Consiglio Nazionale delle Ricerche

IGDS: Interactive Design of Interpolated Surfaces

V. Casarosa - E. D'Ettorre - S. Trumpy

84

CNUCE

Divisione Servizio Elaborazione Dati

A cura di: V. Casarosa
E. D'Ettorre
S. Trumpy

Vittore Casarosa *

Enrica D'Ettorre +

Stefano Trumpy +

IGDS - INTERACTIVE DESIGN OF INTERPOLATED SURFACES

* IBM Scientific Center - PISA

+ CNUCE - Institute of CNR - PISA

IGDS - Interactive Design of Interpolated Surfaces

1. Introduction

In this report we describe IGDS (Interactive Graphic Display of Surfaces), a system for the interactive building and display of three-dimensional surfaces. The problem of representing surfaces of any shape is encountered in many different areas such as in car manufacturing, ship building, and aerospace and aircraft industries, in architecture or civil engineering, etc. Usually, for one reason or another, the shape of these surfaces must be "fair", a car's body must be "fair" mainly for aesthetic reasons, a ship's hull must be "fair" mainly for hydrodynamic reasons and so on. IGDS gives the designer the possibility of modelling a surface by manipulating interactively a set of easily understood input parameters. For this reason this type of "fair" surface is often referred to as "sculptured surfaces". The "sculptured surfaces" are usually treated as interpolated surfaces not having a simple analytical representation. The initial data are commonly constituted by a set of given points on which the surface has to lie; additional points of the surface are usually obtained by using interpolation methods.

As a consequence of the lack of an analytical representation, the representation of the surface is constituted by a large set of numbers, namely the spacial coordinates of the points lying on the surface. A graphic display of these points can accelerate the process that leads from the production of a first rough surface to a final one satisfying the required objectives.

IGDS has been designed to take advantage of the graphic capabilities offered by interactive devices such as the IBM 2250 display unit and the Tektronix 4013 storage scope. The API language has been used to implement IGDS, as it has proven to be the most convenient tool in the implementing of highly conversational systems both from the system programmer's point of view and the user's point of view.

In section II, there will be a general description of IGDS and the environment in which it can be used.

In section III, there is a detailed description of the mathematical model used to represent the surfaces.

In section IV, the commands necessary to work with IGDS are described in detail.

In section V, there is an explanation of how to use IGDS when an IBM 2250 display unit is available.

In section VI there is an explanation of how to use IGDS when a 4013 Tektronix storage tube is available.

In section VII, the implementation of IGDS is described.

In Appendix I, the principal CP commands are described.

In Appendix II, the principal CMS commands are described.

In Appendix III, the steps of the computation of the mathematical model are listed.

2. General description

IGDS is a software package implemented in APL and developed in a CP-CMS environment. In the following paragraphs a general description of IGDS and its environment will be given.

2.1 Access to the system

The IGDS system is constituted by two APL workspaces that are available in the public APL libraries of CNUCE. It can be used following the procedure described below.

Once connected with the computer, the user enters the command on the keyboard of the terminal

LOGIN <name>

<name> is the name of the user's virtual machine. After CP/67 has requested the password of the virtual machine and after the correct password has been given, the user is in the CP environment. Any legitimate CP command can be issued in this environment. One of the commands that can be used in this environment is the following:

```
MSG  <name>  <text>
```

This command enables the user to communicate with other virtual machines and in particular with that of the operator, named CP. When the 2250 is used, a message has to be sent to the operator to have the 2250 attached to the virtual machine.

IPL is another CP command which is used to start an operating system. With the command:

```
IPL  CMS
```

the CMS operating system is loaded into the virtual machine. In the CMS environment the user may enter any of the CMS commands and in particular the command APL.

With this command, the APL system is started and all the public libraries of APL are made accessible to the user.

In the APL environment all system commands must start with a right parenthesis. By using one of the commands:

```
)LOAD  5  IGDS2250
)LOAD  5  IGDS4013
```

depending on the output device available for the session, the IGDS functions are loaded in core. To start IGDS the following two functions must be entered:

```
INIT
START
```

The first function restores the IGDS environment to the situation it was in at the end of the last session. If this is the very first session, it initializes the system.

The second function activates the conversational environment and from this moment the user can issue the IGDS commands which are fully described in this report.

The sequence of commands that follows IPL CMS, namely:

```
 APL
 )LOAD IGDS <device type>
 INIT
 START
```

can be activated with the single CMS command

```
IGDS  <device type>
```

The <device type> must be 2250 or 4013 depending on the output device available for the session. Once entered in the CMS environment, by typing IGDS 2250 or IGDS 4013 it is possible to start the system.

2.2 APL implementation

IGDS consists of a set of APL functions contained in the two workspaces IGDS2250 and IGDS4013. The IGDS2250 workspace contains all the functions necessary in order to use IGDS with the IBM 2250 display unit and the plotter HP.
The IGDS4013 workspace contains all the functions necessary in order to use IGDS with the Tektronix 4013 storage scope and the plotter HP. Because of the additional software needed to support the 2250, the workspace IGDS2250 can be used only with a modified version of the APL system.
The standard program APLCMS running under CP/67 has been modified so as to include a graphic package for handling the 2250. The new system has been called APLFGS because the

graphic package used is the already existing Fortran Graphic Support (FGS) package. This package is a set of Assembler written routines that can be called by Fortran and has been developed at the IBM Scientific Center of Pisa. Though it is less powerful than GSP, the standard IBM package, FGS is easier to use and requires less buffer space in the 2250 control unit.

2.3 Output devices

The graphic output devices supported by IGDS are:

2250 IBM display unit

4013 TEXTRONIX storage tube

7201A HP mechanical plotter

The IBM 2250 is a refreshable scope equipped with an alphanumeric keyboard, a programmed function keyboard and a light pen. These facilities are used for doing the interactive work during a session; the alphanumeric keyboard is used to enter parameters, the programmed function keyboard is used to select actions corresponding to the lightened keys and the light pen is used to select commands to be executed.

When using the 2250, the terminal used to start IGDS may be disconnected since all interactions are handled through the 2250.

The Tektronix 4013 is a storage tube equipped with an alphanumeric keyboard and a crosshair cursor. This terminal may work either in alphanumeric mode or in graphic mode. When operating in alphanumeric mode, it behaves as a TTY-like terminal with the only difference that the writing appears on the screen. When using IGDS with the 4013, all the interactions are handled through the terminal which is switched alternatively from alphanumeric mode to graphic mode and viceversa when needed.

The 7201A Hewlett Packard plotter is constituted by a 30x40 drawing plane where the drawings are made on-line by a pen moving along two axes. This is an optional device that is used by IGDS as an hard-copy device for the 2250 or the 4013 terminal.

2.4 List of commands

We shall give here a short summary of the commands used when working with IGDS. A fully detailed description will be given in the sections following.

The command may be functionally divided into 4 groups:

Group 1: commands for the building and changing of a surface

BUILD - this command is used to create a new surface.

The user must give all the data necessary for defining the new surface and the number of points to be interpolated.

CHANGE - this command is issued to modify an existing surface or to change the number of the interpolated points.

RESTORE - this command substitutes the existing surface with a new one read from a disk file. This file must have been previously written by using the SAVE command.

Group 2: commands for displaying a surface

DISPLAY - this command enables the user to display on the output device a projection of the existing surface. It is possible to display either the initial surface, or the interpolated surface.

PLOT - this command draws on the mechanical plotter the last displayed image.

Group 3: commands for saving intermediate and final results.

SAVE - this command writes onto disk a binary file containing all the data defining the initial surface and the points of the interpolated surface.

WRITE - this command is the same as SAVE except that the file is written in character form and can be read by any other language.

STORE - this command creates, in memory or in disk files, copies of the last displayed image. It can be used to build an archive of pictures ready to be displayed.

Grcup 4: command for ending a session

END - with this command the user ends a session and exits from the IGDS environment. The present status of IGDS is automatically saved so that it can be reproduced at the beginning of another session.

3. Mathematical model

The major properties that make Coons model very suitable for representing the so-called "sculptured surfaces", are:

a) the continuity of the first and second derivatives at the points of the surface, thus obtaining the approximate shape of minimum strength energy;

b) singular points can not exist on the surface because the representation is parametric;

c) the surface is defined by a few input parameters which can be varied individually to produce easily predictable changes in the shape of the surface.

Coons method consists of a parametric bicubic representation in which a surface may be considered as lying on two families of intersecting curves, as is shown in fig. 1. The curves are defined by a set of initial points giving the two families of curves.

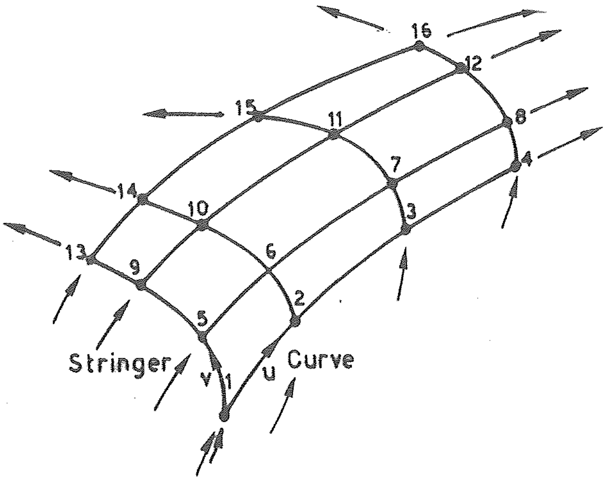


fig. 1

As the knowledge of a cubic parametric representation is essential in order to fully understand the behaviour of a surface, a description follows.

3.1 Cubic parametric curves

The problem to be resolved is that of fitting a spatial curve on a set of given points.
The problem of passing a cubic curve through two given points will first be examined. Four conditions must be specified in order to define a cubic curve; the condition that we impose on the curve is that it must pass through two given points with given tangents at those points. The defined cubic curve may be treated in parametric form.
Fig. 2 represents a curve where parameter t varies from 0 to the length L of the segment joining the two points.

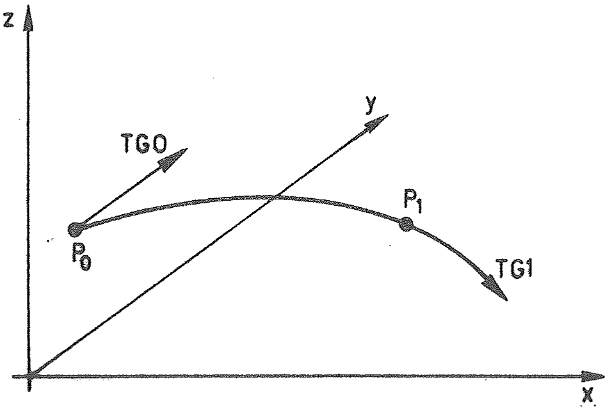


fig. 2

Expression 1) gives the equation of the point P(t) running on the curve.

$$1) \quad P(t) = P0*A0(t)+P1*A1(t)+TG0*B0(t)+TG1*B1(t)$$

Equation 1) may be split into the following three expressions:

$$
2) \quad
\begin{aligned}
X(t) &= X0 * A0(t) + X1 * A1(t) + TG0_x * B0(t) + TG1_x * B1(t)\\
Y(t) &= Y0 * A0(t) + Y1 * A1(t) + TG0_y * B0(t) + TG1_y * B1(t)\\
Z(t) &= Z0 * A0(t) + Z1 * A1(t) + TG0_z * B0(t) + TG1_z * B1(t)
\end{aligned}
$$

It appears from 2) that the coordinates of the generic point P(t) are a function of the coordinates of the two given points P0 and P1 and of the components of the two given tangents TG0 and TG1.

A0, A1, B0 and B1 are cubic polynomials in t satisfying the following conditions:

$$
3) \quad
\begin{aligned}
t = 0 \quad & \begin{cases} P(0) = P0 \\ P'(0) = TG0 \end{cases}\\
t = 1 \quad & \begin{cases} P(L) = P1 \\ P'(L) = TG1 \end{cases}
\end{aligned}
$$

It can be easily verified that the following four polynomials satisfy the imposed conditions.

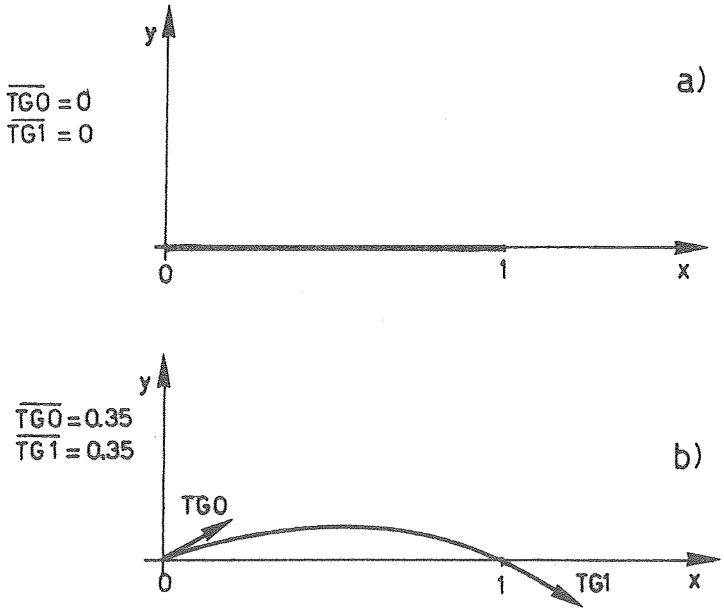
$$A0 = (2t^3 - 2Lt^2 + 1)/L^3$$

4) $$A1 = (-2t^3 + 3Lt^2)/L^3$$

$$B0 = (t^3 - 2Lt^2 + L^2 t)/L^2$$

$$B1 = (t^3 - Lt^2)/L^2$$

It can be seen from the equations 2) that both the direction and the moduli of the tangents at the given points influence the coordinates X(t), Y(t) and Z(t) of the running point and hence the shape of the curve. As an example let us consider a two dimensional curve with P0 = 0,0 and P1 = 1,0 and with TG0 has a slope of 45 degrees and TG1 has a slope of -45 degrees. Fig. 3 shows how the shape of the curve varies depending on the moduli imposed on the tangents.



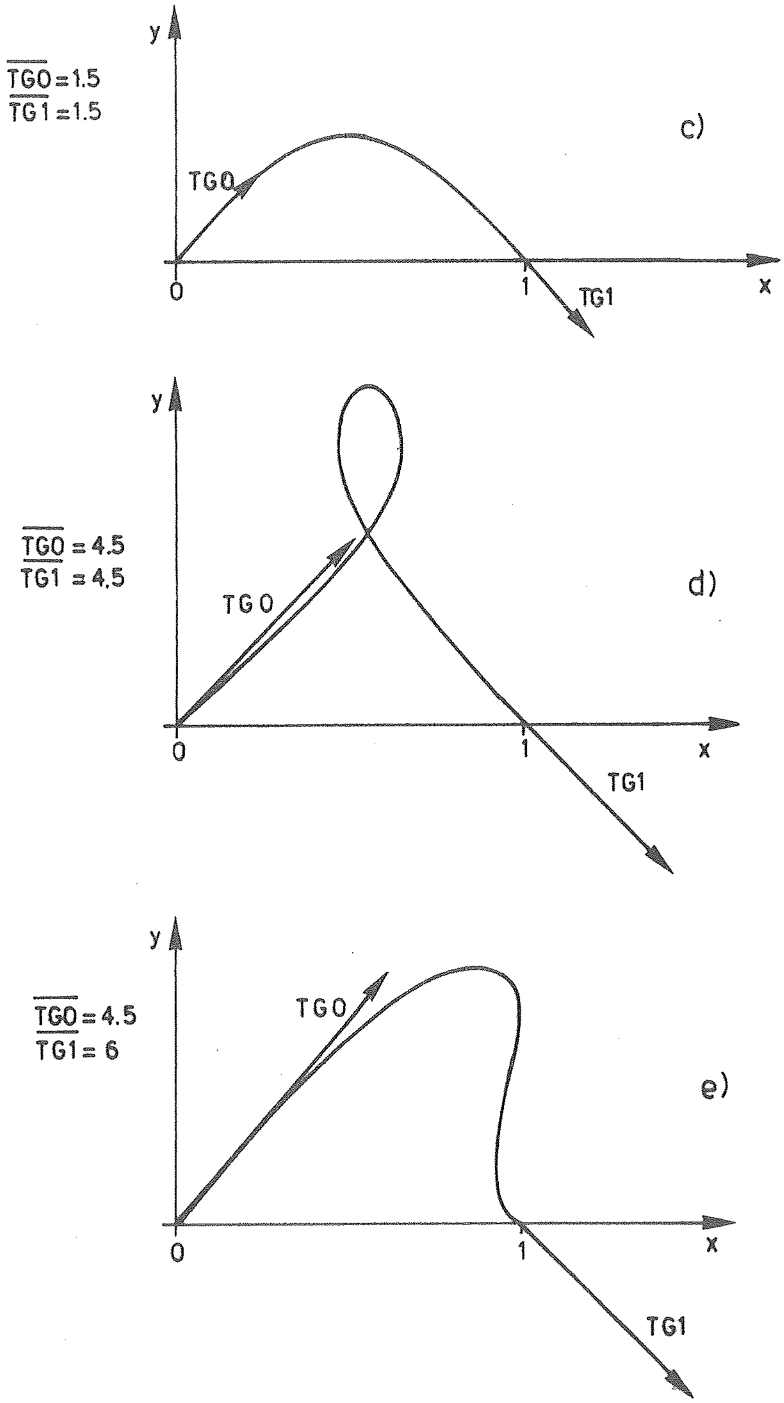

$\overline{TG0}=1.5$
$\overline{TG1}=1.5$
y
c)
TG0
0
1
TG1
x
y
$\overline{TG0}=4.5$
$\overline{TG1}=4.5$
d)
TG0
0
1
x
TG1
y
$\overline{TG0}=4.5$
$\overline{TG1}=6$
TG0
e)
0
1
x
TG1

fig. 3

The problem of fitting a curve, passing through a set of given points will now be considered. It is possible to connect a series of cubic arcs having the same characteristics as those previously described by imposing the same conditions on the connecting points. The imposed condition is that at each intermediate point the curve must have the continuity of the second derivative and hence also of the tangent.

In fig. 4 this condition has been visualized for two generic arcs l and I of a curve.

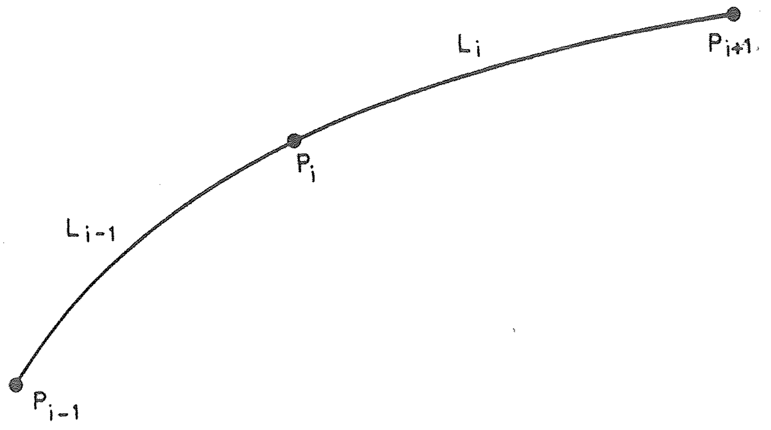


fig. 4

$$5)\quad P''_{i-1}(I_{i-1}) = P''_{i}(0)$$

By deriving twice 1) with respect to the parameter t, and imposing condition 5) we obtain the following equation: where Ki is an expression depending on P

$$6)\quad L_{i} * TG_{i-1} + 2*(L_{i-1} + L_{i}) * TG_{i} + L_{i-1} * TG_{i-1} = K_{i}$$

All the steps of the computation are listed in the Appendix 3.

It is then possible to write an equation like the 6) for each intermediate point of the curve where the TG , TG and TG are the unknowns and K are the known terms. With a set of N given points it is then possible to write N-2 equations containing N unknowns, namely the tangents at the N points. To specify completely the system, we must add 2 more equations. The two missing equations are determined by imposing the boundary conditions for the curve. The boundary conditions must be imposed on the first and last arc of the curve.

Possible conditions are for example:

a) zero curvature at the end points of the curve;

b) constant curvature in the ending arcs of the curve;

c) given tangents at the end points of the curve.

In our model the condition c) has been imposed. As a consequence, a curve is completely defined by the coordinates of all the points and the components of the tangents at the end points.

By writing N-2 equations like 6) and by adding the 2 boundary conditions, we obtain a system of N equations in N unknowns whose solution gives the tangents at all given points.

$$7)\quad \begin{vmatrix} TG_1 \\ TG2 \\ TG3 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ TG_{n-2} \\ TG_{n-1} \\ TG_n \end{vmatrix} \cdot \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_2 & 2\times(L_2+L_1 & L_1 & 0 & & & & \\ 0 & L_3 & 2\times(L_2+L_3) & L_2 & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ & & & & & & & \\ 0 & 0 & 0 & & L_{n-2} & 2\times L_{n-2}+L_{n-3} & L_{n-3} & 0 \\ 0 & 0 & 0 & & 0 & L_{n-1} & 2\times(L_{n-1}+L_{n-2}) & L_{n-2} \\ 0 & 0 & 0 & & 0 & 0 & 0 & 1 \end{vmatrix} = \begin{vmatrix} TG_1 \\ K_1 \\ K2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ K_{n-2} \\ K_{n-1} \\ TG'_n \end{vmatrix}$$

It is now possible to interpolate in each arc as many points as needed by applying the 2). Taking as a constant the number of points interpolated in each arc, in order to obtain a uniform distribution of the points along the whole curve, the following considerations must be taken into account.

1) the segments connecting the input points must not differ too much in length (ratios varying from .5 to 2 are acceptable; the optimum is 1)

2) the module of the tangents at the end points must be close to the length of the ending segments (optimal ratio is 0.7 - 0.8).

3) the distance from the cubic arc to the segment connecting two adjacent points increases with the increasing of tha angles formed by this segment with the two adjacent ones. Because the parameter t varies along the segment and not along the arc, it turns out that the points on the arc are closer to its ends the further the arc is from the segment.

3.2 Bicubic parametric surfaces

In Coons model, the representation of surfaces is parametric. Considering the plane defined by two parameters u and v, it is possible to establish a relation between the points of this plane and the points of the surface. In this way a point on the surface can be expressed as a function of u and v:

$$8)\quad P = P\ (u,v)$$

The basic surface unit in Coons method is called patch. A patch is a portion of surface delimited by four curves such that a relation like 8) has been established between the points of the surface and the unit square in the u,v plane. See fig. 5.

$$0 \leq u \leq 1 \qquad 0 \leq v \leq 1$$

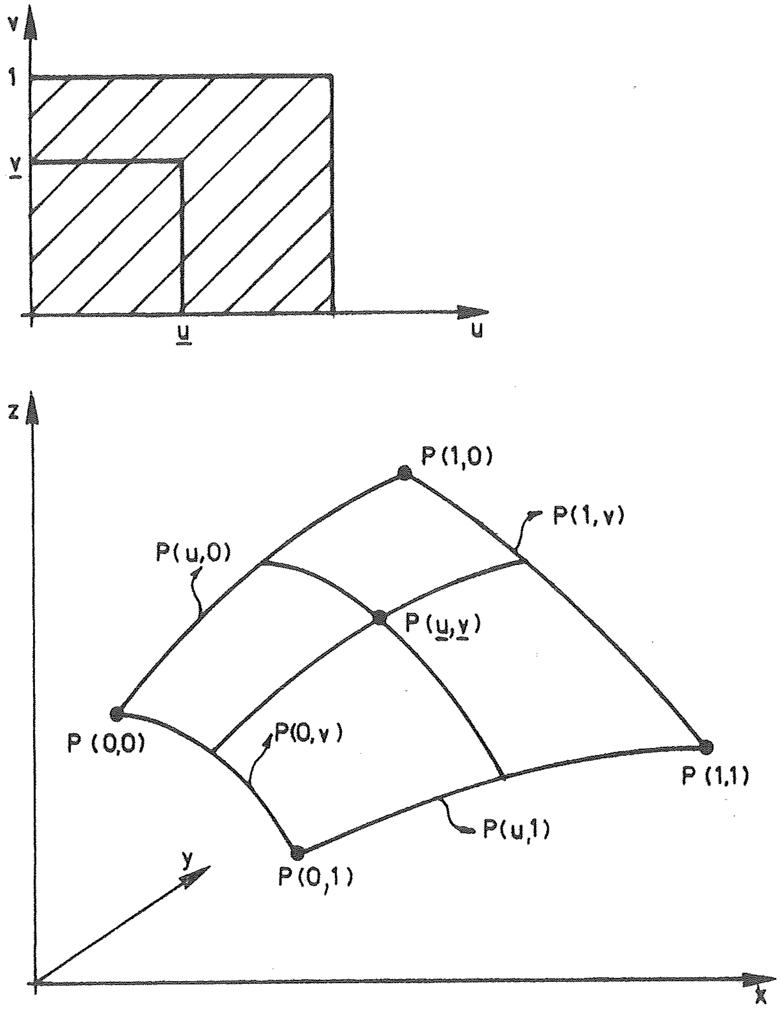


fig. 5

The boundary curves of the patch which correspond to the sides of the unit square, are cubic arcs like those described in the preceding paragraph.

The equation which allows to determine the coordinates of a point on the patch is expressed by the double matrix product:

$$9) \quad P(u,v) = \left| F0(u) \;\; F1(u) \;\; G0(u) \;\; G1(u) \right| \quad \left| M \right| \quad \left| \begin{array}{c} F0(v) \\ F1(v) \\ G0(v) \\ G1(v) \end{array} \right|$$

where M is a 4x4 matrix containing the geometrical data defining the four boundary curves of the patch.

$$10) \quad M = \left|\begin{array}{cc|cc} P(0,0) & P(0,1) & P_v(0,0) & P_v(0,1) \\ P(1,0) & P(1,1) & P_v(1,0) & P_v(1,1) \\ \hline P_u(0,0) & P_u(0,1) & 0 & 0 \\ P_u(1,0) & P_u(1,1) & 0 & 0 \end{array}\right|$$

The upper left quadruplet contains the coordinates of 4 corner points of the patch, namely P(0,0), P(1,0), P(1,0), P(1, 1). The upper right quadruplet contains the tangents at the corner points in the directions of the parameter v, namely Pv(0,0), Pv(0,1), Pv(1,0), Pv(1, 1). The lower left quadruplet contains the tangents at the corner points in the direction of the parameter u, namely Pu(0,0), Pu(0,1), Pu(1,0), Pu(1, 1). The lower right quadruplet is filled with zeros because in the present implementation we have not exploited the possibility given by Coons method of using the second mixed derivatives for introducing twists on the surface.

F0,F1,G0,G1 are cubic polynomial which quantify the influence of the boundary curves on the coordinates of a generic point P corresponding to the values u and v of the parameter u and v. These polynomials are called blending functions and we can see from 11) that they are identical to the 4) except for the length L which has been set equal to 1.

$$F0\ (t) = 2t^3 - 3t^2 + 1$$

11)
$$F1\ (t) = -2t^3 + 3t^2$$

$$G0\ (t) = t^3 - 2t^2 + t$$

$$G1\ (t) = t^3 - t^2$$

In practice, in crder tc obtain the coordinates of an interpolated point, the computation is made on the components of various quantities along the x, y and z axes, resulting in three expressions like 9). A uniform grid can be interpclated on the patch by incrementing the parameters u and v frcm 0 tc 1 with ccnstant steps.

A multi-patch surface is a surface constituted by several patches arranged in such a way that their boundary curves form a rectangular grid as shown in fig. 6.

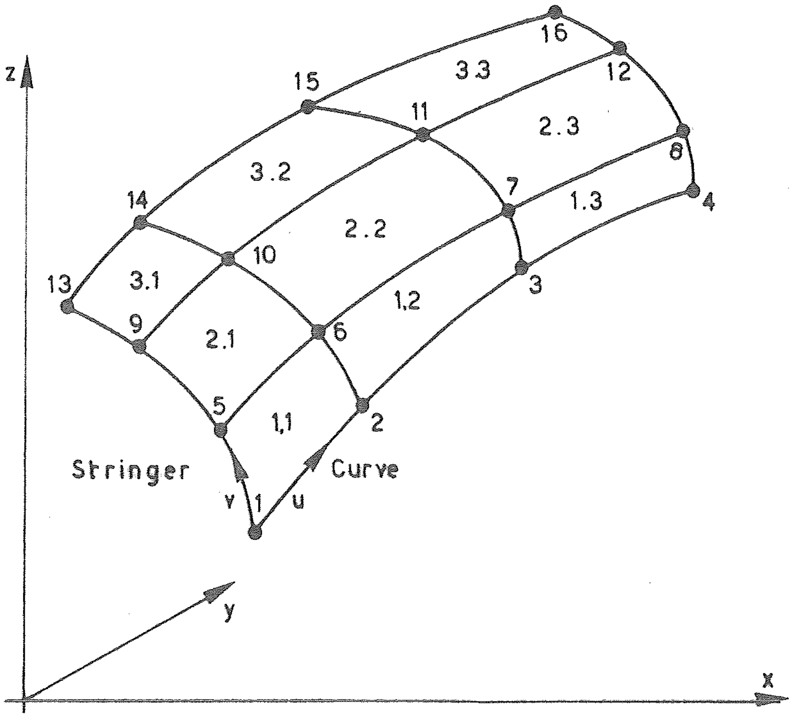


fig. 6

As can be seen from fig. 6 the input points defining the multipatch surface individuate on it two families of intersecting curves. The curves in one set are called CURVES and are characterized by the variation of the parameter u; the curves in the other set are called STRINGERS and correspond to the variation of the parameter v. To obtain the coordinates of a point lying in an arbitrary place on a multi-patch surface, 9) must be applied with M corresponding to the patch containing the point and with the appropriate values of u and v.

In order to determine the tangents at the corner points of an arbitrary patch, the CURVES and the STRINGERS are treated as cubic parametric curves of the type previously described. By applying 7) for each CURVE and for each STRINGER it is possible to compute all the tangents at the given points both in the direction of u and in direction of v. From 7) it comes out that in order to define completely a multi-patches surface, the tangents at the points of the boundary of the surface (that is the end points of the CURVES and of the STRINGERS) must be supplied.

The basic steps for the computation of the coordinate of a point lying on a multi-patches surface are then the following:

1) definition of the input points lying on the CURVES and on the STRINGERS and definition of the tangents at the boundary points;
2) evaluation of all internal tangents along the direction of u and v by applying 7) as many times as necessary;

3) evaluation of the coordinates of the points by applying 9) with B corresponding to the patch containing the point and with appropriate values of u and v.

In IGDS the third step is not executed singularly for each point but is rather computed by setting fixed increments of the parameters u and v which are assumed to be the same for all the patches. In such a way a new grid with a finer resolution than the initial one is obtained.

To conclude this chapter a list follows of the data required by IGDS for obtaining an interpolated surface.

a) the input points arranged in such a way as to form an initial rectangular grid of CURVES and STRINGERS;

b) the tangents along the CURVES at the end points of the CURVES and the tangents along the STRINGERS at the end points of the STRINGERS;

c) the dimension of the grid to be interpolated in each patch, namely the number of CURVES and of STRINGERS to be interpolated between the given ones. These dimensions are taken as a constant for all the patches of the surface.

4. IGDS COMMANDS

IGDS deals with three-dimensional surfaces and two-dimensional projections which are stored either temporarily in memory or permanently in disk files. At any moment there is present in memory only one surface, which is called the "current surface". The current surface can be defined by data coming either from the keyboard, and hence supplied by the user, or from a binary file on disk which must have been previously written in a suitable format.

All the relevant data on the current surface can be stored in a disk file either in binary form or in character form. The binary file is obviously written in the format necessary for re-reading in memory in order to redefine the current surface. The file in character form gives the possibility of printing off-line all the data of the surface and gives an easy way of communicating with programs written in other languages.

In order to be displayed on the output devices, the current surface must be projected in a two-dimensional image. The image last displayed is stored in memory and is called the "current image". Copies of the current image can be saved either in memory (at most two) or in a disk file. These copies may be used to redefine the current image, so that the user can easily re-display previous images.

Every time the information stored in memory is changed, IGDS writes automatically a disk file containing the relevant data. IGDS uses this file at the beginning of the next session in order to restore the situation existing at the end of the last session. In addition it is also used as a back-up copy in case of system failure.

We shall now give a detailed description of the commands of IGDS and of their parameters.

4.1 BUILD

This command is used in order to define the current surface with data coming from the keyboard and supplied by the user. IGDS asks for the following information:

1) the number of curves and the number of stringers; the user must give two integer positive numbers.
2) the coordinates of the points of the surface; the points are conceptually numbered first along the curves and then along the stringers. The coordinates of the points can be entered in any order as each entry must consist of the serial number of a point followed by its coordinates along x,y and z axes.
   Several points identified by progressive serial numbers can be defined with only one entry by entering the serial number of the first point followed by the triplets of the coordinates of the subsequent points.
   As the total number of points is known to IGDS, the coordinates of the points not explicitly defined are set to zero.
3) the components of the tangents at the border points of the surface; each entry must consist of the serial number of a border point followed by the x,y,z components of the tangent along the curves or along the stringers, as appropriate. Recalling the mathematical model, at the corner points both tangents must be defined. For that reason, if the serial number refers to a corner point, the components of the tangents must be

followed by a "1" or by a "0" to indicate whether the tangent is along the curves or along the stringers respectively. Any tangent not explicitly defined is assumed to be equal to zero.

4) the size of the interpolating grid; the user must give two positive integer numbers. The first number specifies how many curves will be interpolated between two adjacent initial curves. The second number specifies how many stringers will be interpolated between two adjacent initial stringers.

Error reports :

INCORRECT ENTRY

the format of the entry is not correct; the entry is ignored and IGDS waits for a new entry.

INCORRECT COMMAND OR PARAMETERS (IGDS4013)

the user has given a parameter on the same line of the command BUILD. The command is ignored and IGDS waits for a new command.

4.2 CHANGE

This command is used to modify the current surface. Depending on the option chosen it is possible to modify the interpolating grid, the components of the tangents and the coordinates of the input points. The input of the new data is made in the same way as that of the BUILD command.

a) CHANGE GRID

The user must give two positive integer numbers defining the new size of the interpolating grid. The first number specifies how many curves will be interpolated between two adjacent initial curves. The second number specifies how many stringers will be interpolated between two adjacent initial stringers.

b) CHANGE TANGENTS

The user must give the new components of the tangents at the border points of the surface. Each entry must consist of the serial number of a border point followed by the x, y and z components of the tangent along the curves or along the stringers, as appropriate. The components of a tangent at a corner point must be followed by a "1" or by a "0" to indicate whether the tangent is along the curves or along the stringers, respectively. The tangents not modified maintain their previous values. When the input of the new values is terminated, the system allows the user to change also the interpolating grid, if wanted, in the same way as the CHANGE GRID command.

c) CHANGE POINT

The user must give the new coordinates of the points. The new coordinates can be entered in any order as each entry must consist of the serial number of the point followed by its coordinates along x,y and z axes. The new coordinates of several points identified by progressive serial numbers can be defined by only one entry by entering the serial number of the first point followed by the triplets of the coordinates of the subsequent points. When the input of the new values is terminated, the system allows the user to change also the tangents and the interpolating grid, if wanted, in the same way as the CHANGE TANGENTS and the CHANGE GRID commands.

Error reports :

INCORRECT ENTRY

the format of the entry is not correct; the entry is ignored and IGDS waits for a new entry.

INPUT NOT FOUND

no current surface is defined. The BUILD command must be used before in order to define it. The command is ignored and IGDS waits for a new command.

INCORRECT COMMAND OR PARAMETERS (IGDS4013)

the option chosen is not a valid one. The command is ignored and IGDS waits for a new command.

4.3 RESTORE

This command is used in order to define the current surface with data coming from a CMS binary file written on the P-disk. Such binary files are usually created by using the SAVE command. IGDS asks only for the name of the file, as the filetype is assumed to be IGDSDATA. After the command has been executed, the number of curves, the number of stringers and the interpolating grid of the new current surface are displayed.

Error reports :

INCORRECT PARAMETER

the filename given by the user is not a valid CMS filename. The command is ignored and IGDS waits for a new command.

FILE NOT FOUND

the file with the given name and type IGDSDATA does not exist on the P-disk. The command is ignored and IGDS waits for a new command.

4.4 DISPLAY

This command must be used in order to display on the sreen of the 2250 or of the 4013 an image of the input points (namely only those given by the user with the BUILD command) or an image of the whole surface (namely the input points and the interpolated points). A brief description of the type of the image also appears on the screen. Every time the DISPLAY command is executed, a new current image is defined. Depending on the option chosen, the current image can be obtained from different sources.

a) DISPLAY INPUT

The current image is constituted by a projection of the input points. The input points at the border of the surface are displayed connected by segments and also the tangents at these points are displayed. The user must specify which projection is wanted, choosing it among the following types of projection and giving the appropriate parameters.

1) CONIC projection.

The user must give the coordinates of a point called the view point. A conic projection is made from the view point on a plane perpendicular to the straight line joining the view point with the center of the surface as shown in fig. 7. To avoid distorsion on the projected image the view point must not be too close to the surface.

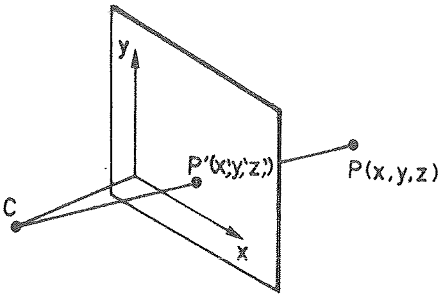


fig. 7

2) ISOMETRIC projection

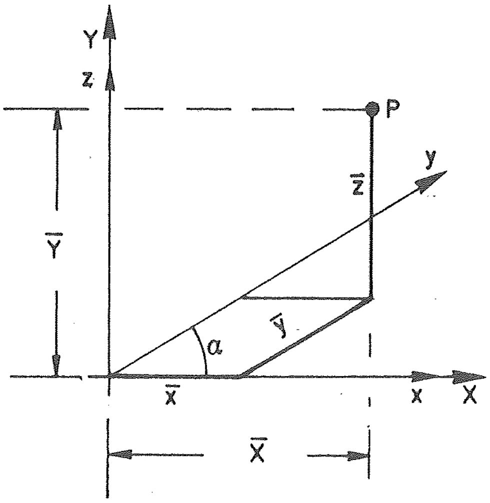


fig. 8

The x,y and z axes are projected on the plane as shown in fig. 8 under a cyclic permutation of the axes. The user must give the axis N to be dropped and the angle from which the surface has to be seen. The x,y and z axes are identified respectively by 1,2 and 3. The angle must vary between 0 and 90 degrees.

3) ASSONOMETRIC projection

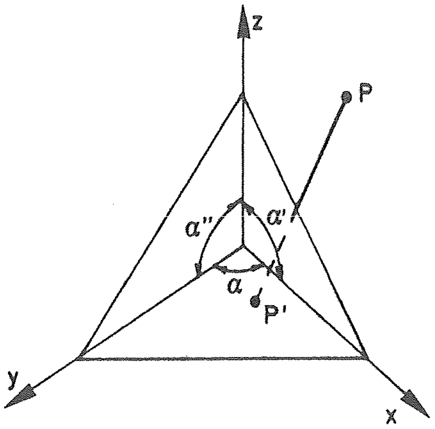


fig. 9

The x,y and z axes are projected on the plane as shown in fig. 9 The user must give the values of the angles whose sum must be 360 degrees. Each angle must be between 90 and 180 degreees. A good projection of the surface is obtained with

$\alpha = 110$ $\alpha' = 130$ $\alpha'' = 120$

4) ORTHOGONAL projection

The user must give the plane of projection with the characters XY, YZ or XZ.

b) DISPLAY SURFACE

The current image is constituted by a projection of the current surface. The input and the interpolated points are displayed connected by segments both along the direction of the curves and of the stringers. The user must specify which projection is wanted according to the specifications described for the display of the input points. For further possibilities when using the 2250 see section V.

c) DISPLAY FILE

The contents of the current image are read from a binary file on the P-disk, whose name must be given by the user and whose type must be IGDSIMAG. The file contains a projection either of the input points or of the interpolated points of a surface and must have been previously written by using the STORE command. The display of the image is made in the same way as described before according to whether the file contains a projection of the input points or a projection of the interpolated points. The only difference is that in this case the user has not to give the type of projection because the file must contain an already projected image.

d) DISPLAY FIRST

The contents of the current image are read from the first of the two copies of a previous image which may exist in memory. This copy must have been previously defined by the use of the STORE command. The same remarks apply as for the DISPLAY FILE option except for the source of the image. No further parameter is required from the user.

e) DISPLAY SECOND

This option behaves exactly as the DISPLAY FIRST option, except that the current image is read from the second of the two copies of a previous image which may exist in memory.

Error reports :

INCORRECT PARAMETER

The format of the input is not correct. The command is ignored. IGDS waits for a new command.

INPUT NOT FOUND

the user has requested a DISPLAY INPUT and no current input is defined. The command is ignored and IGDS waits for a new command.

SURFACE NOT FOUND

the user has requested a DISPLAY SURFACE and no current surface is defined. The command is ignored and IGDS waits for a new command.

IMAGE NOT FOUND

the user has requested a DISPLAY FIRST or a DISPLAY SECOND but the first or the second copy of an image has not been previously created. The command is ignored and IGDS waits for a new command

FILE NOT FOUND

the user has requested a DISPLAY FILE but the file specified does not exist. The command is ignored and IGDS waits for a new command.

4.5 PLOT

This command is used to plot the current image on the HP 72C1A mechanical plotter. No check is made to ensure that the plotter is actually attached to the user's virtual machine. The user must verify the availability of the plotter before issuing this command. The user is requested to specify the type of terminal which is being used.

a) PLOT SELECTRIC

this option specifies the user is working on a terminal with a SELECTRIC keyboard (the = sign is located at the left side of the keyboard).

b) PLOT EBCDIC

br this option specifies that the user is working on a terminal with an EBCIDIC keyboard (the = sign is located at the right side of the keyboard).

Error reports :

IMAGE NOT FOUND

no current image is defined. The command is ignored and IGDS waits for a new command.

INCORRECT COMMAND OR PARAMETERS

the format of the input is not correct. The command is ignored and IGDS waits for a new command.

4.6 SAVE

This command is used to make a copy of the current surface in a binary file on the P-disk. The contents of the current surface is written in a file whose name must be given by the user and whose type is set to IGDSDATA. If a file with the same name and type already exists it is erased before writing the new one. These files constitute a permanent archive of surfaces which can be used at any moment to redefine the current surface by means of the RESTORE command.

Error reports :

INCORRECT PARAMETER

the filename given by the user is not a valid CMS filename. The command is ignored and IGDS waits for a new command.

INPUT NOT FOUND

no current surface is defined. The command is ignored and IGDS waits for a new command.

4.7 WRITE

This command is used tc make a copy of the current surface in a character file cn the P-disk. The contents of the current surface is written in a file whose name must be given by the user and whose type is set to IGDSCHAR. If a file with the same name and type already exists it is erased before writing the new cne. These files constitute a permanent archive cf surfaces which can be easily read from prcgrams written in other languages cr printed off-line. For the exact format of the file, see section 7.

Error reports :

INCORRECT PARAMETER

The filename given by the user is not a valid CMS filename. The entry is ignored and IGDS waits for a new command.

INPUT NOT FOUND

no current surface is defined. The command is ignored and IGDS waits for a new command.

4.8 STORE

This command is used to make a copy of the current image either in a file on the P-disk or in memory depending on the option chosen. These copies may be displayed later by using the options of the DISPLAY command.

a) STORE FILE

The contents of the current image is written in a binary file on the P-disk whose name must be given by the user and whose type is set to IGDSIMAG. If a file with the same name and type already exists, it is erased before writing the new one.

b) STORE FIRST

At any moment there can be in memory at most two copies of a previous image referred to as first copy and second copy. This option is used to define in memory the first copy of the current image. No further parameter is required by the user.

c) STORE SECOND

This option is used to define in memory the second copy of the current image. No further parameter is required by the user.

Error reports:

INCORRECT PARAMETER

the filename given by the user is not a valid CMS filename. The entry is ignored and IGDS waits for a new command.

IMAGE NOT FOUND

no current image is defined. The command is ignored and IGDS waits for a new command.

4.9 END

This command is used to exit from the IGDS environment and to go in the APL environment or in the CMS environment depending on the option chosen. The user does not need to explicitly give the SAVE command or the STORE command because IGDS automatically saves the contents of the memory and restores it at the beginning of the new session.

a) END APL

By choosing this option, the APL environment is entered and the user can issue any legitimate APL statement. In particular to go from the APL environment to the CMS environment, the user must issue the)OFF CMS command. The workspace is not saved, but remains as the active workspace.

b) END CMS

By choosing this option, the CMS environment is entered. The user can issue any legitimate CMS command.

4.10 HELP FEATURE

By using the help feature, information can be obtained from the system whenever it is waiting for input. This feature is in particular of advantage for beginners and less experienced users.
As the use of the help feature varies depending on the display output device used, a complete description is deferred to sec. 5 and sec. 6, where an explanation of the use of IGDS with a 2250 or with a 4013 is given.

5. Using IGDS with a 2250

When using IGDS with a 2250, all the interactions between the system and the user pass through the 2250 by means of the programmed function keyboard (PFK), the alphanumeric keyboard and the light pen. The terminal is used only to start the IGDS and can then be disconnected. When IGDS is running, a menu of commands is always displayed on the screen. The menu of the available commands is the following:

```
END  APL
END  CMS
BUILD
CH   POINT
CH   TANG
CH   GRID
D    INPUT
D    SURFACE
D    FILE
D    FIRST
D    SECOND
ST   FILE
ST   FIRST
ST   SECOND
SAVE
RESTORE
PLOT
WRITE
HELP / RETURN
```

The options of the commands described in sec. IV are all included in the menu; therefore the command and the option choosen are selected simultaneously.

At any moment, IGDS can be in one of two environments: the "help environment" and the "execution environment". Pointing the light pen to a command in the menu gives different results depending on the environment. In the help environment, an explanation about the use of the command is displayed. In the execution environment the command is executed.

The help environment is entered automatically when IGDS starts and can be entered by the user at any moment by pointing the light pen to the HELP command in the menu whenever the system is waiting for input. Upon entering the help environment a general explanation appears on the screen and a light pen input is expected. By pointing the light pen to the RETURN command of the menu, IGDS exits from the help environment and enters the execution environment. The first time that the execution environment is entered, at the beginning of a session, the menu only remains on the screen. At subsequent times the image existing on the screen when the execution environment was left is restored. In addition, every time that the execution environment is entered, the RETURN command is substituted by the HELP command and the message "READY" appears on the screen. This message appears also at the end of the execution of any command.

Once a command has been selected with the light pen, any additional input is explicitly requested by the system and is entered through the alphanumeric keyboard. The end of an input line is indicated by an EOB character, namely by pressing simultaneously the alternate key and key 5 on the keyboard. If an input is constituted by several lines, the end of the input is signalled by pressing key 0 on the programmed function keyboard after the last EOB.

When using the 2250, the DISPLAY SURFACE command gives further possibilities in addition to those already described in sec.4. The default display consists of the points of the surface connected by segments along the curves and along the stringers. By depressing key 6 and key 7 on the programmed function keyboard,an animated display can be obtained where the points of the surface are connected one after the other in the direction of the stringers or in the direction of the curves, respectively. Key 8 causes the points of the surface to be displayed without any connecting lines and key 5 restores the default display.

6. Using IGDS with a 4013 or 4015

When using IGDS with a Tektronix 4013 or 4015 storage scope, all the interactions pass through the keyboard of the terminal. After IGDS has started, it displays the message READY on the screen and unlocks the keyboard (a small red light is turned off) waiting for a command. This same behaviour is repeated at the end of the execution of every command. The user can enter, in the same line, the command and one of its options or parameters, according to the following table, where one of the bracketed items can be either typed on the same line of the command or omitted. In the latter case IGDS asks the user to specify the option.

BUILD		
CHANGE	[CHGRID TANGENTS POINTS]	
RESTORE	[<name> IMAGE]	
DISPLAY	[FIRST SECOND]	
DISPLAY	[FILE	[<name>]]
DISPLAY	[INPUT SURFACE	[CONIC ISOMETRIC ASSONOMETRIC ORTHOGONAL]]

```
PLOT          [SELECTRIC]
              [EBCDIC   ]


              [FIRST                   ]
STORE         [SECOND                  ]
              [FILE        [<name>]    ]

SAVE          [<name>]

WRITE         [<name>]

END           [APL]
              [CMS]

HELP          [SHORT]
              [LONG ]
```

Any command or option can be abbreviated up to only one letter, having in mind that the search is made according the order shown above. Thus, for example, if the user enters the line S FI, after the message READY has appeared on the screen, this is interpreted as STORE FIRST. If the user enters SF IL, this is interpreted as STORE FILE and the user is requested to give the name of the file into which the image has to be saved (the type is set to IGDSIMAG).

About the commands, the only remark to add to the general description of sec. 4 is that the DISPLAY command does not erase the screen before making the display, which is made in the left portion of the screen.

THe HELP request can be issued any time that the keyboard is unlocked and IGDS is waiting for an input (not only after the READY message). The options SHORT and LONG determine the length of the explanatory message that IGDS displays on the screen (without erasing it before) in reply to the HELP request. If the SHORT option is in effect, a very brief message (one or two lines) is displayed, specifying what kind of input IGDS is waiting for. If the LONG option is in effect, a detailed explanation of the command that has been issued is displayed on the screen.

If no option has be given with the HELP request, the user is not requested to specify it, but the last option specified is considered in effect. HELP LONG is the default option, that is, it is the one with which IGDS starts the very first time. The setting of the option remains in effect from one session to the other.

7. IGDS implementation

We shall give here a brief description of the principal functions and global variables contained in the two workspaces IGDS2250 and IGDS4013. This section is included only as a reference source for maintenance and modification purposes. A complete understanding of this section requires the knowledge of APL, and its reading may be completely skipped by those interested only in the use of IGDS.

IGDS GLOBAL VARIABLES common to both workspaces

NOSTR scalar

is the number of the input stringers

NOCUR scalar

is the number of the input curves

INTST scalar

is the number of the stringers to be interpolated in each patch

INTCU scalar

is the number of the curves to be interpolated in each patch

INPPN 2-dimensional matrix with (NOSTR x NOCUR) rows and 3 columns

each row contains the coordinates x,y,z of an input point. The points are ordered along the curves and then along the stringers.

INTPN 3-dimensional matrix of (NOCUR+INTCUx(NOCUR-1)) planes, where each plane is constituted by (NOSTR+INTSTx(NOSTR-1)) rows and 3 columns.

each row contains the coordinates x,y and z of an interpolated point and each plane contains all the points of one curve. The matrix is filled in with the interpolated points ordered along the curves.

INPTG 2-dimensional matrix with 2 x (NOSTR + NOCUR) rows and 4 columns.

the first 3 columns of each row contain the coordinates along x,y and z axes of the tangent at a border point. the fourth column contains the point number the tangent refers to.

PATCHES 5-dimensional matrix with rank NOCUR, NOSTR,3 4 4

each submatrix of rank 3 4 4 contains the geometrical data of a patch of the surface. The 3 planes correspond to the 3 componaents of the various quantities along the coordinate axes. The laycut of each plane is the same of the matrix M (expression 10) described in sec.3.

IMAGE 3-dimensional matrix. The number of planes and rows is that one of INPPN or INTPN, depending on the option requested with the last DISPLAY command.

each row contains the x and y coordinates of a projected point.

DESIM 5-element vector

contains the description of IMAGE

DESIM(1)=0 if IMAGE contains the projection of the input points

DESIM(1)=1 if IMAGE contains the projection of the interpclated points.

The subsequent elements contain the type and the parameters of the projection requested.

Conic projection:

DESIM(2) = 1

DESIM(3 4 5) contain the coordinates of the wiew point

Assoncmetric projection:

DESIM(2) = 2

DESIM(3 4 5) contain the values of 3 angles

Isometric projection:

DESIM(2) = 3

DESIM(3) = 1,2 or 3 if the axis specified is x,y or z respectively

DESIM(4) contains the value of an angle

DESIM(5) unused

Crthcgonal projection:

DESIM(2) = 4

DESIM(3) = 1,2 cr 3 if the plane specified is xy, xz or yz respectively

DESIM(4 5) unused

TEMP1 3-dimensional matrix

contains a copy of IMAGE and therefore has the same rank of IMAGE

DEST1 5-element vector

contains a copy of DESIM related to IMAGE stored in TEMP1

TEMP2 3-dimensicnal matrix

is set up with the contents cf IMAGE when the command STORE SECOND is executed and therefore has the same rank of IMAGE

DEST2 5-element vector

contains a copy of DESIM related to IMAGE stored in TEMP2

STATUS 10-element vector

STATUS(1) is 0 or 1 depending whether the length of the explanation given when using the 4013 is short or long. The subsequent elements are reserved for future use.

Functions common to both workspaces

ASSON arg(s) left arg: 3-dimensional matrix of the points to be projected

right arg: 3-element vector containing the values of 3 angles in degrees.

result 3-dimensional matrix

the values of the 3 angles α, β, γ are converted from degrees in radiants. The first, the second, and the third column of the left argument are multiplied respectively by

$$\frac{\text{sen arctg} \sqrt{\dfrac{\text{tg}\gamma}{\text{tg}\alpha}}}{\text{sen}\gamma} \qquad \frac{\text{sen arctg} \sqrt{\dfrac{\text{tg}\alpha}{\text{tg}\gamma}}}{\text{sen}\alpha} \qquad \frac{\text{sen arctg} \sqrt{\dfrac{\text{tg}\gamma}{\text{tg}\beta}}}{\text{sen}\gamma}$$

The values obtained in this way are arranged in a matrix of the same rank of the left argument. The explicit result is constituted by the parallel projection of such matrix with parameters $\gamma-\frac{\pi}{2}$, $\alpha-\frac{\pi}{2}$

AUTOMSAVE arg(s) none

result none

saves the IGDS global variables in the file BINSAVE IGDSCONT

COEFFICIENTI arg(s) none
result 2-dimensional matrix

This function computes the matrix of coefficients of the system 7), described in sec 3.1.

CONIC arg(s) left arg: 3-dimensional matrix of the point
to be projected, A
right arg: 3-element vector containing the
coordinates of the view point, V
result 3-dimensional matrix

the center of projection (Q), the direction of sight (COS), and the distance of the plane of projection are evaluated by the following formulas:

$$Q = ((\ulcorner / \ulcorner / A) + \llcorner / \llcorner / A) \div 2$$

$$D = \sqrt{(V_x - Q_x)^2 + (V_y - Q_y)^2 + (V_z - Q_z)^2}$$

$$COS = (Q_x - V_x) \div D \quad , \quad (Q_y - V_y) \div D \quad , \quad (Q_z - V_z) \div D$$

The coordinates X and Y of the projection of a point in space P(x,y,z) represented by a row of the left argument, are obtained by computing:

$$K = D \div \left| (P_x - V_x)\, COS\,[1] + (P_y - V_y)\, COS\,[2] + (P_z - V_z)\, COS\,[3] \right|$$

$$\xi = V_x + K(P_x - V_x) \quad ; \quad \mu = V_y + K(P_y - V_y) \quad ; \quad Z = V_z + K(P_z - V_z)$$

$$X = \left| (\xi - Q_x)\, COS\,[2] - (\mu - Q_y)\, COS\,[1] \right| \div \text{sen arcos } COS\,[3]$$

$$Y = (Z - Q_y) \div \text{sen arcos } COS\,[3]$$

if sen =0 then the coordinates X e Y are evaluated by using the following formulas:

$$X = \left\{ -(\xi - Q_x)\,\mathrm{COS}\,[3] + (Z - Q_z)\,\mathrm{COS}\,[1] \right\} \div \mathrm{sen\ arcos\ COS}\,[2]$$

$$Y = (\mu - Q_y) \div \mathrm{sen\ arcos\ COS}\,[2]$$

CCONS arg(s) none

result none

by using the data contained in the global variables NCCUB, NISTB, INPPN and INPTG, the function COONS defines the global variable PATCHES, according to the procedure described in sec. 3.2.

CURVILINEO arg(s) none

result 7-column matrix

this function solves the system of equation 7) described in sec 3.1, by computing the tangents at the internal points of a cubic curve passing through given points and with given tangents at the ends. The necessary data are defined in CCONS, the function calling CURVINEO. Each row of the result is related to a point of the curve and contains, in the order, the distance from the successive point, the 3 coordinates of the point, the 3 components of the tangent at the point.

DEFVAR arg(s) none

result none

initializes IGDS global variables

DISTA arg(s) 3-column matrix

result vector

given the coordinates of a set of points, this function computes the vector of the distances from a point to the successive one.

FIT arg(s) none

result 3-dimensional matrix with the same format as INTPN

by using the data contained in the global variables NOSTR, NOCUR, INTST, INTCU, INPPN and PATCHES, this function computes the coordinates of all the points to be interpolated on the surface. Equation 9) of sec. 3.2 is solved simultaneously for all the patches of the surface.

F0 arg(s) vector

result scalar

this function computes the value of the blending function F0 (see sec. 3.2) for all the values of the input vector.

F1 arg(s) vector

result scalar

this function computes the value of the blending function F1 (see sec. 3.2) for all the values of the input vector.

G0 arg(s) vector

result scalar

this function computes the value of the blending function G0 (see sec. 3.2) for all the values of the input vector.

G1 arg(s) vector

result scalar

this function computes the value of the blending function G1 (see sec. 3.2) for all the values of the input vector.

ISCM arg(s) left arg: 3-dimensional matrix of the points to be projected

right arg: 2-element vector containing the parameters of the projection (1 2 3 to specify the axes x, y,and z respectively and the value of an angle in degrees)

result 3-dimensional matrix

from the coordinates x, y and z of a point are selected those not corresponding to the axis chosen. The coordinate dropped multiplied by cos is summed to the firt of the 2 coordinates selected. The coordinate dropped multiplied by sen is summed to the second one. The result is constituted by the values so obtained.

NOTO arg(s) a code indicating which component has to be evaluated

result vector

this function computes the vector of known terms in the system 7) described in sec. 3.1. This vector refers to the x, y or z component according to the values 1, 2 or 3 of the argument.

ORTHOGP arg(s) left arg: 3-dimensional matrix of the points to be projected

right arg: a number which specifies the coordinate to be dropped

result 3-dimensional matrix

the result is constituted by the left argument without the column specified

PLOTHP arg(s) none

result none

the contents of IMAGE is scaled and then plotted on the plotter HP 2701-A.

PLOTG arg(s) 2-column matrix

result none

each row contains the coordinates of a point. This function plots a segment connecting each pair of adjacent rows (point 1 connected to point 2, point 3 connected to point 4, and so on)

PLOTP arg(s) 2-column matrix

result none

each row contains the coordinates of a point. This function plots a point for each row without a connecting segment.

PLOTV arg(s) 2-column matrix

result none

each row contains the coordinates of a point. This function plots a sequence of segments connecting the points in the order of the rows.

PROSPAR arg(s) left arg: 3-dimensional matrix of the points to be projected

right arg: 2-element vector specifying 2 angles α_1 α_2

result 3-dimensional matrix

the coordinates X and Y of the projection of a point P (contained in a row of the left argument) are obtained by

$$X = P_x \cos\alpha_1 - (P_z + P_x \,\mathrm{sen}\,\alpha_1 + P_y \,\mathrm{sen}\,\alpha_1)$$

$$Y = P_y \cos\alpha_2 - (P_z + P_y \,\mathrm{sen}\,\alpha_2 + P_y \,\mathrm{sen}\,\alpha_2)$$

RESTORE arg(s) none
result none

sets up the global variables of IGDS with the contents of the file BINSAVE IGDSCONT.

SCALE arg(s) right arg: 3-dimensional matrix of the coordinates of the points to be scaled
left arg: 4-element numeric vector specifying the coordinates of the origin of the axes and the greatest value possible of the X and the Y cocrinate
result 3-dimensional matrix

the right argument is scaled according to the parameters specified by the left argument. The result has the same rank as the right argument.

STATE arg(s) a character vector containing the name and the type of a file separated by a blank
result 0 or 1

the result is 1 if the file specified exists on disk, 0 otherwise

WR1 arg(s) right arg: the matrix to write in a file
left arg: 10-element numeric vector

result none

the left argument is constituted by:

V(1) = 1 or 0 to specify if a new file has to be written, or new records have to be added to an already existing file

V(2) = 1, 2, 3, or 4 to specify if the data are in integer, character, floating-point or exponential form

V(3 4 5) specify the format of writing

V(6 7 8 9 10) specify the APLNAME of the file

According to the parameters specified by the left argument, this function writes a new file with the contents of the right argument or adds records to an existing file.

The following functions, that are common to both workspaces, have been copied from the workspace 1 IOFNS and are described in the APLCMS user's Manual.

APLNAME

CMS

CMSNAME

DFT

EFT

FILESIZE

IREAD

OF

READERRCR

WRITE

WRITEERROR

7.1 The workspace IGDS2250

The main function of this workspace is the function START, which must be called to get a session started. The call to START must be preceded by a call to the function INIT, which sets up the global variables needed by IGDS either with null values, if the session is the very first one, or with values read from disk, if the session is the continuation of a preceding one. The function START initializes the graphic environment and enables all the attention sources of the 2250. Subsequently calls the function MENUGEN to prepare the display of the menu. Then the function HELP is called to enter the HELP environment. When the execution environment is entered, this function waits for a light pen attention, recognizes the command selected and calls the function EXVOM in order to execute the command. When the execution of the command is terminated (normally or not), this function waits again for a new light pen attention, and so on.

The flow of control in the workspace IGDS2250 is indicated schematically in the diagram of fig. 10

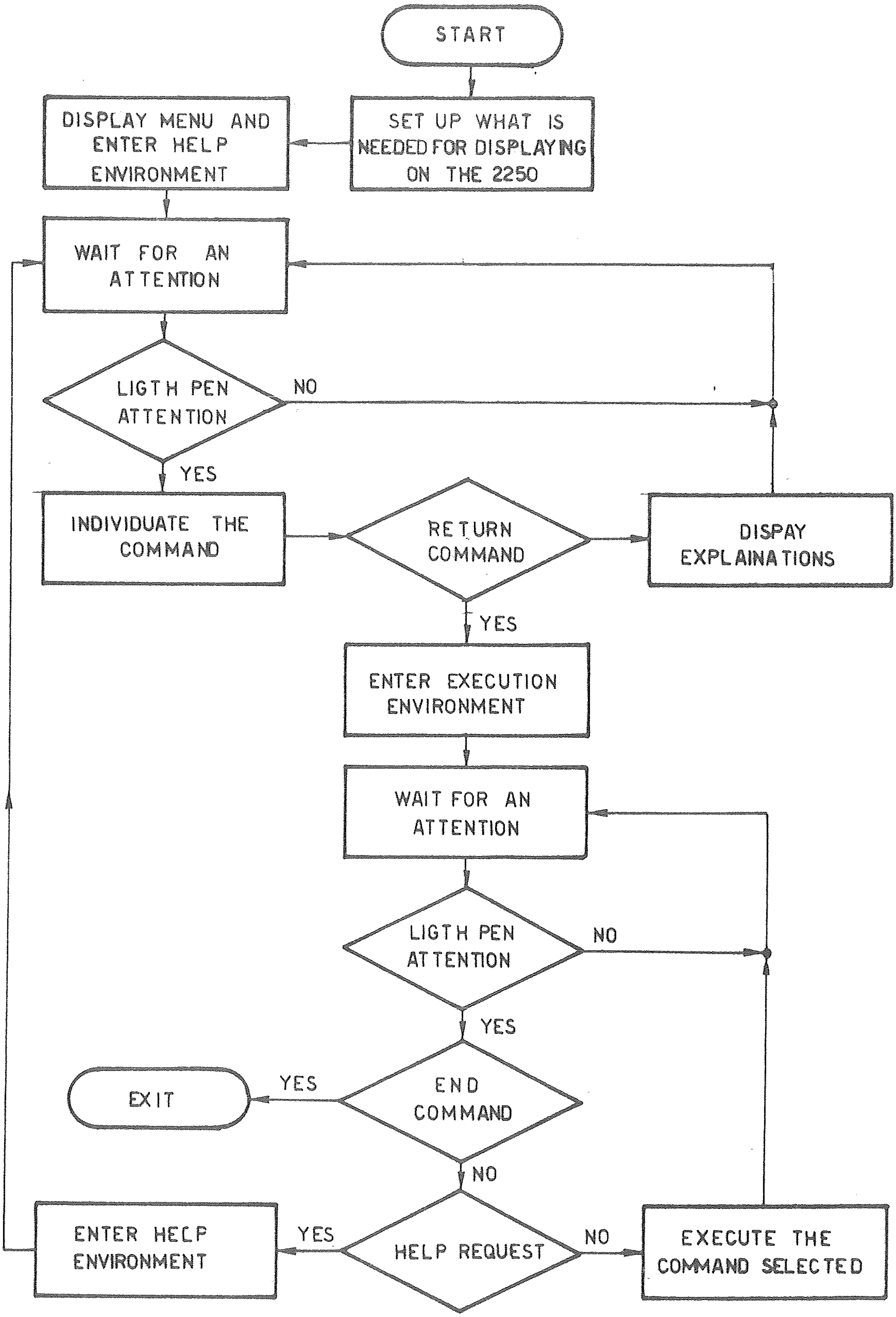

START
SET UP WHAT IS NEEDED FOR DISPLAYING ON THE 2250
DISPLAY MENU AND ENTER HELP ENVIRONMENT
WAIT FOR AN ATTENTION
LIGTH PEN ATTENTION
NO
YES
INDIVIDUATE THE COMMAND
RETURN COMMAND
DISPAY EXPLAINATIONS
YES
ENTER EXECUTION ENVIRONMENT
WAIT FOR AN ATTENTION
LIGTH PEN ATTENTION
NO
YES
EXIT
YES
END COMMAND
NO
ENTER HELP ENVIRONMENT
YES
HELP REQUEST
NO
EXECUTE THE COMMAND SELECTED

fig. 10

Functions used only in the workspace IGDS2250

ASSOMP arg(s) right arg: 3-dimensional matrix of the points to be projected

left arg: empty vector or 3-element vector of the values of 3 angles

result none

if the left argument is an empty vector, the values of 3 angles are read from the keyboard. Then a call is made to the function ASSON to project the right argument. The global variables IMAGE and DESIM are properly set up.

BUILD arg(s) none

result none

executes the command BUILD

CHGRID1 arg(s) none

result none

verifies if the command CHANGE GRID is executable and, if so, calls the function CHGRID to execute the command

CHGRID arg(s) none

result none

executes the command CHANGE GRID

CHPOINT1 arg(s) none

result none

verifies if the command CHANGE POINT is executable and in such case calls the function CHPOINT to execute the command.

CHPOINT arg(s) none

result none

executes the command CHANGE POINT

CHTANG1 arg(s) none

result none

verifies if the command CHANGE TANGENT is executable and in such case calls the function CHTANG to execute the command

CHTANG arg(s) none

result none

executes the command CHANGE TANGENT

COMPIC arg(s) none

result none

executes the command PLOT

CCMRES arg(s) none

result none

executes the command RESTORE

CCMSAVE arg(s) none

result none

executes the command SAVE

COMTEMP1 arg(s) none

result none

executes the command STORE FIRST

COMTEMP2 arg(s) none

result none

executes the command STORE SECOND

CCMWRI arg(s) none

result none

executes the command WRITE. The global variables STATUS, NOCUR, NOSTR, INTCU, INTST, PATCHES, INPPN, INPTG, INTPN are written in a file on disk in card image with the following FORTRAN format:

```
10I5    /    STATUS
4I5     /    NOCUR, NOSTR, INTCU, INTST   (1 card)
```

```
2I5        /      INPPN    (1 card)
(8F10.3)   /    , INPPN  (as many as needed)
2I5        /      INPTG  (1 card)
(8F10.3)   /    , INPTG  (as many as needed)
3I5        /      INTPN  (1 card)
(8F10.3)   /    , INTPN  (as many as needed)
5I5        /      PATCHES   (1 card)
(8F10.3)   /    , PATCHES  (as many as needed)
```

CONICP arg(s) right arg: 3-dimensional matrix of the points to be projected
left arg: empty vector or 3-element vector of the coordinates of the view point
result none

if the left argument is an empty vector, the coordinates of the view point are read from the keyboard. Then a call is made to the function CONIC to project the right argument. The global variables DESIM and IMAGE are properly set up.

DISPINP arg(s) none
result none

executes the command DISPLAY INPUT

DISPOBJ arg(s) none
result 0 or 2

executes the command DISPLAY SURFACE. If no current surface is defined, or the type of the projection chosen is not a valid one, the result is 0. = Otherwise the interpolated points are projected and the proper display is made by calling the function DSURF.

DISPNAM arg(s) none

result 0 or 2

executes the command DISPLAY FILE. The name of the file to be displayed is read from the keyboard. If the file with the given filename and type IGDSIMAG does not exist or contains the input points, the result is set to 0. If the file contains the interpolated points, the result is set to 2

DISPT1 arg(s) none

result 0 or 2

executes the command DISPLAY FIRST. If the global variable TEMP1 is empty or contains a projection of the input points, the result is set to 0. If TEMP1 contains a projection of the input points, the result is set to 2.

DISPT2 arg(s) none

result 0 or 2

executes the command DISPLAY SECOND. If the global variable TEMP2 contains a projection of the input points or is empty, the result is set to 0. If TEMP2 contains a projection of the interpolated points, the result is set to 2.

DISP1 arg(s) none
result none

displays on the screen of the 2250 the explanations about IGDS.

DISP2 arg(s) none
result none

displays on the screen of the 2250 the menu of commands.

DPOINT arg(s) none
result none

the global variable IMAGE containing a projection of the input points is scaled. The tangents not equal to 0 at the border points of the surface are visualized with arrows. To make this,the coordinates of the end of every arrow are calculated. The input points, the arrows and the parameters of the projection used are then displayed on the screen.

DSURF arg(s) none
result none

the global variable IMAGE containing a projection of the interpolated surface is scaled. The points are displayed connected with segments along the stringer and along the curves. If a PFK attention occours, the proper display is made. If the HELP command is selected, the HELP environment is entered. When the execution environment is reentered, this function waits for a new attention.

ENDAPL arg(s) none
result none
executes the END APL command.

ENDCMS arg(s) none
result none
executes the END CMS command.

EXCOM arg(s) an integer number between 1 and 19
result 0, 1 or 2
calls the functions which execute the IGDS commands, as the argument specifies. The result is that one of the function called is the function has an explicit result, otherwise is 0.

EXGEN arg(s) right arg: an integer number between 0 and 11
left arg: an integer number between 0 and 6
result none
prepares the IGDS messages to be displayed on the screen. The left argument selects an error message. The

right argument selects a message describing the additional input for the command being executed. If both arguments are 0, the execution of this function causes the disappearance of IGDS messages displayed on the screen.

GETINP arg(s) none

result an empty vector or a character vector

waits for an input from the user unlocking the keyboard. After the user has entered an input string and has given an EOB attention, the input string is read and constitutes the result. If the user before giving the ECB attention asks for the HELP command, the HELP environment is entered and the keyboard is locked. When the execution environment is reentered,an input line is requested again tc the user. If the user signals the end of an input constituted by several lines by pressing the key 0 on the PFK, the result of this function is an empty vector.

HELP arg(s) none

result none

executes the command HELP. Makes a copy of the buffer of the 2250, calls the function SPIEG and DISP1 to display the explanations about IGDS and then waits for a light pen attention. If the RETURN command has been selected, this function restores the previous image on the screen, otherwise calls again the functions SPIEG and DISP1 to display the explanation related to the command.

IMDESCR arg(s) none
result none

displays on the screen the type and the parameters of the projection of the image appearing on the screen.

ISOMP arg(s) right arg: 3-dimensional matrix of the points to be projected
left arg: an empty vector of 3-element vector
result none

If the left argument is an empty vector this function waits for the input line constituted by the parameters of the projection and reads it. Then a call is made to the function ISOM to project the right argument.The global variables DESIM and IMAGE are properly set up.

MENUGEN arg(s) none
result none

prepares the menu to be displayed on the screen.

ORTHOGP arg(s) right arg: 3-dimensional matrix of the point to be projected.
left arg: an empty vector or a 2-element character vector which specifies the plane of projection.

result none

if the left argument is an empty vector, this function waits for an input line containing the parameter of the projection and reads it. Then a call is made to the function ORTHOG to project the right argument. The global variables IMAGE and DESIM are properly set up.

PRCIEZ arg(s) 3-dimensional matrix of the points to be projected

result 0 or 1

waits for an input line containing the type of projection by which the argument has to be projected and the parameters of the projection. After the line has been read, if the specified type is correct, the argument is projected and the result is set to 0. Otherwise an error message is displayed and the result is set to 1.

RDGEN arg(s) 1 Or 2

result none

if the argument is 1, the function prepares the display of the message 'READY'. If the argument is 2, the message 'READY' displayed on the screen disappears.

SPIEG arg(s) an integer number between 1 and 19

result none

this function reads from the file EXPL IGDSEXPL the record specified by the argument. the file EXPL IGDSEXPL is in character form. The first record contains the general explanations about IGDS which are displayed upon entering the HELP environment. The subsequent records contain the EXplanations about the use of the commands in the same sequence as the commands appear on the menu.

STFILE arg(s) none
result none
executes the command STORE FILE

VERIF arg(s) a character vector
result the scalar 0 or a numeric vector
this function checks if the argument represents a legal API number. If so, the character vector is converted in a numeric vector which constitutes the explicit result. Otherwise the result is the scalar 0.

The following functions correspond to the routines with the same name of the package Fortran Graphic Support (FGS) and have been copied from the workspace 1 FGSFNS. Their description can be found in the APLFGS Technical Report or in the FGS User's Manual.

```
ERDISP
GECBF
GECBP
GEPM
GEVM
GEXEC
GSRT
GSTCP
GTEXT
GTRU
GXY
GXYP
INBA
INCSR
INGA
KATNT
KATNW
KDSP
KERR
RDBA
RDTC
RDXYP
READY
RESET
RMCSR
SADR
SDSP
SPFR
TMBA
TMGA
WRBA
```

Complete cross-reference of the workspace IGDS2250

```
FUNCTIONS              CALLED FUNCTIONS       CALLING FUNCTIONS

------------------------------------------------------------------
APLNAME                OF                     AUTOMSAVE
                                              COMWRI
                                              COMRES
                                              DISPNAM
                                              LEGGI
                                              RESTORE
                                              SPIEG
                                              STFILE
------------------------------------------------------------------
ASSON                  PROSPAR                ASSONP
------------------------------------------------------------------
ASSONP                 EXGEN                  PROIEZ
                       GEXEC1
                       BLANK
                       GETINP
                       VERIF
                       ASSON
------------------------------------------------------------------
AUTOMSAVE              CMS                    CHGRID
                       WRITE                  COMTEMP1
                       APLNAME                COMTEMP2
                                              PROIEZ
------------------------------------------------------------------
BLANK                                         ASSONP
                                              BUILD
                                              CHPOINT
                                              CONICP
                                              ISOMP
                                              ORTHOGP
                                              PROIEZ
------------------------------------------------------------------
BUILD                  EXGEN                  EXCOM
                       GEXEC1
                       BLANK
                       GETINP
                       VERIF
                       CHPOINT
------------------------------------------------------------------
CHGRID                 EXGEN                  CHGRID1
                       GEXEC1                 CHTANG
                       GETINP
                       BLANK
                       VERIF
                       FIT
                       AUTOMSAVE
                       SPFK
```

CHGRID1	EXGEN GEXEC1 SPFK CHGRID	EXCOM
CHPOINT	SPFK GEXEC1 GETINP BLANK VERIF CHTANG EXGEN	BUILD CHPOINT1
CHPOINT1	EXGEN GEXEC1 CHPOINT	EXCOM
CHTANG	EXGEN GEXEC1 GETINP BLANK VERIF CCONS SPFK CHGRID	CHPOINT CHTANG1
CHTANG1	EXGEN GEXEC1 SPFK CHTANG	EXCOM
CL	KDSP SADR WRBA SDSP	GETINP
CMS		AUTOMSAVE CL COMWRI RESTORE STATE STFILE WRT
CMSNAME	OF	WRT
COEFFICIENTI		CURVILINEO
COFPLC	EXGEN GETINP PLOTHP GEXEC1	EXCOM
COMRES	EXGEN IREAD APLNAME GEXEC1 GETINP	EXCOM

```
                         STATE
                         OF
                         SDSP
                         TXT
                         DFT
                         GTBU
                         SADR
                         WREA
                         KDSP
-----------------------------------------------------------------
CONTEMP1                 AUTOMSAVE             EXCOM
                         EXGEN
                         GEXEC1
-----------------------------------------------------------------
CONTEMP2                 AUTOMSAVE             EXCOM
                         EXGEN
                         GEXEC1
-----------------------------------------------------------------
CONWRI                   EXGEN                 EXCOM
                         GEXEC1
                         GETINP
                         CMS
                         APLNAME
                         WRT
-----------------------------------------------------------------
CONIC                                          CONICP
-----------------------------------------------------------------
CONICP                   EXGEN                 PROIEZ
                         GETINP
                         GEXEC1
                         BLANK
                         VERIF
                         CONIC
-----------------------------------------------------------------
COONS                    CURVILINEO            CMTANG
-----------------------------------------------------------------
CURVILINEO               DISTA                 COONS
                         NOTC
                         COEFFICIENTI
-----------------------------------------------------------------
DEFVAR                                         RESTORE
-----------------------------------------------------------------
DFT                                            COMRES
                                               IMDESCR
                                               WRT
-----------------------------------------------------------------
DISPINP                  EXGEN                 EXCOM
                         GEXEC1
                         PROIEZ
                         DPCINT
-----------------------------------------------------------------
DISPNAM                  EXGEN                 EXCOM
                         GEXEC1
                         GETINP
                         STATE
                         OF
                         IREAD
                         APLNAME
                         DPOINT
                         DSURF
------------------------------------------------------------------
DISPOBJ                  EXGEN                    EXCOM
                         GEXEC1
                         PROIEZ
                         DSURF
------------------------------------------------------------------
DISPT1                   EXGEN                    EXCOM
                         GEXEC1
                         DPOINT
                         DSURF
------------------------------------------------------------------
DISPT2                   EXGEN                    EXCOM
                         GEXEC1
                         DPOINT
                         DSURF
------------------------------------------------------------------
DISP1                    SADR                     HELP
                         WREA
                         SDSP
                         GEXEC1
------------------------------------------------------------------
DISP2                    SADR                     HELP
                         SDSP
                         WREA
                         GEXEC
------------------------------------------------------------------
DISTA                                             CURVILINEO
------------------------------------------------------------------
DPOINT                   SDSP                     DISPINP
                         SCALE                    DISPNAM
                         GEPM                     DISPT1
                         GXY                      DISPT2
                         GEVM
                         GXYP
                         INDESCR
                         GTRU
                         SADR
                         WREA
                         KDSP
                         GEXEC1
------------------------------------------------------------------
DSURF                    SPFK                     DISPNAM
                         SCALE                    DISPOBJ
                         SDSP                     DISPT1
                         INDESCR                  DISPT2
                         KDSP
                         GEVM
                         GXYP
                         GXY
                         GTRU
                         SADR
                         WREA
                         RDGEN
                         GEXEC1
                         KATNT
                         KATNW
                         HELP
------------------------------------------------------------------
```

```
ENDAPL                  RESET                   EXCOM
-------------------------------------------------------------
ENDCMS                  RESET                   EXCOM
-------------------------------------------------------------
ERDSP                                           START
-------------------------------------------------------------
EXCOM                   EXGEN                   START
                        GEXEC1
                        ENDAPL
                        ENDCMS
                        BUILD
                        CHPOINT1
                        CHTANG1
                        CHGRID1
                        DISPINP
                        DISPOBJ
                        DISPNAM
                        DISPT1
                        DISPT2
                        STFILE
                        CONTEMP1
                        CONTEMP2
                        COMSAVE
                        COMRES
                        COMPLO
                        COMWRI
                        HELP
-------------------------------------------------------------
EXGEN                   SDSP                    ASSONP
                        TXT                     BUILD
                        GTRU                    CHGRID
                        SADE                    CHGRID1
                        WRBA                    CHPOINT
                        KDSP                    CHPOINT1
                                                CHTANG
                                                CHTANG1
                                                COMPLO
                                                COMRES
                                                CONTEMP1
                                                CONTEMP2
                                                COMWRI
                                                CONICP
                                                DISPINP
                                                DISPNAM
                                                DISPOBJ
                                                DISPT1
                                                DISPT2
                                                EXCOM
                                                GETINP
                                                ISOMP
                                                ORTHOGP
                                                PROIEZ
                                                STFILE
-------------------------------------------------------------
FIT                     F0                      CHGRID
                        G0
                        F1
                        G1
-------------------------------------------------------------
```

```
F0                                              FIT
--------------------------------------------------------------
F1                                              FIT
--------------------------------------------------------------
GECBF                                           INPGEN
--------------------------------------------------------------
GECBP                                           TXT
--------------------------------------------------------------
GEFM                                            DPOINT
                                                TXT
--------------------------------------------------------------
GETINP                 CL                       ASSONP
                       GEXEC1                   BUILD
                       INCSR                    CHGRID
                       KATNT                    CHPOINT
                       KATNW                    CHTANG
                       RDXYP                    COMPLO
                       RMCSR                    COMRES
                       HELP                     COMWRI
                       RDTC                     DISPNAM
                       EXGEN                    ISOMP
                                                ORTHOGP
                                                PROIEZ
                                                STFILE
                                                CONICP
--------------------------------------------------------------
GEVM                                            DPOINT
                                                DSURF
                                                INPGEN
                                                MENUGEN
--------------------------------------------------------------
GEXEC                                           DISP2
--------------------------------------------------------------

                                                BUILD
GEXEC1             GEXEC                        ASSONP
                                                CHGRID
                                                CHGRID1
                                                CHPOINT
                                                CHPOINT1
                                                CHTANG
                                                CHTANG1
                                                COMPLO
                                                COMRES
                                                COMTEMP1
                                                COMTEMP2
                                                COMWRI
                                                CONICP
                                                DISPINP
                                                DISPNAM
                                                DISPOBJ
                                                DISPT1
                                                DISPT2
                                                DISP1
                                                DPOINT
                                                DSURF
                                                EXCOM
                                                START
                                                GETINP
                                                ISOMP
```

```
                                                         ORTHOGP
                                                         PROIEZ
                                                         STFILE
------------------------------------------------------------------
GSRT                                                     INPGEN
------------------------------------------------------------------
GTEXT                                                    INPGEN
                                                         TXT
------------------------------------------------------------------
GTEU                                                     COMRES
                                                         DPOINT
                                                         DSURF
                                                         EXGEN
                                                         MENUGEN
                                                         RDGEN
                                                         SPIEG
------------------------------------------------------------------
GXY                                                      DPOINT
                                                         DSURF
                                                         MENUGEN
------------------------------------------------------------------
GXYP                                                     DPOINT
                                                         TXT
                                                         DSURF
                                                         INPGEN
                                                         MENUGEN
------------------------------------------------------------------
G0                                                       FIT
------------------------------------------------------------------
G1                                                       FIT
------------------------------------------------------------------
HELP                       SADE                          DSURF
                           SDSP                          EXCOM
                           RDEA                          START
                           SPIEG                         GETINP
                           DISP1
                           KATNT
                           KATNW
                           RDXYP
                           DISP2
------------------------------------------------------------------
INLESCR                    TXT                           DPOINT
                           DPT                           DSURF
------------------------------------------------------------------
INCSR                                                    GETINP
------------------------------------------------------------------
INGA                                                     START
                                                         INPGEN
                                                         MENUGEN
------------------------------------------------------------------
INPGEN                     INGA                          START
                           GSRT
                           GEVM
                           GXYP
                           GECBP
                           KDSP
                           GTEXT
------------------------------------------------------------------
IREAD                      READERROR                     DISPNAM
```

```
                                                      COMRES
                                                      RESTORE
-----------------------------------------------------------------
ISOM                                                  ISOMP
-----------------------------------------------------------------
ISCMP                       EXGEN                     PROIEZ
                            GEXEC1
                            BLANK
                            GETINP
                            VERIF
                            ISCM
-----------------------------------------------------------------
KATNT                                                 DSURF
                                                      START
                                                      GETINP
                                                      HELP
-----------------------------------------------------------------
KATNW                                                 DSURF
                                                      START
                                                      GETINP
                                                      HELP
-----------------------------------------------------------------
KDSP                                                  CL
                                                      COMRES
                                                      DPOINT
                                                      DSURF
                                                      EXGEN
                                                      START
                                                      INPGEN
                                                      MENUGEN
-----------------------------------------------------------------
MENUGEN                     INGA                      START
                            TXT
                            GEVM
                            GXYP
                            GXY
                            KDSP
                            GTRU
-----------------------------------------------------------------
NK                                                    VERIF
-----------------------------------------------------------------
NOTO

                                                      CURVILINEO
-----------------------------------------------------------------
OF                                                    APLNAME
                                                      CMSNAME
                                                      COMRES
                                                      DISPNAM
                                                      VERIF
-----------------------------------------------------------------
ORTHOG                                                ORTHOGP
-----------------------------------------------------------------
ORTHOGP                     EXGEN                     PROIEZ
                            GEXEC1
                            BLANK
                            GETINP
                            ORTHOG
-----------------------------------------------------------------
PICTG                                                 PLOTHP
```

```
------------------------------------------------------------------------
PLOTHP                     SCALE                      COMPLO
                           PLOTG
                           PLOTP
                           PLOTV
------------------------------------------------------------------------
PLOTP                                                 PLOTHP
------------------------------------------------------------------------
PLOTV                                                 PLOTHP
------------------------------------------------------------------------
PRCSPAR                                               ASSON
------------------------------------------------------------------------
PROIEZ                     EXGEN                      DISPINP
                           GEXEC1                     DISPOBJ
                           GETINP
                           BLANK
                           CONICP
                           ISOMP
                           ASSONP
                           ORTHOGP
                           AUTOMSAVE
------------------------------------------------------------------------
RDEA                                                  HELP
------------------------------------------------------------------------
RDGEN                      SDSP                       DSURF
                           TXT                        START
                           GTRU
                           SADR
                           WRBA
------------------------------------------------------------------------
RDTC                                                  GETINP
------------------------------------------------------------------------
RDXYP                                                 START
                                                      GETINP
                                                      HELP
------------------------------------------------------------------------
READ                       READERROR                  LEGGI
                                                      SPIEG
------------------------------------------------------------------------
READERROR                                             IREAD
                                                      READ
------------------------------------------------------------------------
READY                                                 START
------------------------------------------------------------------------
RESET                                                 ENDAPL
                                                      ENDCMS
                                                      START
------------------------------------------------------------------------
RESTORE                    STATE
                           DEFVAR
                           CMS
                           IREAD
                           APLNAME
------------------------------------------------------------------------
RMCSR                                                 GETINP
------------------------------------------------------------------------
SAIR                                                  CL
                                                      COMRES
                                                      DISP1
```

		DISP2 DPOINT DSURF EXGEN START HELP RDGEN
SCALE		DPOINT DSURF PLOTHP
SDSP		CL COMRES DISP1 DISP2 DPOINT DSURF EXGEN HELP RDGEN SPIEG
SPIK		CHGRID CHGRID1 CHPOINT CHTANG CHTANG1 DSURF
SPIEG	SDSP TXT READ APLNAME GTRU	HELP
START	RESET READY ERDSP INGA INPGEN MENUGEN WRBA SALE KDSP HELP RDGEN GEXEC1 KATNT KATNW RDXYP EXCCM	
STATE	CMS	COMRES DISPNAM RESTORE
STFILE	EXGEN GEXEC1	EXCOM

```
                       GETINP
                       CMS
                       WRITE
                       APLNAME
------------------------------------------------------------
TXT                    GEPM                   COMRES
                       GXYP                   EXGEN
                       GECBP                  IMDESCR
                       GTEXT                  MENUGEN
                                              RDGEN
                                              SPIEG
------------------------------------------------------------
VERIF                  NK                     ASSONP
                       OP                     BUILD
                                              CHGRID
                                              CHPOINT
                                              CHTANG
                                              CONICP
                                              ISOMP
------------------------------------------------------------
WREA                                          COMRES
                                              CL
                                              DISP1
                                              DISP2
                                              DPOINT
                                              DSURF
                                              EXGEN
                                              START
                                              RDGEN
------------------------------------------------------------
WRITE                  WRITEERROR             AUTOMSAVE
                                              STFILE
                                              WRT
------------------------------------------------------------
WRITEERROR                                    WRITE
------------------------------------------------------------
WRT                    CMS                    COMWRI
                       CMSNAME
                       DFT
                       EFT
                       WRITE
------------------------------------------------------------
```

7.2 The workspace IGDS4013

The main function of the workspace IGDS4013 is the function START, which must be called to get a session started. The call to START must be preceded by a call to the function INIT, which sets up the global variables needed by IGDS either with null values, if the session is the very first one, or with values read from disk, if the session is the continuation of a preceding one. The function START writes the message "READY" on the screen of the terminal and calls the function GETCOM which waits for a command and its parameters. Upon return from the function GETCOM, if no error has been found, the appropriate function is called to execute the command. At the end of the execution, the message "READY" is printed again, the function GETCOM is called, and so on until the command END is issued.

The flow of control in the workspace IGDS4013 is indicated schematically in the diagram of fig. 11

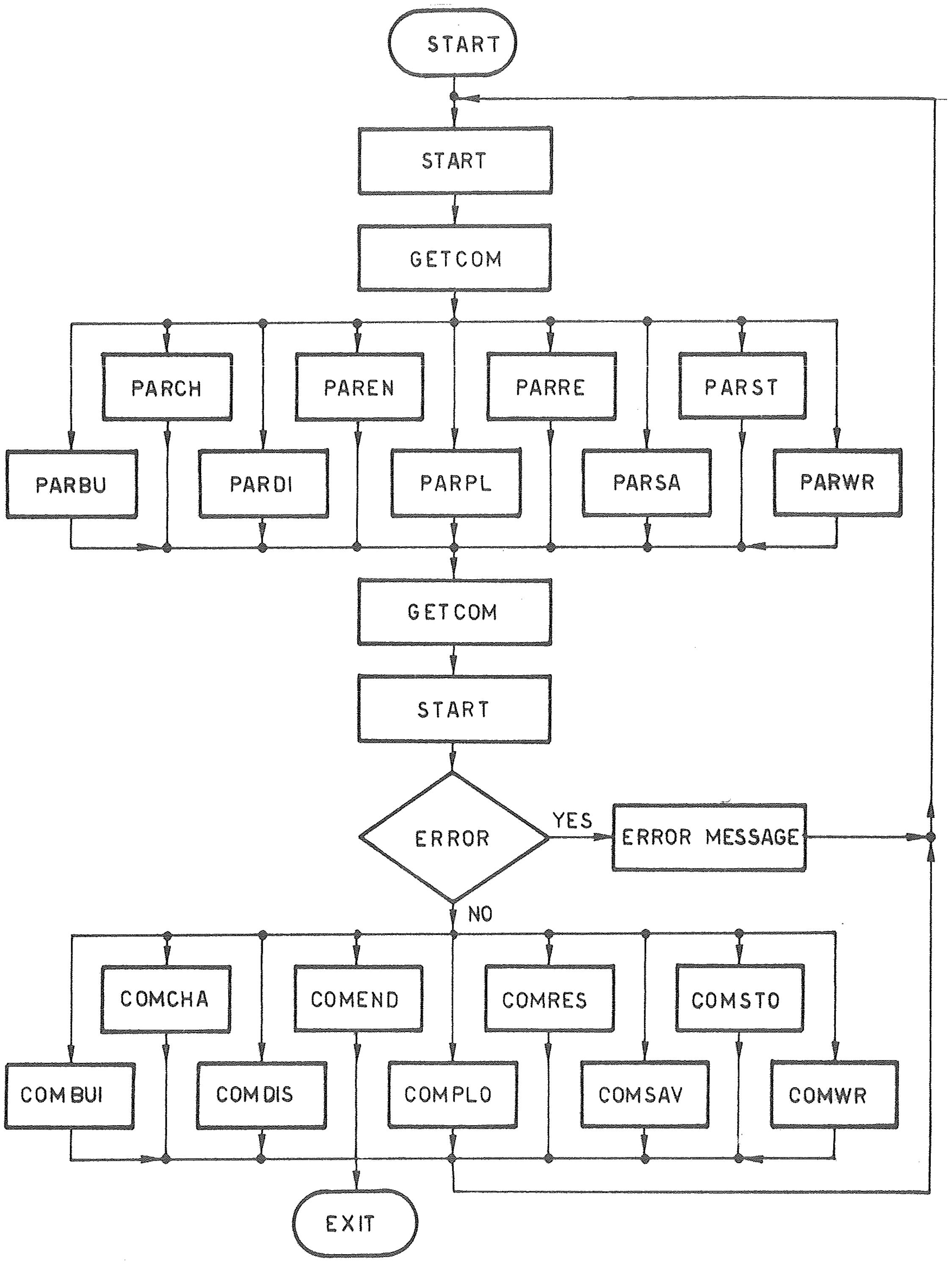

START
START
GETCOM
PARCH
PAREN
PARRE
PARST
PARBU
PARDI
PARPL
PARSA
PARWR
GETCOM
START
ERROR
YES
ERROR MESSAGE
NO
COMCHA
COMEND
COMRES
COMSTO
COMBUI
COMDIS
COMPLO
COMSAV
COMWR
EXIT

fig. 11

Functions used only in the workspace IGDS4013

ALFINP arg(s) a code indicating the function calling ALFINP

result a character string read from the keyboard

after having read the keyboard, a check is made to see if it is a HELP request, with or without length specification. If so, the status is changed according to the request or is left unchanged, and the function EXPL is called, passing to it the code of ALFINP caller. Upon return from EXPL, a new reading is made from the keyboard.

CHGRID arg(s) a code indicating the caller of CHGRID

result none

the global variables INTCU and INTST are set according to the values obtained from the keyboard.

CHPOINT arg(s) a code indicating the caller of CHPOINT

result none

the global variable INPPN is changed with the values obtained from the keyboard

CHTANG arg(s) a code indicating the caller of CHTANG

result none

one or more entries of the global variable INPTG are changed with the values obtained from the keyboard.

COMBUI arg(s) an empty vector. The argument is not used in the actual implementation

result none

executes the command BUILD. By reading from the keyboard and by calling the functions CHPOINT, CHTANG, CHGRID, COONS and FIT, sets the new values for the following global variables: NOCUR, NOSTR, INPPN, INPTG, INTCU, INTST, INTPN.

COMCHA arg(s) a 1-element vector specifying the option chosen

result none

executes the command CHANGE. According to the value of the argument and by calling CHGRID, CHTANG and CHPOINT executes the three options of the command CHANGE, namely CHANGE GRID, CHANGE POINTS, CHANGE TANGENTS, changing the values of the global variables involved in the change.

COMDIS arg(s) a vector specifying what to display, the type of projection and the parameters of the projection

result none

executes the command DISPLAY. After having set the global variables DESIM and IMAGE with the appropriate values, the function DISPL is called to perform the actual display.

COMEND arg(s) a code specifying the option chosen

result none

executes the command END. Note that if the option chosen has been END CMS, this function works only under APLCMS.

COMPLO arg(s) a code specifying the type of terminal

result none

executes the command PLOT.

COMRES arg(s) a vector containing the APLNAME of the file to be read

result none

executes the command RESTORE. The global variables of IGDS are set with new values read from a binary file on disk.

COMSAV arg(s) a vector containing the APLNAME of the file to be written.

result none

executes the command WRITE. The global variables NOCUR, NOSTR, INTCU, INTST, INPPN, INPTG, INTPN, PATCHES are written in a binary file on disk.

COMSTO arg(s) a vector specifying the option chosen and, if required, the APLNAME of the file to be written

result none

executes the command STORE. Depending on the option chosen, the global variables DESIM and IMAGE are copied into the global variables DEST1 and TEMP1 or DEST< and TEMP2, or are written in a binary file on disk.

COMWRI arg(s) a vector containing the APLNAME of the file to be written

result none

executes the command WRITE. The global variables STATUS, NOCUR, NOSTR, INTCU, INTST, PATCHES, INPPN, INPTG, INTPN are written in a file on disk in card image with the following FORTRAN format:

10I5	/	STATUS
4I5	/	NOCUR, NOSTR, INTCU, INTST (1 card)
2I5	/	INPPN (1 card)
(8F10.3)	/	, INPPN (as many as needed)
2I5	/	INPTG (1 card)
(8F10.3)	/	, INPTG (as many as needed)
3I5	/	INTPN (1 card)
(8F10.3)	/	, INTPN (as many as needed)
5I5	/	PATCHES (1 card)
(8F10.3)	/	, PATCHES (as many as needed)

DESCR arg(s) the vector DESIM

result a character string

the result is a description of the displayed image and will be displayed on the screen.

DISPLA arg(s) none

result none

the contents of IMAGE is scaled and then displayed on the screen of the 4013, together with a brief description of the image.

EXPL arg(s) a code indicating the caller

result none

this function is called by ALFINP or NUMINP if the user issues an HELP request instead of entering the required input. The code indicating the caller of ALFINP or NUMINP is passed to EXPL so that the appropriate explanations can be given. If the length indicator is set to long, the function EXPLONG is called

EXPLONG arg(s) a code indicating the caller of EXPLONG

result none

this function reads from a diskfile the appropriate explanations, as indicated by the requirement, and displays them on the screen.

GETCOM arg(s) none

result a vector encoding the command and its parameters

this function gets a line from the user and search in the command table for the first word of the line. If the command is found, the appropriate function is called to examine the rest of the line and to ask for further parameters, if they are needed. Upon return, if the return ERROR code is zero, the result is set to the command number, catenated with an encoding of its parameters. If the command is not found in the table or if the called functions returns with an error code ≠ 0, the result is set to 0.

LUT arg(s) right arg: a character string

left arg: a characer matrix

result a row number

this function looks up the left argument for the first row matching the character string up to the length of the character string itself. The result is set to to the row number if a match is found, otherwise is set to 0.

NUMINP arg(s) a code indicating the function calling NUMINP

result a numerical value read from the keyboard after having read the keyboard, a check is made to see if it is a HELP request, with or without length specification. If so, the status is changed according to the request or is left unchanged and the function EXPL is called, passing to it the code of NUMINP caller. Upon return from EXPL, a new reading is made from the keyboard.

OUT arg(s) a character string

result none

this function handles the length of the line sent to the terminal, when it is used in graphic mode.

PARBU arg(s) a character string

result none

the argument contains what follows the command BUILD in a input line. This function checks only that the line is empty, otherwise an error indicator is set.

PARCH arg(s) a character string

result none

the argument contains what follows the command CHANGE in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result accordingly.

PARDI arg(s) a character string
result none

the argument contains what follows the command DISPLAY in an input line. This function checks the correctness of the option and parameter specified (either in the argument or requesting it to the user) and encodes them in the result.

PAREN arg(s) a character string
result none

the argument contains what follows the command END in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result accordingly.

PARPI arg(s) a character string
result none

the argument contains what follows the command PICT in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result accordingly.

PARRE arg(s) a character string
result none

the argument contains what follows the command RESTORE in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result accordingly.

PARSA arg(s) a character string
result none

the argument contains what follows the command SAVE in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result accordingly.

PARST arg(s) a character string
result none

the argument contains what follows the command STORE in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result accordingly.

PARWR arg(s) a character string
result none

the argument contains what follows the command WRITE in an input line. This function checks the correctness of the option specified (either in the argument or requesting it to the user) and sets the result.

START arg(s) none

result none

this is the main function of the workspace and its behaviour has been described at the beginning of sec. 7.2.

TEKC arg(s) none

result none

this function displays on the screen the message contained in the left argument, starting at the screen coordinates specified in the right argument

TEKG arg(s) a character string

result none

this function displays on the screen a segment connecting each pair of adjacent points, whose coordinates are coded in the argument.

TEKP arg(s) a character string

result none

this function displays on the screen all the points whose coordinates are coded in the argument, without connecting segments.

TEKV arg(s) a character string

result none

this functicn displays on the screen a series cf segments ccnnecting the points whose coordinates are coded in the argument, in the order given.

TKX arg(s) 2-column matrix

result none

each rcw cf the argument contains the screen coordinates of a point. This functicn encodes the coordinates of the pcints in the sequence of characters necessary to drive the terminal in graphic mode.

Complete cross-reference of the workspace IGDS4013

```
FUNCTIONS      CALLED FUNCTIONS        CALLING FUNCTIONS


-------------------------------------------------------------
ALFINP                 EXPL                  GETCOM
                                             PARCH
                                             PARDI
                                             PAREN
                                             PARPL
                                             PARRE
                                             PARSA
                                             PARST
                                             PARWR
-------------------------------------------------------------
APLNAME                OF                    AUTOMSAVE
                                             PARDI
                                             PARRE
                                             PARSA
                                             PARST
                                             PARWR
                                             RESTORE
-------------------------------------------------------------
ASSON                  PROSPAR               COMDIS
-------------------------------------------------------------
AUTOMSAVE              CMS                   COMBUI
                                             COMCHA
                       WRITE                 COMDIS
                       APLNAME               COMRES
                                             COMSTO
-------------------------------------------------------------
CHGRID                 NUMINP                COMBUI
                                             COMCHA
-------------------------------------------------------------
CHECINT                NUMINP                COMBUI
                                             COMCHA
-------------------------------------------------------------
CHTANG                 NUMINP                COMBUI
                                             COMCHA
-------------------------------------------------------------
CMS                                          AUTOMSAVE
                                             COMSAV
                                             COMSTO
                                             RESTORE
                                             STATE
                                             WRT
-------------------------------------------------------------
CMSNAME                   OF                 WRT
-------------------------------------------------------------
COEFFICIENTI                                 CURVILINEO
```

COMBUI	NUMINP CHPOINT CHTANG COONS CHGRID FIT AUTOMSAVE	START
COMCHA	CHPOINT CHTANG CHGRID COONS FIT AUTOMSAVE	START
COMDIS	OF IREAD CONIC ISOM ASSCN ORTHOG AUTOMSAVE DISPLA	START
COMEND	OF	START
COMPLO	PLOTHP	START
COMRES	OF IREAD AUTOMSAVE	START
COMSAV	CMS WRITE OF	START
COMSTO	AUTOMSAVE CMS OF WRITE	START
COMWRI	WRT	START
CONIC		COMDIS
COONS	CURVILINEO	COMBUI COMCHA
CURVILINEO	DISTA NOTC COEFFICIENTI	COONS
DERVAR		RESTORE
DESCR		DISPLA
DFT		WRT

DISPLA	SCALE TEKV TKX TEKG TEKP TEKC DESCR	COMDIS
DISTA		CURVILINEO
EFT		WRT
EXPL	EXPLONG	ALFINP NUMINP
EXPLONG		EXPL
FIT	F0 F1 G0 G1	COMBUI COMCHA
F0		FIT
F1		FIT
GETCOM	ALFINP LUT PARBU PARCH PARRE PARLI PARPL PARST PARSA PARWE PAREN	
G0		FIT
G1		FIT
IREAD		COMDIS COMRES RESTORE
ISCM		COMDIS
LUT		GETCOM PARCH PARDI PAREN PARPL PARST
NCTC		CURVILINEO
NUMINP	EXPL	CHGRID CHPOINT

		CHTANG COMBUI PARDI
OF		APLNAME CMSNAME COMDIS COMEND COMRES COMSAV COMSTO RESTORE
ORTHOG		COMDIS
OUT		TEKG TEKP
PARBU		GETCOM
PARCH	ALFINP LUT	GETCOM
PARDI	ALFINP LUT STATE APLNAME NUMINP	GETCOM
PAREN	ALFINP LUT	GETCOM
PARPL	ALFINP LUT	GETCOM
PARRE	ALFINP STATE APLNAME	GETCOM
PARSA	ALFINP APLNAME	GETCOM
PARST	ALFINP LUT APLNAME	GETCOM
PARWR	ALFINP APLNAME	GETCOM
PLCTG		PLOTHP
PLCTHP	SCALE PLOTG PLCTV PLOTP	COMPLO
PLCTP		PLOTHP
PLCTV		PLOTHP

RECSPAR		ASSON
RESTORE	STATE DEFVAR APLNAME CMS OF IREAD	
SCALE		DISPLA PLOTHP
START	GETCOM COMBUI COMCHA COMDES COMDIS COMPLO COMSTO COMSAV COMWRI COMEND	
STATE	CMS	PARDI PARRE RESTORE
TEKC	TKX	DISPLA
TEKG	OUT	DISPLA
TEKP	OUT	DISPLA
TEKV		DISPLA
TKX		DISPLA TEKC
WRITE		AUTOMSAVE COMSAV COMSTO WRT
WRT	CMS CMSNAME DFT EFT WRITE	COMWRI

Appendix I - principal CP commands

We give here a brief description of the most common CP commands as a quick reference guide. For a complete description of these commands and for additional commands see the CP/67 CMS USER'S GUIDE FORM NO. GH20-0859.

LOGIN <userid>

the virtual machine whose name is <userid> is logged into the system, if the correct password is given, and the CP environment is entered.

LOGOUT

The virtual machine is logged out of the system and any attached device is released. The session is terminated.

IPL CMS

causes a copy of the CMS nucleus to be loaded in the user's virtual memory and to enter the CMS environment.

MSG <userid> <message>

sends the <message> specified in the command to the virtual machine whose name is <userid>. If the specified <userid> is CP, the message is sent to the system operator.

DETACH <devaddr>

detaches the specified device from the virtual machine configuration.

<devaddr> is the virtual address of a device that has been previously attached to the virtual machine.

```
QUERY  ( VIRTUAL )
       ( TIME    )
       ( FILES   )
```

gives some information about the status of the system.

VIRTUAL types at the terminal the configuration of the virtual machine

TIME types at the terminal the connexion time, the virtual time and the total time used in the actual session

FILES types at the terminal the number of the files not yet printed or punched or not yet read

```
BEGIN
```

the system enters the same environment in which was before entering the CP environment.

```
XFER OOD  TO ( <userid> )
             (   OFF    )
```

the output of punch operation is tranferred to the virtual reader of the virtual machine whose name is <userid>. No real deck is punched. If OFF is specified, subsequent punch operations are really performed.

Appendix II - principal CMS commands

We give here a brief description of the most common CMS commands as a quick reference guide. For a complete description of these commands and for additional command see the CP/67 CMS USER'S GUIDE FORM NO. GH20-0859.

```
LISTF  {<filename>  {<filetype>}}
            *            *
```

types at the terminal information about the file(s) specified in the command. The star means all filename and/or all fipetypes.

```
ERASE  {<filename>  {<filetype>}}
            *            *
```

erases the file(s) specified in the command from the P-disk. The star means all filenames and/or all filetypes.

STAT

types at the terminal information about the status of the P-disk, namely the number of the files written, the number of records in use, etc.

```
OFFLINE  {PRINT
         {PUNCH  <filename>  <filetype>}
         {READ
```

controls the input/output devices.

PRINT - the specified file is printed on the line printer

PUNCH - the specified file is punched on the puncher

READ - a deck of cards is read from the virtual card reader and it is stored on the P-disk with the given filename and filetype

PRINTF <filename> <filetype> n1 n2

types at the terminal the contents of the specified file from line n1 to line n2. If n1 and n2 are omitted, the whole file is printed.

ATTN

if the ATTN key is depressed, the CP environment is entered.

Appendix III - mathematical steps

In fig. 4, the connection between two subsequent cubic arcs of a spline curve is shown. The condition of continuity in the point connecting the two arcs is expressed by 5)

$$5)\quad P''_{i-1}(L_{i-1}) = P''_{i}(0)$$

The steps of the computation which lead to the expression 6) are now shown.

$$6)\quad L_i * TG_{i-1} + 2*(L_{i-1} + L_i) * TG_i + L_{i-1} * TG_{i+1} = K_i$$

By deriving twice the 4) with respect to x, it comes out:

$$\frac{d^2\ A0}{dt^2} = (12t - 6L) / L^3$$

$$\frac{d^2\ A1}{dt^2} = (-12t + 6L) / L^3$$

$$\frac{d^2\ B0}{dt^2} = (6t - 4L) / L^2$$

$$\frac{d^2\ B1}{dt^2} = (6t - 2L) / L^2$$

Once substituted 1) derived twice on 5), we obtain:

$$\frac{6L_{i-1}}{L_{i-1}^{3}} x P_{i-1} - \frac{6L_{i-1}}{L_{i-1}^{3}} x P_{i} + \frac{2L_{i-1}}{L_{i-1}^{2}} x TG_{i-1} + \frac{4L_{i-1}}{L_{i-1}^{2}} x TG_{i} =$$

$$= \frac{-6L_i}{L_i^3} x P_i + \frac{6L_i}{L_i^3} x P_{i+1} - \frac{4L_i}{L_i^2} x TG_i - \frac{2L_i}{L_i^2} x TG_{i+1}$$

and then:

$$\frac{1}{L_{i-1}} x TG_{i-1} + \frac{2 x (L_i + L_{i-1})}{L_{i-1} x L_i} x TG_i + \frac{1}{L_i} x TG_{i+1} =$$

$$= \frac{3}{L_i^2} x (P_{i+1} - P_i) + \frac{3}{L_{i-1}^2} x (P_i - P_{i-1})$$

Finally we obtain:

$$L_i x TG_{i-1} + 2 x (L_i + L_{i-1}) x TG_i + L_{i-1} x TG_{i+1} =$$

$$= \frac{3 L_{i-1}^2 (P_{i+1} - P_i) + L_i^2 (P_i - P_{i-1})}{L_{i-1} x L_i}$$

where the known term is an expression depending on

P_i , P_{i-1} , P_{i+1} .