRESS 0F THE X-FILE
                           *                            + INDEX-BIT
00003   0 55 0 00335              ADD     =016777776     -2, WITH FADE-0UT 0F THE FL0ATING-INDICAT0R
00004   0 35 0 00013              STA     XFAD
00005   2 76 0 00001              LDA     1,2
00006   0 55 0 00335              ADD     =016777776
00007   0 35 0 00014              STA     YFAD
00010   2 76 0 00002              LDA     2,2
00011   0 35 0 00015              STA     INAD
00012   0 51 0 00000              BRR     INFLIN        RETURN-BRANCH
                           *
00013   0 00 00000         XFAD   PZE                   BEGINNING-ADDRESS 0F THE X-FILE - 2
00014   0 00 00000         YFAD   PZE                   BEGINNING-ADDRESS 0F THE Y-FILE - 2
00015   0 00 00000         INAD   PZE                   ADDRESS 0F THE RESULT-CELL
                           *
                           *******************************************************************
```

```
                             PAGE
                      ******************************************************************
                      *
                      *
                      *      INFLAP(XP,YP)
                      *
                      *
                      *      LOAD AND STORE THE POINT COORDINATES
                      *
                      *
                      ******************************************************************
                      *
00016  0 00 00000     INFLAP PZE                  ENTRY
00017  0 71 0 00563          LDX      EOADR       ADDRESS OF THE 1ST PARAMETERADDRESS
00020  2 75 0 00000          LDB      0,2         1ST PARAMETERADDRESS
00021  0 46 00060            XXB                  TO X-REGISTER
00022  2 76 0 00000          LDA      0,2         1ST WORD OF XP
00023  0 35 0 00035          STA      XP
00024  2 76 0 00001          LDA      1,2         2ND WORD OF XP
00025  0 35 0 00036          STA      XP+1
00026  0 46 00020            CBX                  ADDRESS OF THE 1ST PARAMETERADDRESS TO XR
00027  2 71 0 00001          LDX      1,2         2ND PARAMETERADDRESS
00030  2 76 0 00000          LDA      0,2         1ST WORD OF YP
00031  0 35 0 00037          STA      YP
00032  2 76 0 00001          LDA      1,2         2ND WORD OF YP
00033  0 35 0 00040          STA      YP+1
00034  0 51 0 00016          BRR      INFLAP      RETURN-BRANCH
                      *
00035                 XP     RES      2           XP-BUFFER
00037                 YP     RES      2           YP-BUFFER
                      *
                      ******************************************************************
```

```
                              PAGE
                        ***************************************************************
                        *
                        *
                        *     INFLA3(IND1,IND2,IND3)
                        *
                        *
                        *     INFLA WITH 3 INDICES
                        *
                        *
                        ***************************************************************
                        *
00041  0 00 00000       INFLA3 PZE                    ENTRY
                        *
00042  0 76 0 00336            LDA    =1
00043  0 01 0 00050            BRU    INFLA         BRANCH TO INFLA
                        *
                        ***************************************************************
```

```
                              PAGE
                      **************************************************************
                      *
                      *
                      *       INFLA4(IND1,IND2,IND3,IND4)
                      *
                      *
                      **************************************************************
                      *
00044  0 00 00000     INFLA4  PZE                      ENTRY
                      *
00045  0 76 0 00044            LDA      INFLA4
00046  0 35 0 00041            STA      INFLA3
00047  0 76 0 00337            LDA      =2
00050  0 35 0 00273   INFLA    STA      PANZ            NUMBER OF PARAMTERS - 2
00051  0 55 0 00336            ADD      =1
00052  0 55 0 00074            ADD      EOIND
00053  0 35 0 00272            STA      Z
00054  0 71 1 00272            LDX      *Z              INDEX OF THE LAST POINT
00055  0 43 0 00274            BRM      XVER            RESULT = X - XP
00056  0 35 0 00264            STA      XL+1
00057  0 35 0 00254            STA      X1+1
00060  0 36 0 00263            STB      XL
00061  0 36 0 00253            STB      X1
00062  0 43 0 00310            BRM      YVER            GET Y
00063  0 35 0 00266            STA      YL+1            RESULT = Y - YP
00064  0 35 0 00260            STA      Y1+1
00065  0 36 0 00257            STB      Y1
00066  0 36 0 00265            STB      YL
00067  0 46 00001              CLA
00070  0 35 0 00271            STA      ZAEHL           STANDARTISIZE COUNTER
00071  0 76 0 00074            LDA      EOIND           1ST PARAMETERADDRESS
00072  0 35 0 00272            STA      Z
```

```
                                 PAGE
                        *****************************************************************
                        *
                        *        IN THE FOLLOWING LOOP IT IS TESTED, WHETHER THE CONNECTING LINE
                        *        BETWEEN TWO SUCCEEDING POINTS, CROSSES THE POSITIVE  UNLIMITED
                        *        LINE Y=YP.
                        *        THEREFORE THE PAIRED VALUES X1,Y1 AND X2,Y2 ARE USED, EACH
                        *        OF WHICH CONTAINS THE DIFFERENCE COORDINATES P1-P RESP. P2-P .
                        *        P(XP,YP) IS THE POINT TO BE TESTED, P1 AND P2 ARE THE TWO
                        *        SUCCEEDING POINTS OF THE FILE.
                        *****************************************************************
                        *
00073  0 71 1 00272     SCHLA  LDX     *Z          GET THE NEXT POINT-INDEX
00074  0 43 0 00274            BRM     XVER        COMPUTE X-XP
00075  0 36 0 00255            STB     X2
00076  0 35 0 00256            STA     X2+1
00077  0 64 0 00254            MUL     X1+1        MULTIPLICATION OF X1 AND X2 TO ASCERTAIN THE
                        *                          ACCORDANCE IN SIGNS
00100  0 72 0 00340            SKA     =040000000  SKIP AT POSITIVE MULTIPLICATION RESULT
00101  0 01 0 00147            BRU     XVV         NEGATIVE MULTIPLICATION RESULT AND X1 AND X2
                        *                          DIFFERENT SIGNS (CASE 6)
                        *
                        *                          X1 AND X2 HAVE EQUAL SIGNS OR ARE 0
00102  0 52 0 00341            SKB     =077777776    SKIP, IF THE MULTIPLICATION RESULT = 0
00103  0 01 0 00135            BRU     XGV         MULTIPLICATION RESULT IS UNEQUAL 0
                        *
                        *                          AT LEAST ONE X-VALUE = 0
00104  0 55 0 00254            ADD     X1+1
00105  0 55 0 00256            ADD     X2+1
00106  0 50 0 00342            SKE     =0          SKIP, IF X1 AND X2 =0
00107  0 01 0 00117            BRU     X1U         BRANCH TO X1U
                        *                          BOTH X-VALUES ARE =0, I.E. THEY LIE ON THE
                        *                          VERTICAL X = XP (CASE 1)
```

```
00110  0 43 0 00310          BRM   YVER       GET Y2=YP-Y
00111  0 36 0 00257          STB   Y1         EXP. OF Y1 IS NO LONGER NEEDED
                       *                      AND THEREFORE OVERWRITTEN AT ONCE
00112  0 75 0 00340          LDB   =040000000   MASK FOR SIGN BIT
00113  0 70 0 00260          SKM   Y1+1       SKIP,IF Y1 AND Y2 HAVE EGUAL SIGNS
00114  0 01 0 00240          BRU   INLIN      POINT ON THE PERPENDICULAR LINE, BECAUSE
                       *                      OF UNEQUAL SIGNS ( CASE 1B )
00115  0 35 0 00260          STA   Y1+1       POINT NOT ON THE LINE,NO INTERSECTION-POINT
                       *                      POSSIBLE (CASE 1A)
00116  0 01 0 00221          BRU   SCHLEN     BRANCH TO END OF LOOP
                       *
                       *                      ONE X-VALUE =0, ONE X-VALUE UNEQUAL 0
00117  0 73 0 00342      X1U SKG   =0         SKIP, IF > 0
00120  0 01 0 00137          BRU   NUM        ONE X = 0, ONE NEG., NO INTERSECTIONPOINT WITH
                       *                      THE POS. UNLIMITED LINE POSSIBLE (CASE 2)
00121  0 76 0 00254          LDA   X1+1       ONE X =0, THE OTHER POSITIVE, INTERSECTION-
                       *                      POINT IS GIVEN AT UNEQUAL Y-SIGN (CASE 3)
                       ***********************************************************************
                       *
                       *         NO X-VALUE < 0
                       *         AT LEAST ONE X-VALUE > 0
                       *         CASE 3 AND 4
                       *
                       ***********************************************************************
00122  0 43 0 00310     XPOS BRM   YVER       GET Y2=Y-YP
00123  0 36 0 00257          STB   Y1
00124  0 75 0 00340          LDB   =040000000   SIGN MASK
00125  0 70 0 00260          SKM   Y1+1       SKIP, IF SIGNS OF Y1 AND Y2 ARE EGUAL.
                       *                      THEN: (CASE 3A AND 4A)
00126  0 61 0 00271          MIN   ZAEHL      Y-SIGNS DIFFERNT, INTERSECTION POINT
                       *                      INCREASE COUNTER (CASE 3B AND 4B)
```

```
                                      PAGE
                                ***************************************************************
                                *
                                *     EXCHANGE AT THE FOLLOWING POSITION:
                                *
                                *     <A>=MANTISSA OF Y2
                                *     <Y1>=EXPONENT OF Y2
                                *     <X2,X2+1>=EXPONENT AND MANTISSA OF X2
                                *
                                *     X2 AND Y2 SHALL BECOME X1 AND Y1; THE NEXT POINT SHALL BECOME
                                *     X2 AND Y2.
                                *
                                ***************************************************************
                                *
00127  0 35 0 00260             UMS   STA   Y1+1      MANTISSA OF Y2 --> Y1+1
00130  0 76 0 00255                   LDA   X2        EXPONENT X2 --> X1
00131  0 35 0 00253                   STA   X1
00132  0 76 0 00256                   LDA   X2+1      MANTISSA OF X2 --> X1+1
00133  0 35 0 00254                   STA   X1+1
00134  0 01 0 00221                   BRU   SCHLEN    BRANCH TO THE END OF LOOP
                                *                     EXCHANGE END
00135  0 53 0 00254             XGV   SKN   X1+1      EQUAL X-SIGNS; SKIP IF X1 AND X2 NEGATIVE
00136  0 01 0 00122                   BRU   XPOS      BOTH POSITIVE  (CASE 4)
00137  0 76 0 00255             NUM   LDA   X2        NEGATIVE X-VALUES, EXCHANGE X (CASE 5)
00140  0 35 0 00253                   STA   X1        AND COMPUTE Y
00141  0 76 0 00256                   LDA   X2+1      X2 --> X1
00142  0 35 0 00254                   STA   X1+1
00143  0 43 0 00310                   BRM   YVER      Y2 HAS NOT YET BEEN FETCHED
00144  0 35 0 00260                   STA   Y1+1      BRING AND STORE TO Y1 AT ONCE
00145  0 36 0 00257                   STB   Y1
00146  0 01 0 00221                   BRU   SCHLEN    BRANCH TO THE END OF LOOP
```

```
                              PAGE
                       ************************************************************
                       *      DIFFERENT X-SIGNS, AT DIFFERENT Y-SIGNS INTERSECTION-POINT
                       *      POSSIBLE (CASE 6)
                       *
00147  0 43 0 00310    XVV    BRM     YVER        Y2=Y-YP
00150  0 35 0 00262           STA     Y2+1        STORE MANTISSA
00151  0 46 00020             CBX                 SAVE EXPONENT INTO X-REGISTER
00152  0 64 0 00260           MUL     Y1+1        MULTIPLICATION OF MANTISSAS OF Y1 AND Y2
00153  0 72 0 00340           SKA     =040000000   SKIP, IF RESULT OF MULTIPL. IS POSITIVE
00154  0 01 0 00174           BRU     YVV         Y-SIGN DIFFERENT, UNEQUAL 0 (CASE 6B)
00155  0 52 0 00341           SKB     =077777776   SKIP,IF MULTIPL.-RESULT = 0 (CASE 6A)
00156  0 01 0 00166           BRU     XUM         BOTH Y EQUAL SIGNS, UNEQU. 0,NO INTERSEC.-POINT
00157  0 55 0 00260           ADD     Y1+1        ONE Y-VALUE=0
00160  0 55 0 00262           ADD     Y2+1
00161  0 50 0 00342           SKE     =0          SKIP, IF Y1 AND Y2 =0
00162  0 01 0 00164           BRU     $+2         ONE Y-VALUE=0, ONE UNEQUAL 0.
00163  0 01 0 00240           BRU     INLIN       BOTH Y=0, DIFFERENT X-SIGNS (CASE 6C)
00164  0 73 0 00342           SKG     =0          SKIP,IF Y-VALUE UNEQU. 0 AND POS.; THEN CASE 6A
00165  0 01 0 00174           BRU     YVV         ONE Y=0, ONE NEGATIVE (CASE 6B)
                       *      EXCHANGE AT THE FOLLOWING POSITION:
                       *
                       *      <XR>=EXPONENT OF Y2
                       *      <Y2+1>=MANTISSA OF Y2
                       *      <X2,X2+1>=EXPONENT AND MANTISSA OF X2
                       *
                       ************************************************************
00166  0 53 0 00273    XUM    SKN     PANZ        ALL POINTS ALREADY WORKED OFF
00167  0 01 0 00171           BRU     $+2         NO,EXCHANGING
00170  0 01 0 00223           BRU     ENDE        YES, BRANCH TO REVALUATION AND RETURN
00171  0 37 0 00257           STX     Y1          EXPONENT OF Y2 --> Y1
00172  0 76 0 00262           LDA     Y2+1        <A>:= MANTISSA OF Y2
00173  0 01 0 00127           BRU     UMS         BRANCH TO FURTHER EXCHANGING
```

```
                                   PAGE
                          ***********************************************************************
                          *                                   X-VALUES AND Y-VALUES HAVE DIFFERENT SIGNS
                          *                                   LINEAR EQUATION HAS TO BE COMPUTED
                          *
00174  0 76 0 00260       YVV     LDA    Y1+1       MANTISSA OF Y1
00175  0 75 0 00257               LDB    Y1         EXPONENT OF Y1
00176  1 41 0 00255               FLM    X2         Y1*X2
00177  0 35 0 00270               STA    P1+1       STORE RESULT
00200  0 36 0 00267               STB    P1
00201  0 76 0 00262               LDA    Y2+1       MANTISSA OF Y2
00202  0 46 00040                 CXB               EXPONENT OF Y2
00203  1 41 0 00253               FLM    X1         Y2*X1
00204  1 35 0 00267               FLS    P1         Y2*X1-Y1*X2
00205  0 50 0 00342               SKE    =0         SKIP, IF RESULT OF SUBTRACTION =0
00206  0 01 0 00210               BRU    $+2        UNEQUAL 0
00207  0 01 0 00240               BRU    INLIN      NUMERATOR=0, POINT ON THE STRAIGHT LINE
00210  0 35 0 00267               STA    P1         SAVE MANTISSA FOR SIGN COMPARISON  WITH THE
                          *                         DENOMINATOR
00211  0 46 00040                 CXB               EXPONENT OF Y2
00212  0 76 0 00262               LDA    Y2+1       MANTISSA OF Y2
00213  1 35 0 00257               FLS    Y1         DENOMINATOR: Y2-Y1
00214  0 75 0 00340               LDB    =040000000
00215  0 70 0 00267               SKM    P1         SKIP, IF SIGN OF NUMERATOR AND DENOMINATOR
                          *                         IS EQUAL, I.E. RESULT X IS POSITIVE
00216  0 01 0 00166               BRU    XUM        UNEQUAL SIGN, NEGATIVE RESULT ,NO INTERSECTION-
                          *                         POINT, BRUNCH TO EXCHANGING
00217  0 61 0 00271               MIN    ZAEHL      POSITIVE RESULT, INTERSECTION-POINT
00220  0 01 0 00166               BRU    XUM        BRANCH TO EXCHANGING
00221  0 53 0 00273       SCHLEN  SKN    PANZ       END OF LOOP; SKIP, IF ALL POINTS
                          *                         ARE WORKED OFF
00222  0 01 0 00227               BRU    SCHLE      NOT ALL WORKED OFF
```

```
                                  PAGE
                           ******************************************************************
                           *
00223  0 76 0 00271        ENDE   LDA    ZAEHL     COUNTER RESULT OF INTERSECTION POINTS
00224  0 14 0 00336               ETR    =1        EVEN NUMBER: RESULT =0;
                           *                       POINT LIES OUTSIDE
00225  0 35 1 00015        RUECK  STA    *INAD     ODD NUMBER: RESULT=1
                           *                       POINT LIES INSIDE
00226  0 51 0 00041               BRR    INFLA3    RETURN
00227  0 60 0 00273        SCHLE  SKR    PANZ      PANZ:=PANZ-1, SKIP, IF PANZ < 0
00230  0 01 0 00234               BRU    SE        LOOP CONTINUATION
00231  0 76 0 00264               LDA    XL+1      ALL POINTS BUT THE LAST ARE WORKED OFF
00232  0 75 0 00263               LDB    XL        TAKE OVER THE LAST POINT ALREADY STORED
00233  0 01 0 00075               BRU    SCHLA+2
00234  0 61 0 00272        SE     MIN    Z         SET NEXT POINT-INDEX-ADDRESS
00235  0 01 0 00073               BRU    SCHLA     BRANCH TO LOOP START
00236  0 76 0 00343        INPU   LDA    =-1       POINT COINCIDES
00237  0 01 0 00225               BRU    RUECK
00240  0 60 0 00273        INLIN  SKR    PANZ      LINE MET, INTERROGATION FOR POINT-COINCIDENCE
00241  0 01 0 00244               BRU    $+3
00242  0 76 0 00341        INLIN1 LDA    =-2       LINE MET
00243  0 01 0 00225               BRU    RUECK
00244  0 61 0 00272               MIN    Z
00245  0 71 1 00272               LDX    *Z
00246  0 43 0 00274               BRM    XVER
00247  0 50 0 00342               SKE    =0
00250  0 01 0 00240               BRU    INLIN
00251  0 43 0 00310               BRM    YVER      AT POINT-COINCIDENCE DIRECT BRANCH TO INPU
00252  0 01 0 00240               BRU    INLIN
```

```
                                 PAGE
                           ***********************************************************************
                           *
                           *     FIELDS
                           *
00253                      X1    RES      2
00255                      X2    RES      2
00257                      Y1    RES      2
00261                      Y2    RES      2
00263                      XL    RES      2
00265                      YL    RES      2
00267                      P1    RES      2
00271  0 00 00000          ZAEHL PZE
00272  0 00 00000          Z     PZE
00273  0 00 00000          PANZ  PZE
                           *
```

```
                              PAGE
                        ****************************************************************
                        *
                        *     FIELDS
                        *
00253                   X1    RES     2
00255                   X2    RES     2
00257                   Y1    RES     2
00261                   Y2    RES     2
00263                   XL    RES     2
00265                   YL    RES     2
00267                   P1    RES     2
00271  0 00 00000       ZAEHL PZE
00272  0 00 00000       Z     PZE
00273  0 00 00000       PANZ  PZE
                        *
```

```
                           PAGE
                    ******************************************************************
                    *
                    *FETCH XVER AND YVER, X- RESP. Y-VALUE OUT OF FILE
                    *      COMPARISON WITH POINT XP,YP. IF THERE IS NO COINCIDENCE
                    *      XP,YP IS SUBTRACTED, THE RESULT IS IN A AND B.
                    *
                    ******************************************************************
                    *
00274  0 00 00000    XVER   PZE
00275  1 25 1 00013         LDP     *XFAD      NEXT X-VALUE
00276  0 50 0 00036         SKE     XP+1       IS MANTISSA = MANTISSA OF XP
00277  0 01 0 00306         BRU     XSUB+1     MANTISSAS UNEQUAL
00300  0 46 00014           XAB                YES, COMPARE EXPONENTS TOO
00301  0 50 0 00035         SKE     XP
00302  0 01 0 00305         BRU     XSUB       EXPONENTS UNEQUAL
00303  0 46 30003           CLR                EXPONENTS EQUAL, RESULT =0
00304  0 51 0 00274         BRR     XVER       RETURN
00305  0 46 00014    XSUB   XAB
00306  1 35 0 00035         FLS     XP         X - XP
00307  0 51 0 00274         BRR     XVER       RETURN
                    *
                    *                          AT THE ENTRY YVER  MUST HAVE THE MANTISSA
                    *                          OF THE APPERTAINING X-VALUE IN A-REGISTER
00310  0 00 00000    YVER   PZE
00311  0 53 0 00273         SKN     PANZ       SKIP, IF PANZ NEGATIVE, I.E.THE LAST VALUE
                    *                          ALREADY STORED SHALL BE TAKEN OVER
00312  0 01 0 00316         BRU     NLY
00313  0 76 0 00266         LDA     YL+1
00314  0 75 0 00265         LDB     YL
00315  0 51 0 00310         BRR     YVER       RETURN
00316  0 35 0 00334  NLY    STA     XM         STORE MANTISSA OF X-VALUE
00317  1 25 1 00014         LDP     *YFAD      GET Y-VALUE
```

```
00320  0 50 0 00040            SKE    YP+1     SKIP, IF CØINCIDENCE ØF MANTISSAS ØF Y AND YP
00321  0 01 0 00332            BRU    YSUB+1   NØ CØINCIDENCE
00322  0 46 00014               XAB             CØMPARISØN ØF EXPØNENTS
00323  0 50 0 00037            SKE    YP       SKIP, IF CØINCIDENCE ØF Y AND YP
00324  0 01 0 00331            BRU    YSUB     NØ CØINCIDENCE
00325  0 46 30003               CLR             RESULT: Y - YP
00326  0 50 0 00334            SKE    XM       IS MANTISSA ØF  THE APPERTAINING X-VALUE =0 TØØ
00327  0 51 0 00310            BRR    YVER     NØ, RETURN
00330  0 01 0 00236            BRU    INPU     PØINT CØINCIDENCE
00331  0 46 00014        YSUB   XAB
00332  1 35 0 00037            FLS    YP       Y - YP
00333  0 51 0 00310            BRR    YVER     RETURN
00334  0 00 00000         XM     PZE
                          ****************************************************************
                          *
                          *      GLØBAL NAMES
                          *
                          $INFLIN,INFLAP,INFLA3,INFLA4
                          *
                          ****************************************************************
                                 END

00335  16777776
00336  00000001
00337  00000002
00340  40000000
00341  77777776
00342  00000000
00343  77777777
```

7. References

/1/ I. Sutherland — SKETCHPAD - A man machine graphical communication system. Technical Report No. 296, MIT, Cambridge, Jan. 1963

/2/ J. Encarnaçao, W. Giloi — PRADIS - An advanced programming system for 3-D-displays. AFIPS Conference Proceedings, Vol. 40, 1972, pp. 985

/3/ J. Encarnaçao — A survey of and new solutions for hidden-line problem Proceedings of the CG-Symposium, Delft University, 1970

/4/ I. Sutherland, R.F. Sproull, R.A. Schumacher — Sorting and hidden-surface problem AFIPS Conference Proceedings Vol. 42, 1973, pp. 685

/5/ J. Encarnaçao — Untersuchungen zum Problem der rechnergesteuerten räumlichen Darstellungen auf ebenen Bildschirmen Dissertation an der Fakultät Elektrotechnik der Technischen Universität Berlin, Berlin 1971

/6/ R. Eckert — FLAVI - Untersuchung und Verallgemeinerung Studienarbeit am Institut für Informationsverarbeitung I, Technische Universität Berlin Betreuer: W. Giloi und J. Encarnaçao

/7/ R. Eckert

Implementierung eines weitgehend allgemeinen Visibilitätsverfahrens für Funktionen zweier Variablen
Diplomarbeit am Institut für Informationsverarbeitung I,
Technische Universität Berlin
Betreuer: W. Giloi und J. Encarnaçao

/8/ E. Kniepen

Verallgemeinerung und Optimierung des Visibilitätsverfahrens FLAVI, das auf mehrere Flächen, die außer durch Gleichungen auch durch Wertelisten gegeben sein können, angewendet werden soll.
Studienarbeit am Institut für Informationsverarbeitung I,
Technische Universität Berlin
Betreuer: W. Giloi und J. Encarnaçao

/9/ E. Kniepen

Interaktives Entwerfen und Darstellung von Raumflächen mit Unterdrückung der unsichtbaren Kanten und Aufbau eines Seitensystems für die zugehörige Datenstruktur.
Diplomarbeit am Institut für Informationsverarbeitung I,
Technische Universität Berlin,
Betreuer: P. Rechenberg u. J. Encarnaçao

/10/ P. Mahnkopf

FLAVIS - Ein allgemeines Programmsystem zur Untersuchung der Visibilität allgemeiner Flächen im Raum, die durch Gleichungen oder Wertelisten gegeben sind.
Institutsinterne Beschreibung
HHI - Abt. IV, Berlin 1973

/11/ R. Galimberti, U. Montanari — An algorithm for hidden-line elimination. Com. of ACM, Vol. 12, April 1969

/12/ A. Appel — The notion of quantitive invisibility and the machine rendering of solids. Proc. ACM, 1967

/13/ J.E. Warnock — A hidden surface algorithm for computer generated halftone pictures. Technical Report 4-15, June 1969 University of Utah, Salt Lake City

/14/ G.S. Watkins — A real-time visible surface algorithm. Computer Science Dep., University of Utah, UTECH-CSc-70-101, June 1970

/15/ R. Weiss — BE VISION; A package of IBM 7000 FORTRAN programs to draw orthographic views of combinations of plane and quadric surfaces. Journal of ACM, Vol. 13, No. 2, April 1966