Comput. & Graphics Vol. 12, Nos. 3/4, pp. 441–456, 1988
Printed in Great Britain.

0097-8493/88 $3.00 + .00

MULTI-PURPOSE COMPUTER GRAPHICS SYSTEM SMOG-85

A. M. MATSOKIN, V. A. DEBELOV, V. G. SIROTIN and S. A. UPOL'NIKOV
Computing Center, Siberian Division, the USSR Academy of Sciences, Novosibirsk, USSR

Abstract—This paper describes the SMOG-85 system, a set of software components for graphical and geometric data processing. One may distinguish among the parts of the system functional components implementing main capabilities of the system and technological components providing a local operating environment for functional components. A general structure of the system and interfaces between separate components, or processors, are described. Also described are structures and main capabilities of functional components—an illustration processor, a drafting processor, a 3D modelling processor—and main parameters of technological components—a graphical output processor, an interaction processor and a data base processor. The system could be applied to preparation of documents containing graphic illustrations; construction of geometric 2D and 3D objects, their visual analysis and computation of mass properties; and preparation and editing of draft documentation.

INTRODUCTION

The SMOG-85 system[1] was developed at the Computing Center of the Siberian Division of the USSR Academy of Sciences. Work connected with the creation of this system started in 1970. The first stage developed was the illustration processor, intended for the presentation of the results of scientific research (1971). During this period, general principles were formulated whose further development became a conceptual basis of SMOG-85. Three basic principles can be distinguished:

(1) independence of the system functionality from the operating environment where it functions;

(2) independence of functional capabilities upon particular application fields;

(3) separation of graphical and geometric data from algorithms and programs which process them.

Naturally, these principles are rather general and should be considered in more detail.

A set of graphic device drivers, available through their interfaces and data base system, serves as an operating environment. According to the first principle, we distinguish technological components. In fact, technological components "virtualize" graphical devices and unify the access to the external storage. At present, the idea to virtualize graphic devices has become conventional in computer graphics. It is particularly referred to in various standards[2–4]. The selection of technological components made it possible to essentially improve the performance of SMOG-85 and to adapt it to different hardware installations. For example, the illustration processor with graphic data output processor is, in fact, used as a graphic standard in many enterprises.

SMOG-85 was developed as multi-purpose system, i.e. if possible, independent of applications system intended for graphical and geometric data processing on a computer. SMOG-85 was developed in accordance with the module-hierarchical principle. The system is implemented as a subroutine library. This approach essentially simplifies both the interface between separate modules and the interaction with application systems. Functionality of each module is available by means of subroutine CALLs in FORTRAN. It is difficult to process geometric and graphical data on a computer since its representation in computer memory is not adequate due to the absence of a formal model of a geometric object and a display image fit for different applications. Data representation structures used in the standards GKS and IGES provide support for this view. Thus, in the standard GKS[3], simple linear structures over graphical primitives (which are adequate to capabilities of existing graphical devices rather than to the image) are used for graphical data representation. On the other hand, IGES data structures[5] are an eclectic combination of conventional techniques for different mathematical models of geometric data in the form of the complicated hierarchical list structures. The absence of a universal geometric model accounts for the fact that programs for graphical and geometric data processing, as a rule, are too complicated and are directed to a particular application. In the development of SMOG-85, separation of graphical and geometric data from programs of their processing was of primary importance. Therefore, further development of geometric models and methods of their processing was required.

We will describe below a general structure and basic data models used in SMOG-85 and separately, its main components. In the description of some components, i.e. processors, of primary attention are functional capabilities.

1. GENERAL STRUCTURE AND DATA MODELS

The general structure of SMOG-85 is shown in Fig. 1. Interaction between components is realized as data streams. Three basic data models are distinguished in the system: graphical segments and metafiles, draft models and 3D models. For representation of these models, hierarchical list structures are used. The storage and processing of these structures are provided by the

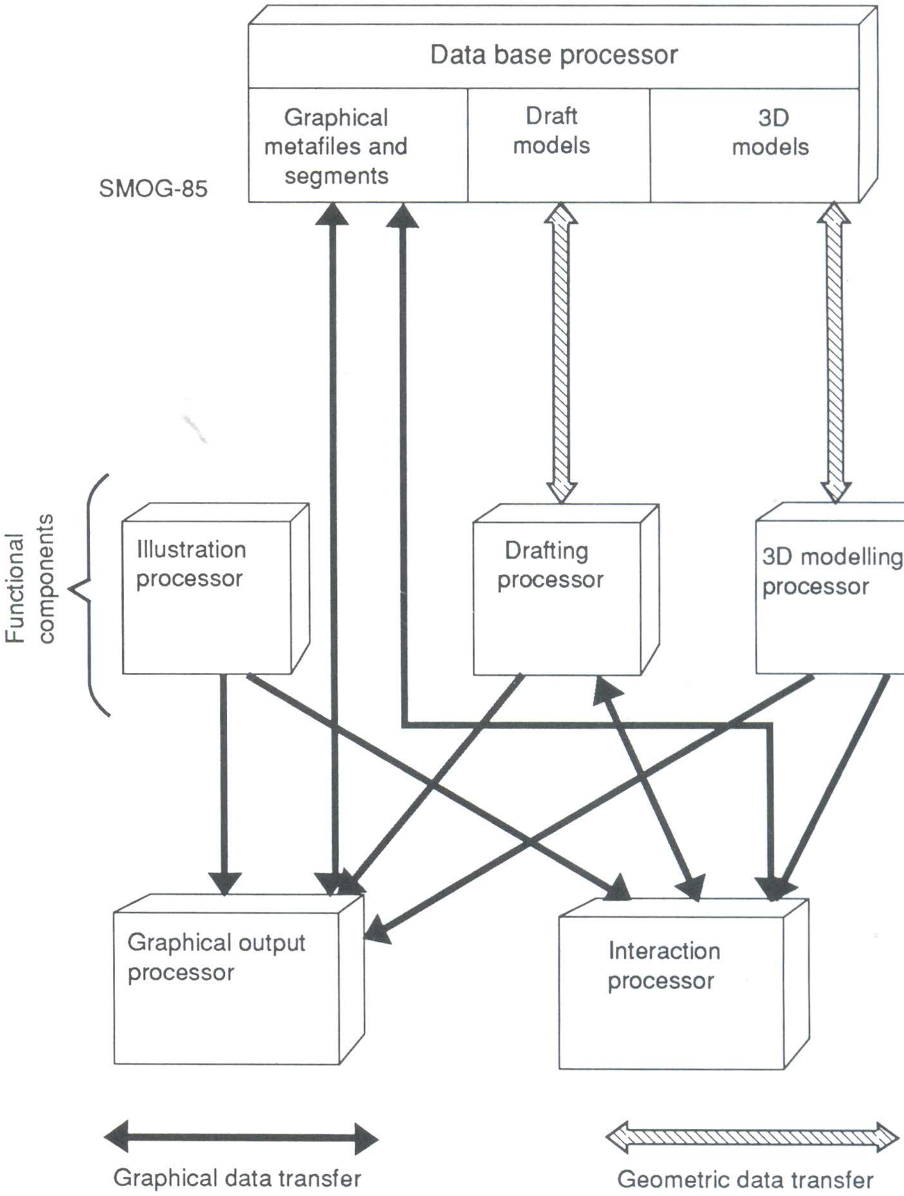

Fig. 1. Structure of SMOG-85 (scheme).

data base processor. The construction and interpretation of representations are implemented by special processors.

2. ILLUSTRATION PROCESSOR

The illustration processor allows the construction of images:

—plots of functions of the form $y = f(x)$, or $y = y(t)$, $x = x(t)$;
—plots of functions of the form $z = f(x, y)$, or $z = z(u, v)$, $y = y(u, v)$, $x = x(u, v)$ (surfaces);
—isolines;
—maps of velocity vectors;
—2D and 3D histograms;
—3D scenes, which are a combination of wireframe models and collection of shields (Figs. A1–A6, A11).

The source of data for programs implementing these functions can be given analytically either discretely in the form of vectors (for plots) or in the form of height matrices (for surfaces). The height matrix allows one to define values on grids with a uniform or a non-uniform mesh size.

As is seen, the above-described capabilities allow one to construct graphical interpretations of objects of conventional mathematical theories. The illustration processor also allows preparation of text documents containing different formulas and notation. As it takes place, a variety of fonts for digits and signs of the Russian, Latin and some other alphabets can be used.

The illustration processor was the first part to appear; it has gained wide recognition in many different scientific and design systems.

The illustration processor generates a stream of graphical data which can be visualized either with the help of the graphical output processor or the graphic interaction processor.

Thorough development of corresponding mathematical algorithms has resulted in attractive looking processor-generated plots, isolines, etc.

3. 3D MODELLING PROCESSOR

The 3D modelling processor performs the following activities:

—forming of geometric of 3D objects;
—calculation of main mass properties by models;
—construction of flat images of 3D objects.

The general structure of the processor is shown in Fig. 2.

The following basic geometric objects are distinguished: plane curves, polygons, polyhedrons and spatial curves and surfaces. To describe objects, linear or parametric representations are used. Both types of representation are based on the boundary technique.

The following techniques are used for constructing the models:

—description of basic geometric shapes (conic sections, parallelepipeds, cylinders, cones, etc.);
—parametric representation of the Coons curves and surfaces with possible linear and cubic interpolation;
—design of polyhedral models by means of the Boolean operations: unions, intersections, subtractions;
—construction of surfaces by displacement or rotation of plane curves (sweeping operations).

Peculiar to boundary representation of models is the absence in their structure of data on boundary elements connections, which has allowed essential simplification of the Boolean operations implementation on polyhedra[7]. The data on boundary elements orientation make calculation of mass properties possible.

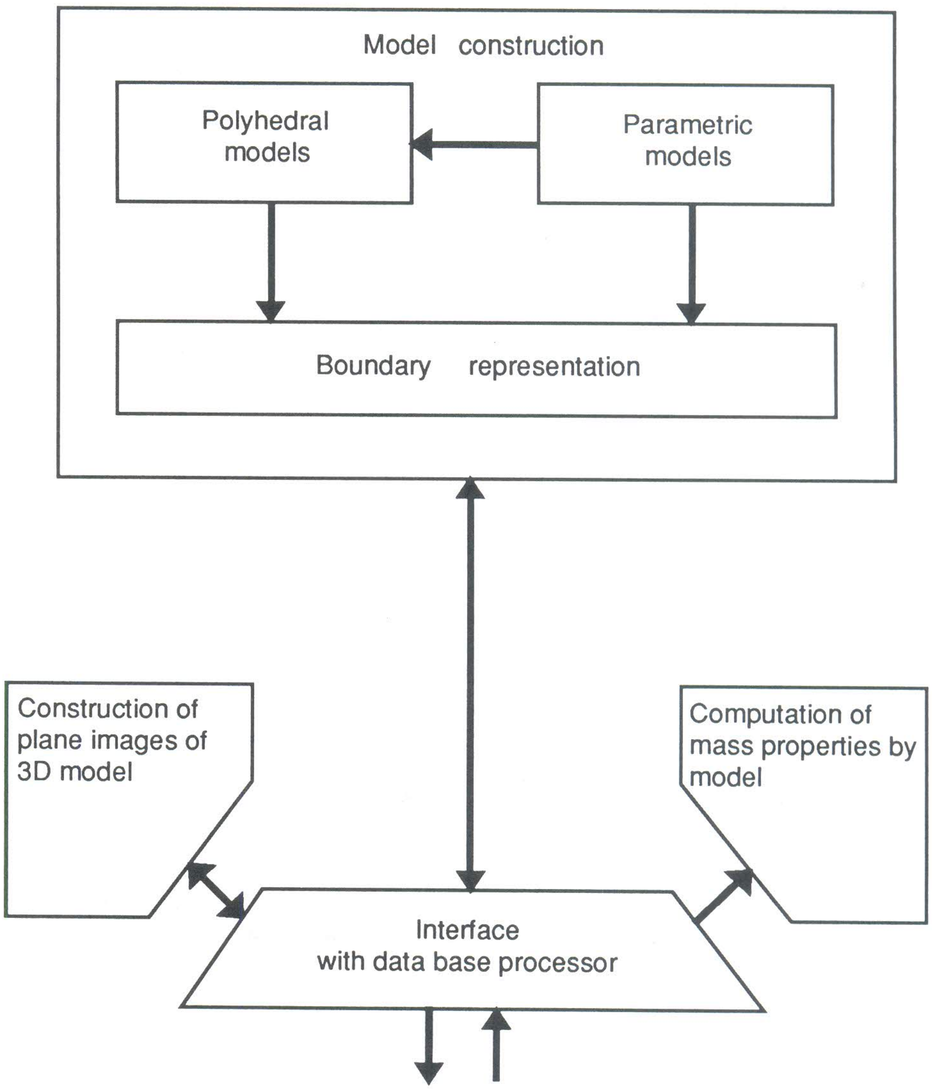

Fig. 2. Functional scheme of the 3D modelling processor.

The processor provides calculation of the following properties: dimension, area, volume, center of gravity and moments of inertia.

Flat image construction is based on algorithms of the Boolean operations on polygons. The processor allows one to construct various sections and arbitrary axonometric projections of 3D objects with hidden parts of sides removed. Flat images of 3D models can be visualized with the help of the graphical output processor. The use of the polyhedral representation of 2D and 3D objects essentially simplifies interfaces with both raster and vector graphical devices.

3D models and flat images are generated and processed by using features of the data base processor[6]. Both are typically used for modelling mechanical parts obtained with the help of extruding, machining, drilling and objects with smooth external shapes (car bodies, turbine casing, aeroplane fuselage, etc.).

Figures A7 to A10 show examples of using the processor of 3D modelling.

4. DRAFTING PROCESSOR

The drafting processor is intended for the following procedures:

—generation and editing of draft models;
—calculating some geometric properties of 2D objects shown in a draft (distances between objects, centers of gravity, areas, etc.);
—obtaining the data from a draft model, necessary for the work of different post-processors (program generators for devices with NC, programs for different calculations, etc.);
—assembling the complicated drafts from views stored in the data base;
—drawing the drafts on paper in accordance with the conventional standards.

The drafting processor structure is presented in Fig. 3. Figure A12 represents results of the drafting processor work.

In correspondence with the USSR standards on documents, the drafting processor is based on a three-level hierarchical model:

1st level: document,

2nd level: list,

3d level: view in the draft[8].

The view represents a set of geometric elements

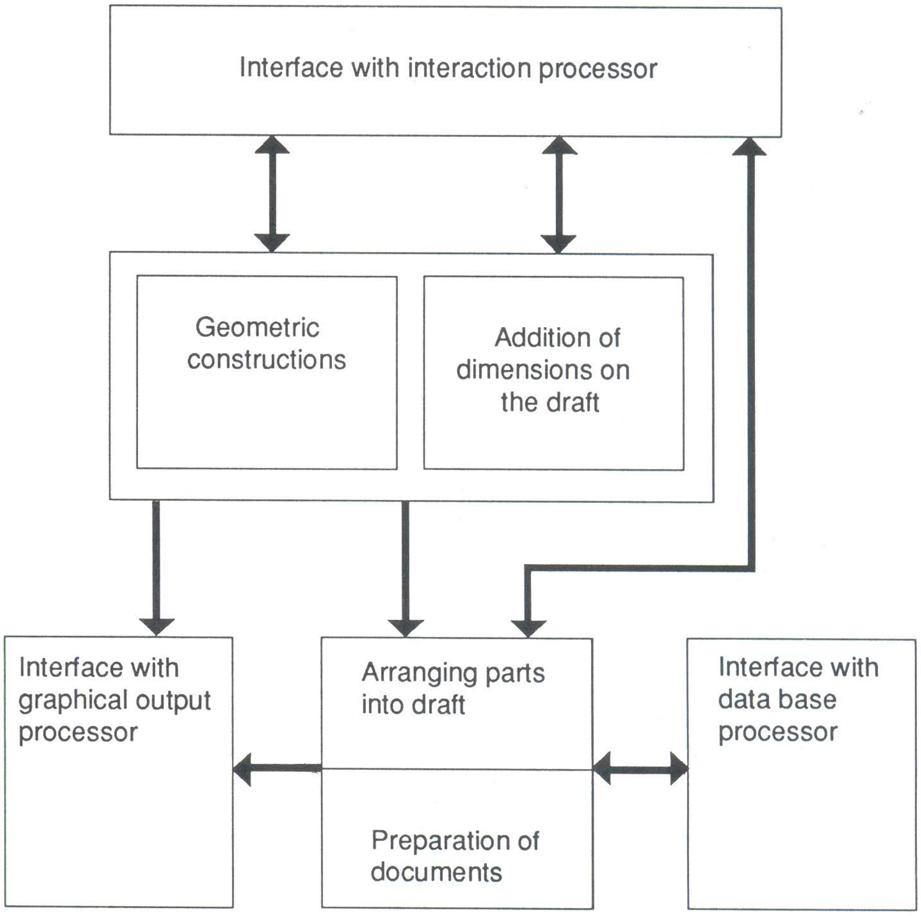

Fig. 3. Drafting processor structure.

(points, segments, arcs, etc.) and technological elements (dimensions, special symbols, etc.) with imposed "tangent," "parallel," etc. connections. Algebraic systems of a special form[9] are used as a theoretical basis for modelling of views.

5. GRAPHICAL OUTPUT PROCESSOR

The graphical output generator is intended for the following functions:

—translation of image descriptions from a virtual graphical device language (a metafile language) into a language of a particular device;
—accumulation of sets of graphical metafiles to provide an optimal operation of graphical output devices;
—accumulation of statistics on utilization of graphical resources by a user.

The structure of the graphical output processor is shown in Fig. 4.

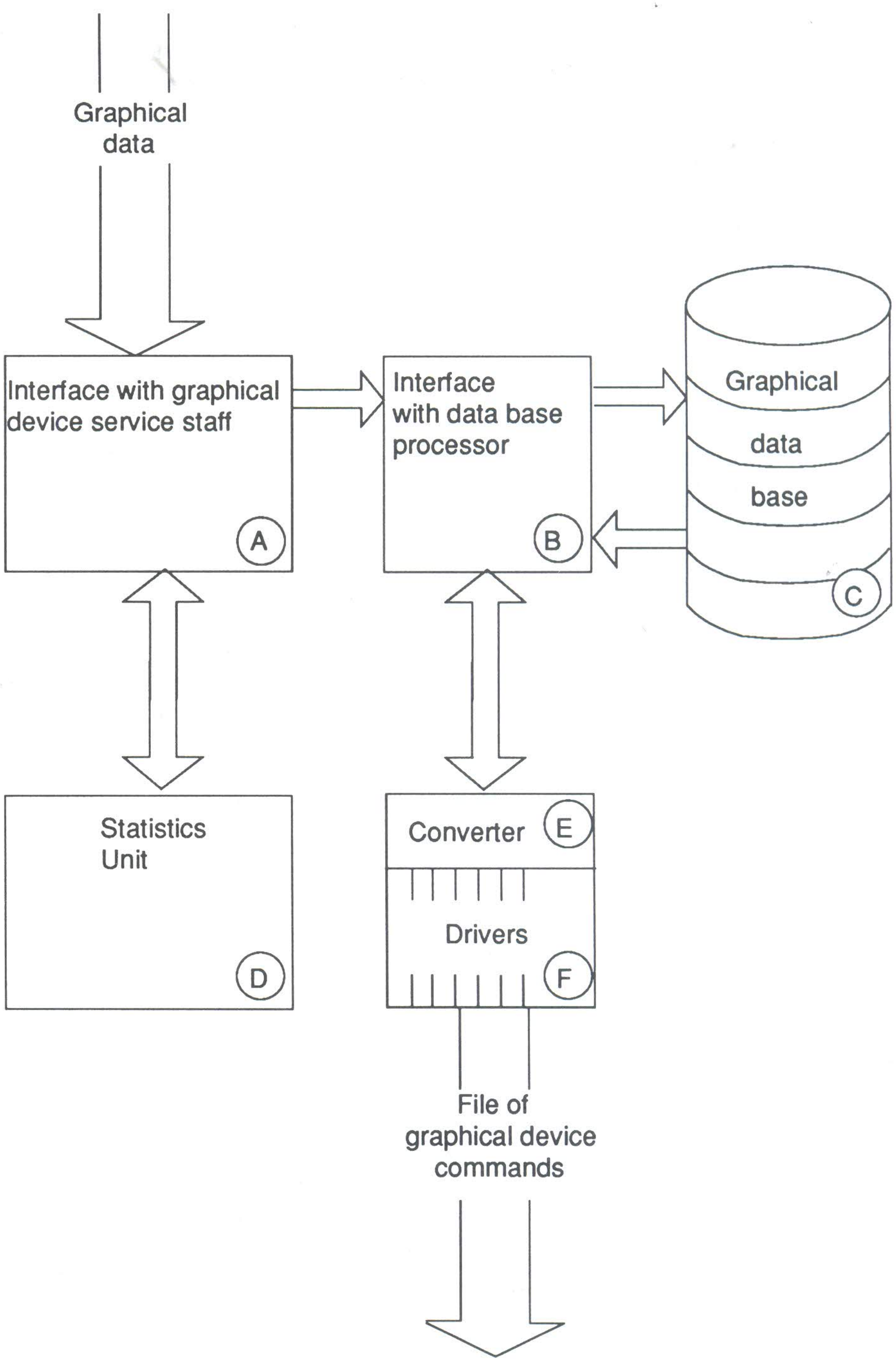

Fig. 4. Structure of graphical output processor.

Depending on conditions, different modifications of the processor can be used. The maximal configuration is shown in Fig. 4. It is intended for a large computer center or for a terminal workstation in a computer network and is oriented to an intensive stream of graphical output. In the "personal" version, when a graphical device is connected on-line to a computer, components C and D are eliminated from the system, and components A and B become transparent for the graphical data.

Component A is the main component of the processor. It supports the work of the operators of graphical devices, namely:

—assembles the completed graphical metafiles from separate graphical segments coming to the processors in real time;
—transmits the data on graphic metafile authorship and its volume (in meters of run of a pen and pages) to operators and to a statistics unit;
—controls the transmission of metafile sets intended for visualization on a particular device (raster or vector);
—gives in interactive mode necessary data on the state of the data base, etc.

In some cases it seems appropriate to look at a display image (stored as metafile in the data base) on a display before it is plotted. This can be done with the help of the processor component.

A set of drivers (component E) is determined by a set of devices, a driver for each type of device.

The statistics stored while using the processor makes it possible to produce necessary financial calculations with users and to come to justified solutions about the increase/decrease and quantity/quality of graphic devices.

The statistics stored at the Computing Center of the Siberian Division of the USSR Academy of Sciences in Novosibirsk indicates that during the last 10 years of using SMOG-85 more than 840 thousand of graphical pages were issued.

6. INTERACTION PROCESSOR

The processor provides:

—an interface between different-type computers;

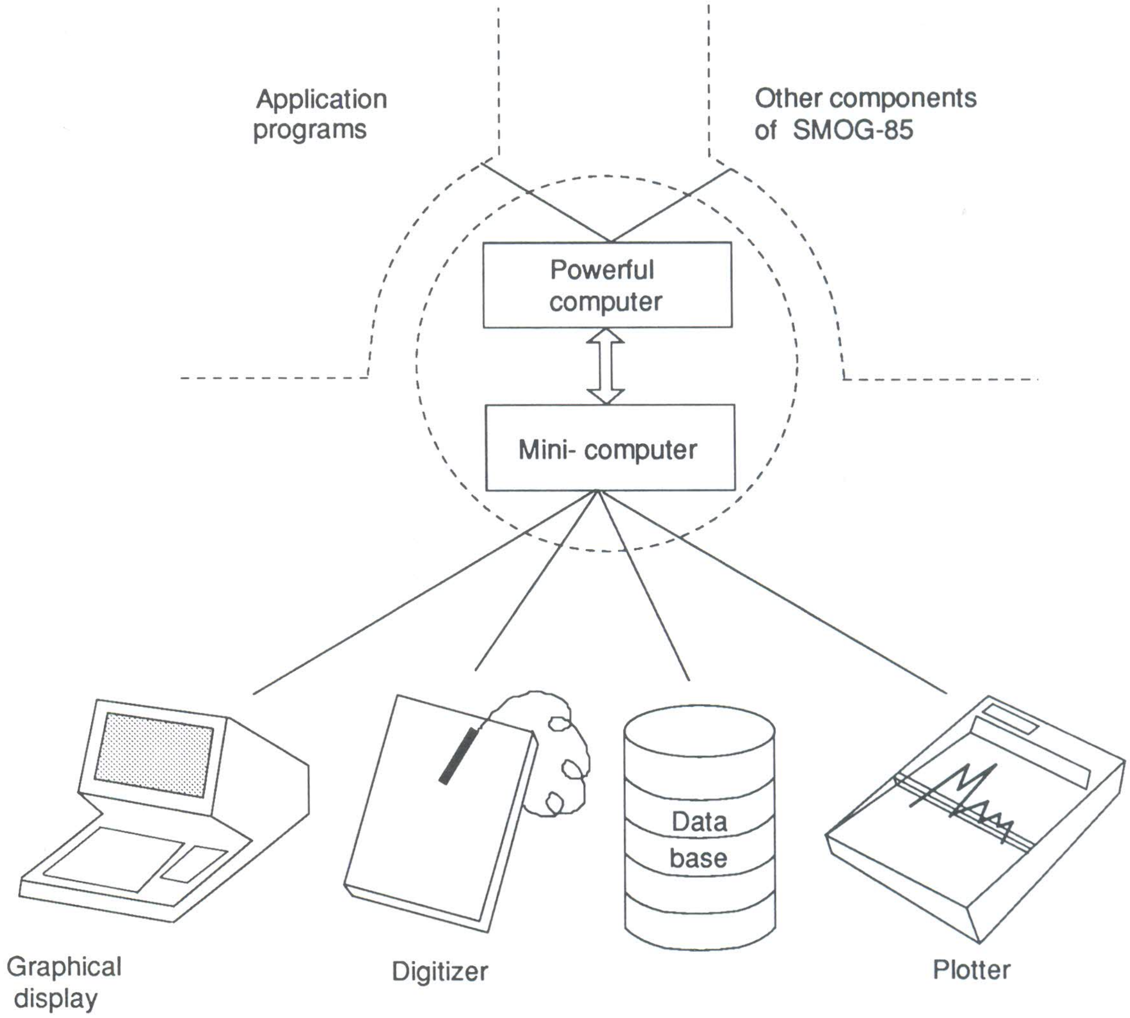

Fig. 5. Structure of interaction processor.

—the fulfilment of functions distributed between two varied computers.

For a computer close to a user we select a minicomputer as a part of a workstation and a powerful computer as another. The host computer may be a part of the computer network.

The processor is designed so software can be easily increased and modified (also by the user) on any of the two different-type computers. The corresponding part of the processor may also be used separately on either computer.

The structure of the processor is shown in Fig. 5.

7. DATA BASE PROCESSOR

This processor provides generation and processing of data structures, representing:

—3D models;
—draft models;
—device-independent graphical segments;
—graphical metafiles.

It has been specially developed to meet the requirements from other processors of SMOG-85.

Practically, it is a means of working with dynamic byte arrays distributed in main and external memories of a computer. The processor provides rapid access to the elements of such arrays. This is required by many algorithms of a 3D modelling processor and a drafting processor.

In addition, the processor provides the function for creation of hierarchic data structures and processes them. Main concepts of data structures are object and bush. Bush is a structure defined by a selected object (the root of the bush) and consists of all the objects which may be achieved from the root, using the references "up" only. The following operations are defined with objects of these types: creation of an object, generation and destruction of connections, successive access to objects of a particular bush, access to leaves of a bush, etc.

At present, SMOG-85, as a whole or in its components, has been installed in more than one hundred enterprises in the USSR. It has also been installed in the Computing Center of the Bulgarian Academy of Sciences and in INRIA (France). The supplier of the SMOG-85 system is NPO "Centrprogramsistem" in the city of Kalinin.

Development of the system is continuing.

REFERENCES

1. V. A. Debelov and A. M. Matsokin, Computer graphics system SMOG-85. Preprint No. 504, Novosibirsk (1985).
2. ACM-SIGGRAPH, GSPC'79. Status report of the graphics standards planning committee. *Comp. Graphics* **13** (1979).
3. ISO, Information processing—Graphical Kernel System. GKS—Functional Description. ISODIS 7942 (Nov. 1982).
4. ISO TC 97/SC21/WG2 No 356. Information processing.—Computer Graphics.—Interfacing techniques for dialogues with graphical devices (CGI)—Second Initial Draft (Nov. 1985), pt. 1–6.
5. ANSI Y14. 26M-81. Digital representation for communication of product definition data—USA.
6. A. M. Matsokin and S. A. Upol'nikov, Package SPACE as means of construction and processing of 3D objects. (Programmnye sredstva mashinnoi grafiki.) Novosibirsk (1984), S. 77–90.
7. S. A. Upol'nikov, Implementation of polyhedra (Programmnye sredstva mashinnoi grafiki.) Novosibirsk (1984), S. 91–103.
8. V. M. Golubev and V. G. Sirotin, Software for generation of data bank of engineering documents (Programmnye sredstva mashinnoi grafiki.) Novosibirsk (1984), S. 52–58.
9. V. G. Sirotin, Software for draft compilation (Programmnye sredstva mashinnoi grafiki.) Novosibirsk (1984), S. 59–70.

APPENDIX 1

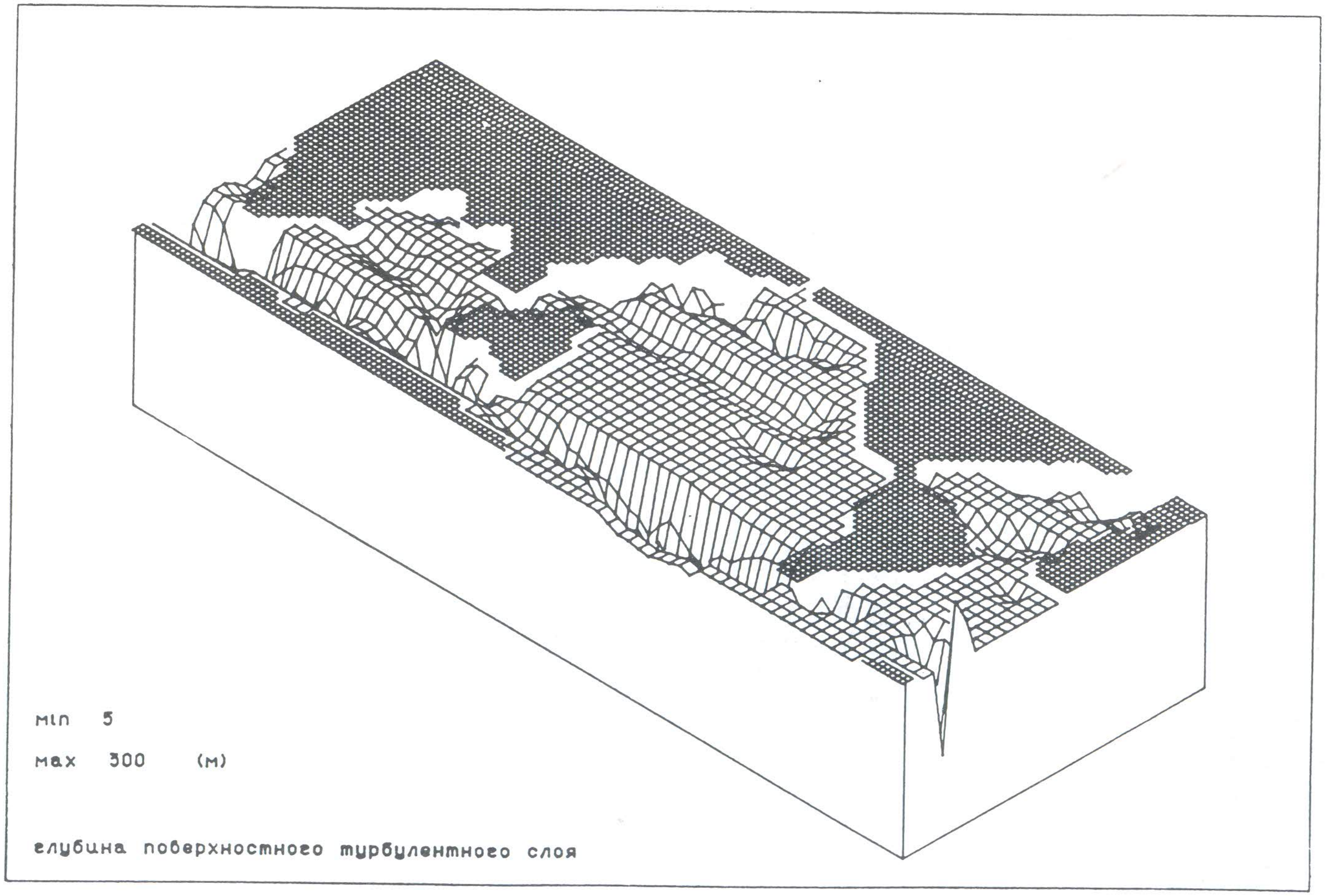

Fig. A1.

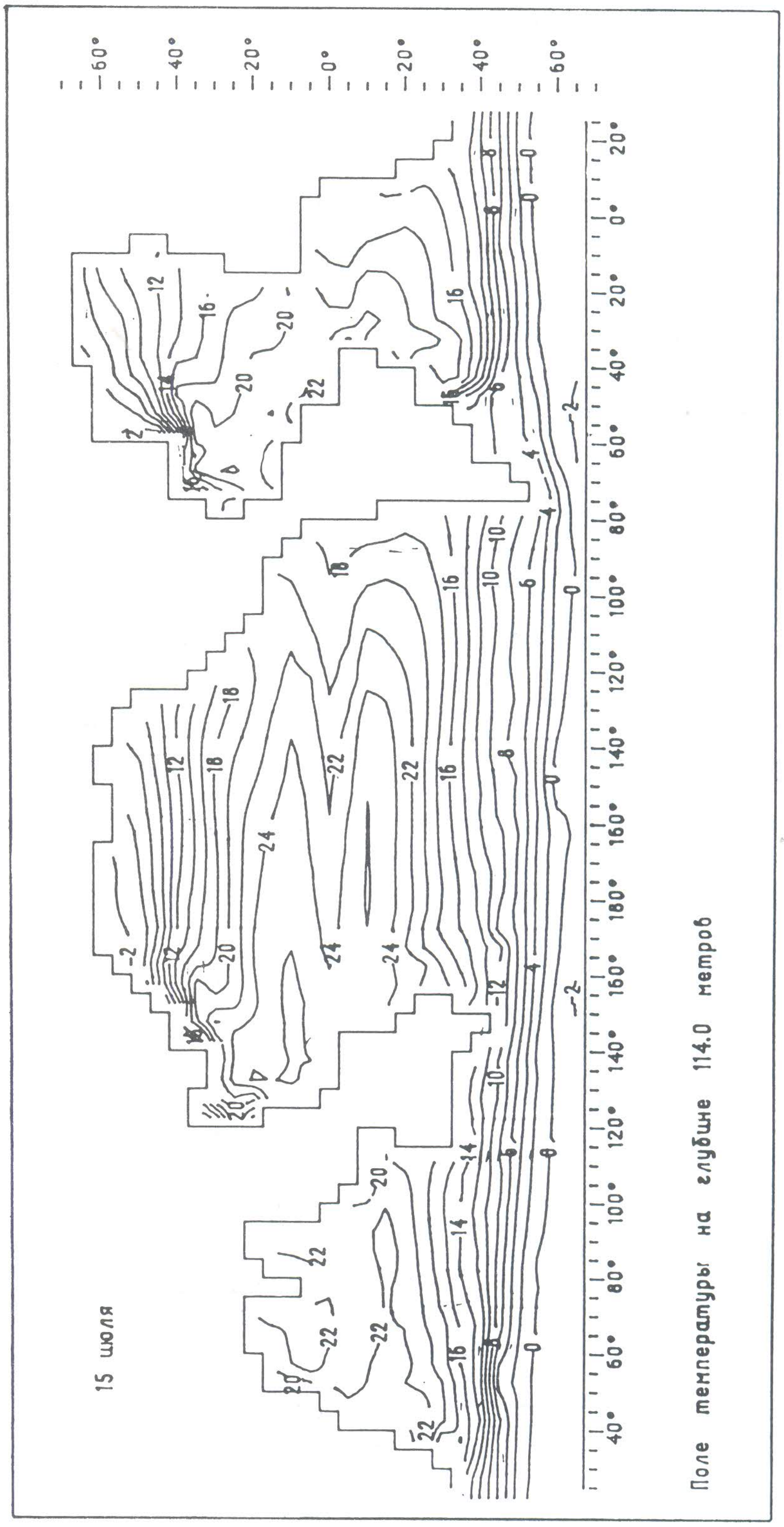

Fig. A2.

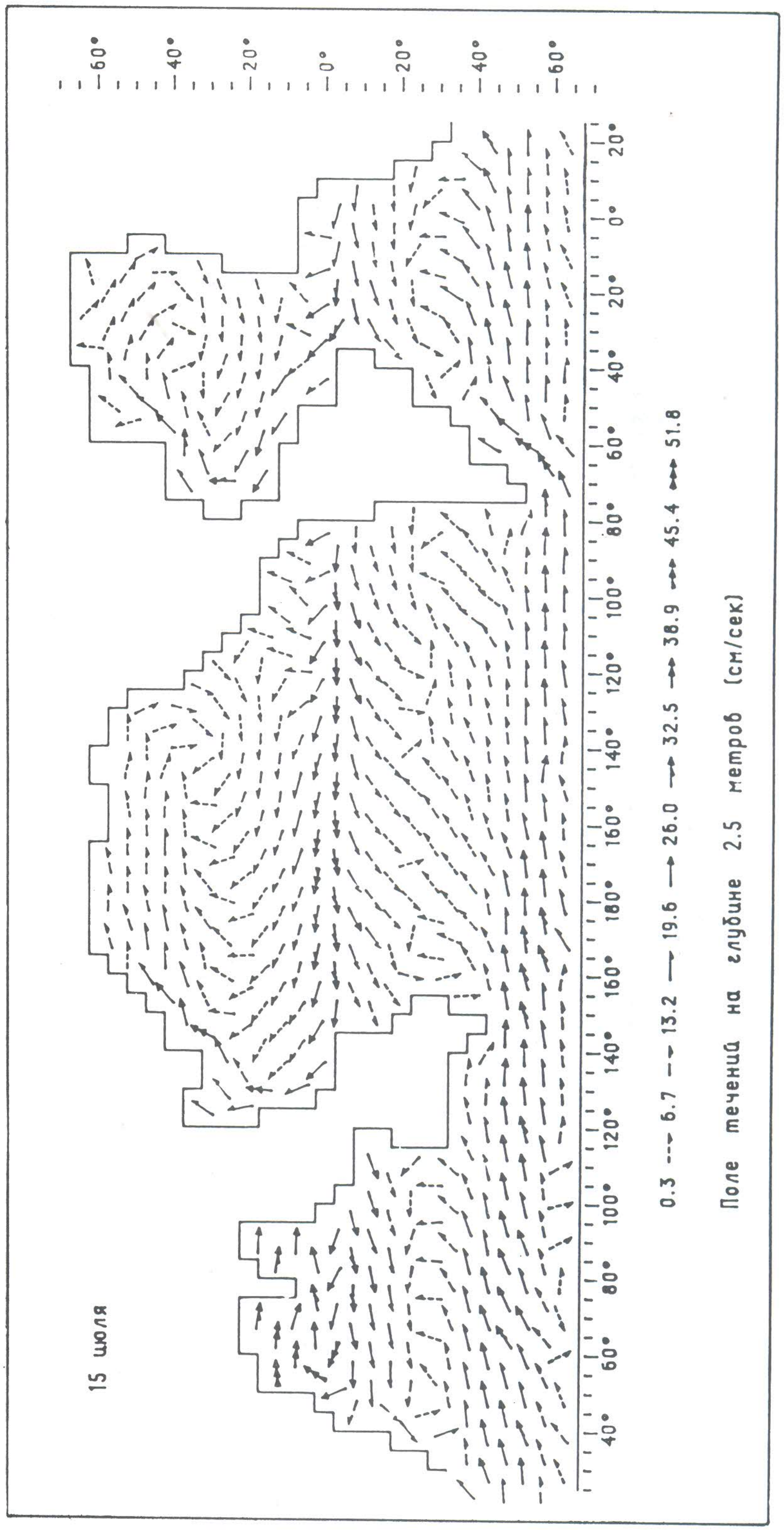
15 июля
0.3 6.7 13.2 19.6 26.0 32.5 38.9 45.4 51.8
Поле течений на глубине 2.5 метров (см/сек)

Fig. A3.

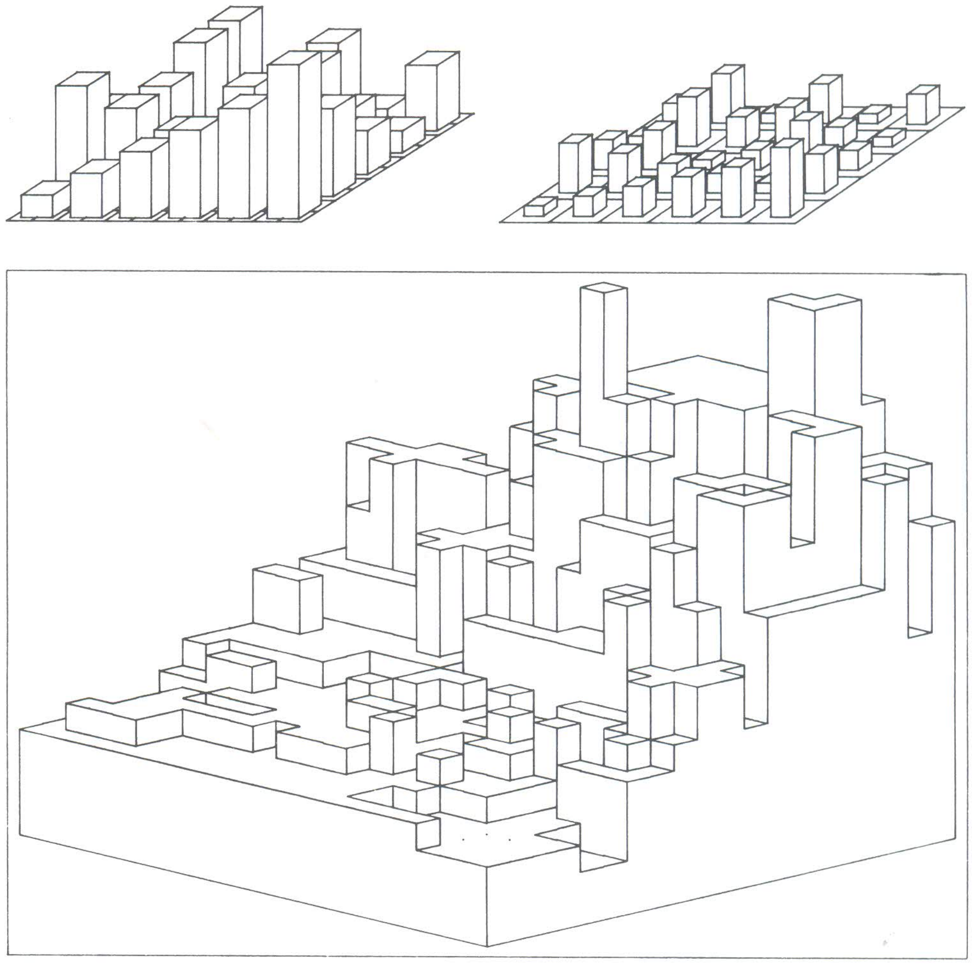

Fig. A4.

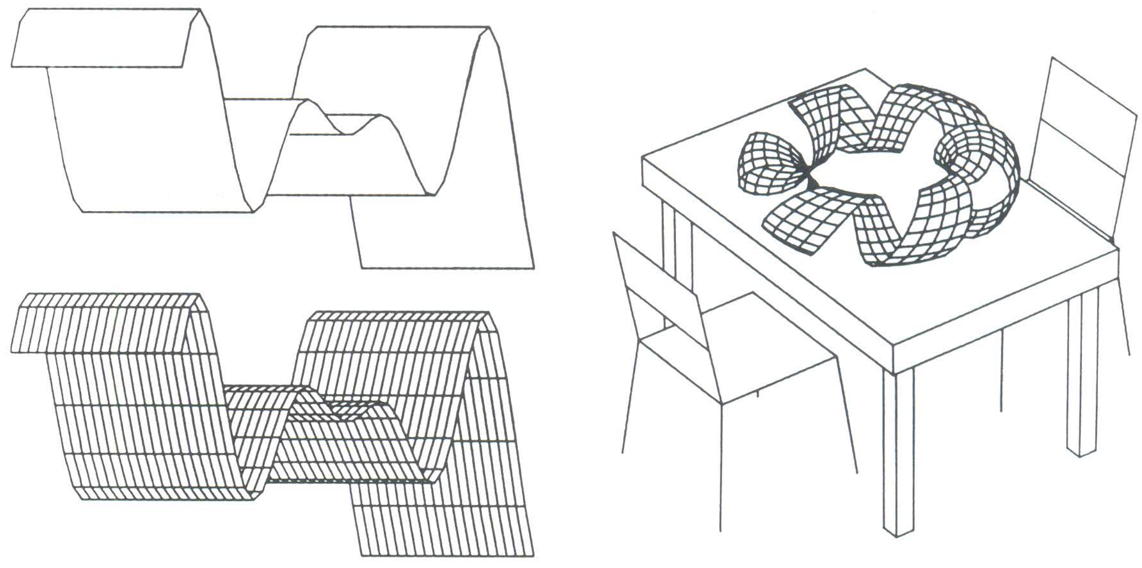

Fig. A5.

Разложение в ряд Тейлора:

$$f(z)=\sum_{n=0}^{\infty} a_n(z-a)^n,$$

$$a_n=\frac{1}{n!}f^{(n)}(a)=\frac{1}{2\pi i}\int_S \frac{f(\zeta)}{(\zeta-a)^{n+1}}d\zeta .$$

Fig. A6.

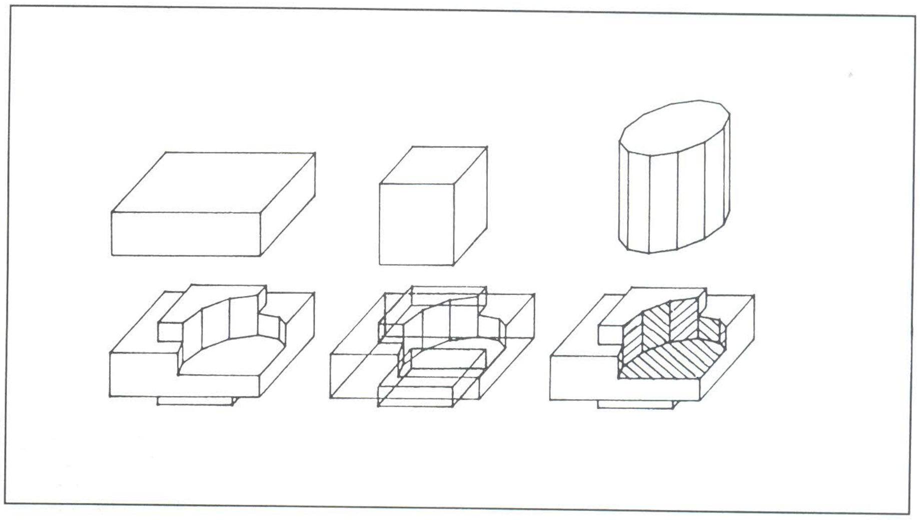

Fig. A7.

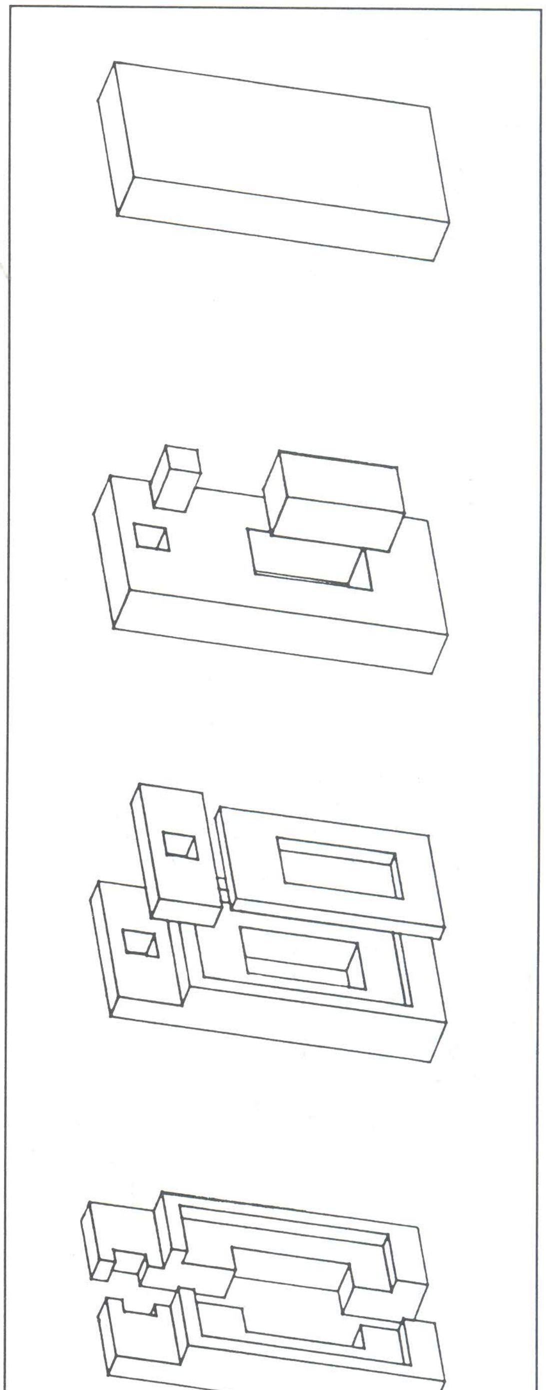

Fig. A8.

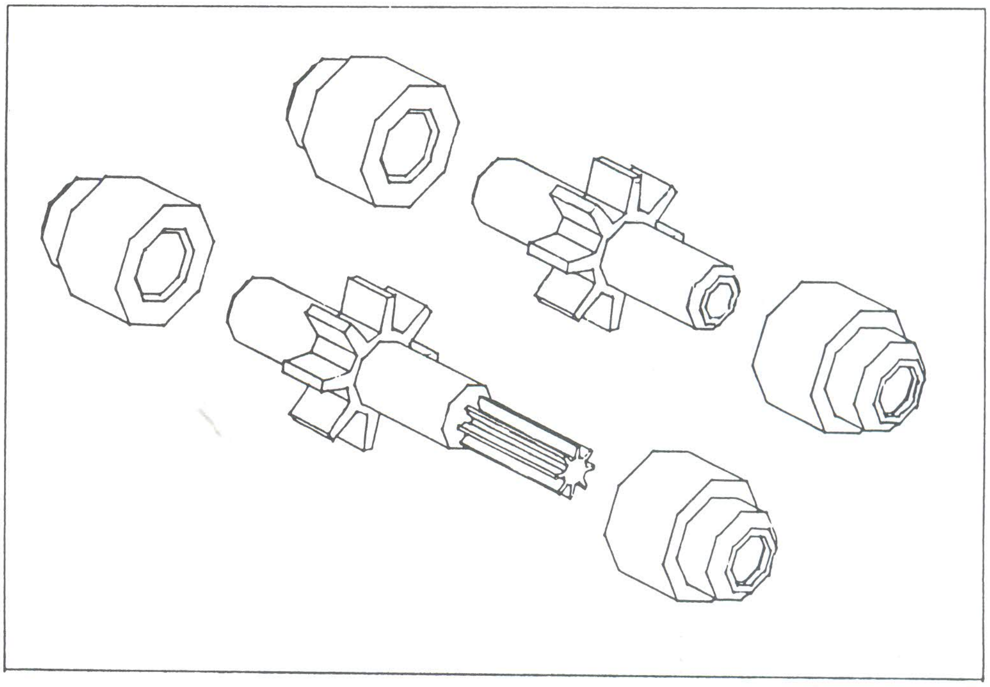

Fig. A9.

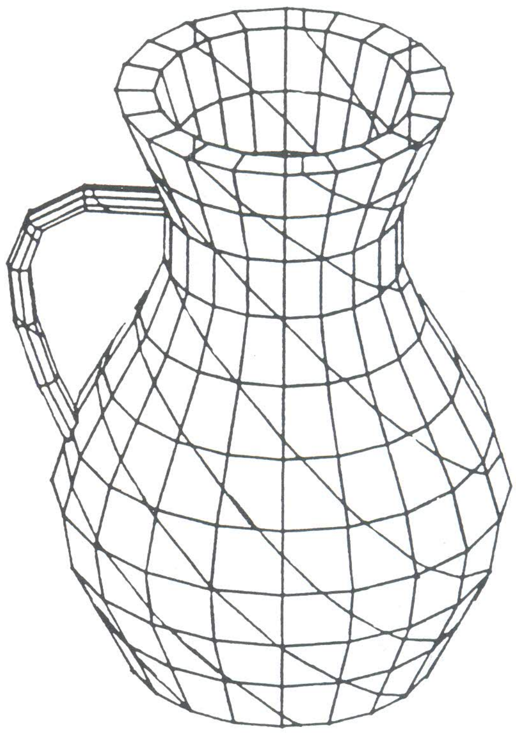

Fig. A10.

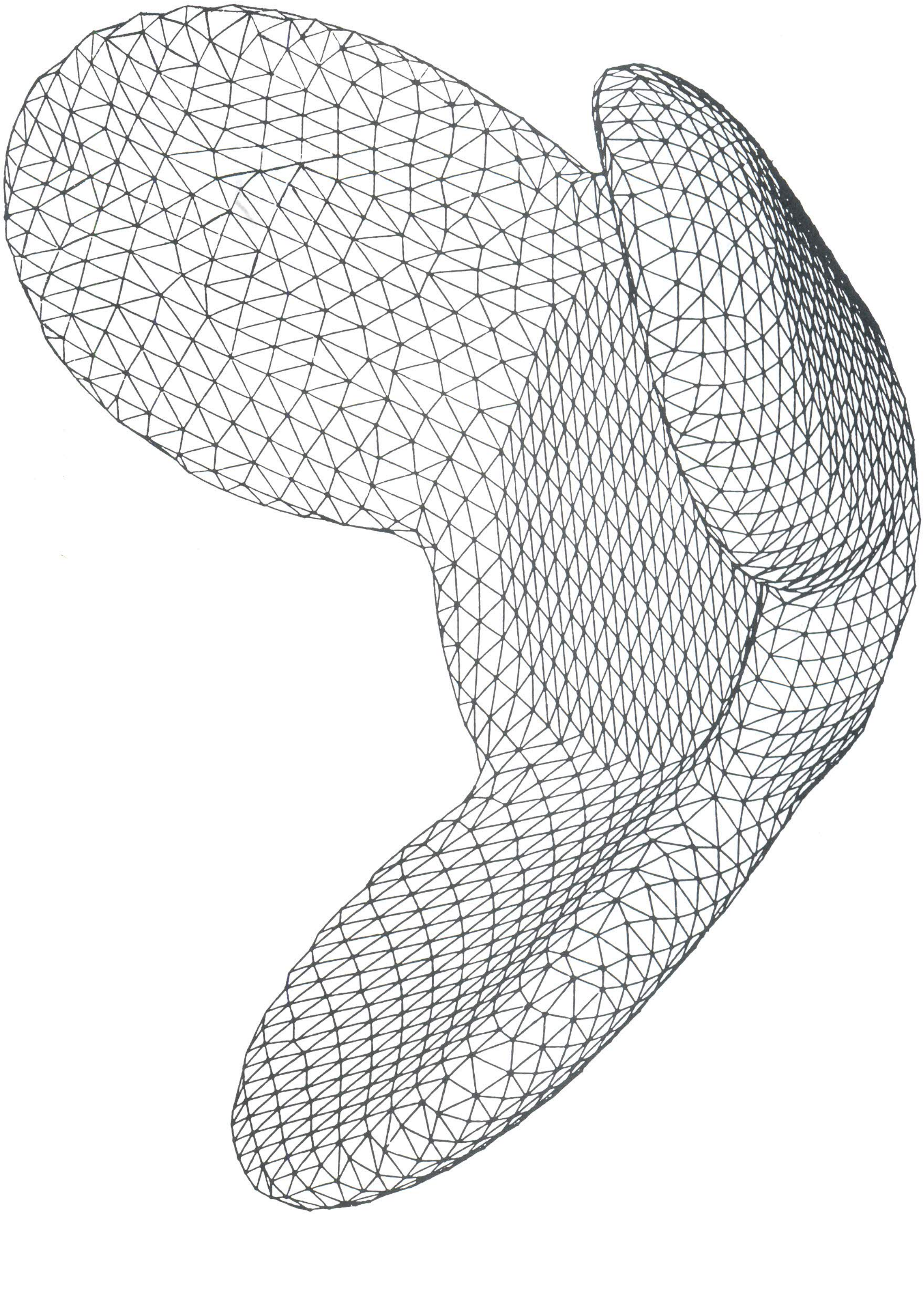

Fig. A11.

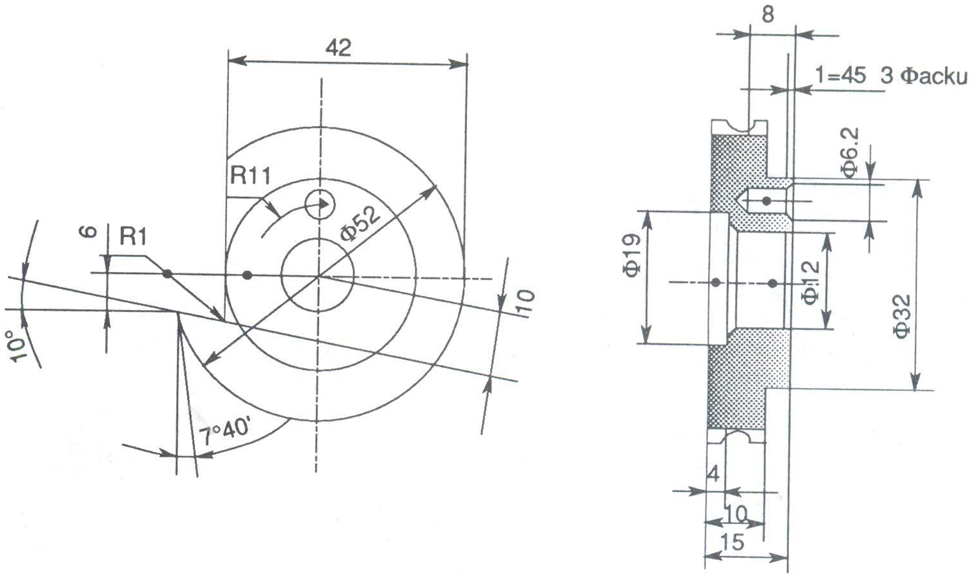

Fig. A12.