

# G-PLOT: GRAPHICS IN THE GEOLOGICAL COMPUTER<sup>1</sup>

S. HENLEY

Institute of Geological Sciences, Keyworth, Nottingham

C. J. FAGE

Institute of Geological Sciences, London

and

K. G. JEFFERY and E. M. GILL

Natural Environment Research Council Central Computing Group, Rutherford Laboratory, Chilton, Didcot, Oxfordshire, U.K.

(Received 14 February 1977; revised 27 June 1977)

**Abstract**—G-PLOT is a package of programs for graphic display of data. It has been designed as an integral part of the G-EXEC data-handling system, and adopts the principles of G-EXEC: it is generalized with respect to data, it is structured and modular, and it has a simple (near-English) user interface. In addition, a number of standards have been defined to allow the greatest possible flexibility of usage of the programs, as well as to provide a consistent and powerful set of modules for the applications programmer. The two most significant of these standards are the use of the 'plot-frame' concept and the definition of logical raster levels between the data and a 'logical plotter'. By the use of these standards, it is possible not only to send generated plots to any available graphic device, but also to edit plots by selection and manipulation of logical 'subframes'—units of plot data such as axes, titling, map frames, and grids. By the combination of G-PLOT facilities with planned enhancements to the G-EXEC system, it will be possible to generate sequences of plot frames from iterative simulation models, or from successively changing parameters (scaling or rotation for example) in normal application programs.

**Key Words:** Graphics. Plotting. I/O. Program system. Soft machine. Packets.

## INTRODUCTION

In geology, more than in most sciences, graphic presentation of data is of great importance. Not only the spatial relationships (in two or three dimensions) are required, but also graphs of one variable against another or against a time axis.

In recent years there has been significant growth of geological data bases and data-handling systems, and the effective production of graphic output from data within such a system requires a convenient interface between the graphic applications programs and the data-base handling programs.

The G-EXEC system (Jeffery and Gill, 1976a; Cubitt, 1976) consists of an integrated collection of programs for data management and analysis, and provides such an interface. Moreover, it allows the generation of graphics from any data file through the generalized design of the system.

The data-base handling system, however, comprises only one of a number of distinct 'soft machines' which may be identified (Jeffery and Gill, 1976b; Henley and others, 1977). Another soft machine may be defined for interactive editing of character-form files (as contrasted with binary files in the data base); a third, not yet completed, has some of the features of each of the previously mentioned, for interactive control of simula-

tion models and for interactive graphic operations—both of which purposes require a combination of fast processing (perhaps including the use of parallel processor technology) with flexible machine/user interaction through a range of devices.

Thus in this wider context, there are a number of inter-communicating soft machines, of which G-EXEC is an implementation of one. In such an environment, a system for geological graphics will cross soft-machine boundaries, with processing (*sensu stricto*) carried out by one, and a character-form file transmitted in a standard communication file structure to another soft machine for local editing and plotting.

This paper is concerned principally with the design of the G-PLOT graphics package within the G-EXEC data-base handling soft machine, but also describes the complementary graphics data-handling facilities of other soft machines.

## G-EXEC DESIGN PRINCIPLES

The G-EXEC system has been developed in the context of a consistent design philosophy (Jeffery and Gill, 1976a). This philosophy is based on the need for a data-handling system for the environmental scientist, which provides both a simple interface to the user and a well-structured programmer interface. The design principles are discussed more fully in the various papers by Jeffery, Gill, and Cubitt, and are only summarized here.

Desirable features of a system designed to meet the criteria include:

<sup>1</sup>Published by permission of the Director, Institute of Geological Sciences, and the Secretary, Natural Environment Research Council.



(1) Generality. The ability to process data files with any data content: independence of data content is achieved by the use of a 'self-describing' file structure.

(2) Integration. Any processes may be carried out on any data files in any sequence defined by the user.

(3) Modularity. The principles of structured programming are followed to achieve a system which is easy both to maintain and to extend.

(4) Portability. The system is written in standard ANSI FORTRAN and may be implemented with little modification on most modern computers.

#### THE SOFT-MACHINE CONCEPT

The G-PLOT package has been designed in the soft-machine framework, of which G-EXEC is but one component. A soft machine may be defined as a system which insulates the user from hardware and low-level machine characteristics, and is thus not the same machine as the hardware in which the system is implemented. The software which forms the soft machine can be used readily to optimize user requirements and hardware efficiency requirements for any particular user population acting on any installation. However, divergent user requirements with extremely diverse hardware characteristics will cause any system to run at less than optimum efficiency, and so it becomes necessary to construct an integrated set of several soft machines to satisfy the wide user demands.

The G-systems (G-EXEC, G-MODL, G-FILE) form soft machines which together cover most of the requirements for data handling in the environmental sciences as well as in many other applications. G-EXEC provides the central data-base handling soft machine; G-FILE allows interactive editing, job submission, and output control, including picture editing, in a time-sharing environment; G-MODL will provide a generalized framework for modeling and interactive graphics.

#### THE G-PLOT PACKAGE

The G-PLOT package within G-EXEC consists of a set of modules designed in such a manner that they can form the basis of a comprehensive generalized set of graphics programs, and will allow the display of generated plots on any graphic device which might be available, or the communication of plot files to other installations for editing and display.

For the applications programmer, a generalized graphic package should provide a set of low-level modules which include all the facilities he will require (scaling, transformation, and basic plotting functions) while presenting a 'programmer interface' at a consistent structural level.

The ability to plot to any device is made possible by interposing a communications file or 'logical-plotter' level between the program and the physical plotter. It should be noted that this is at a higher level than the files of plotter commands which may be transferred from a driver program to an off-line plotter—the latter are generally in a specific, highly compressed binary format,

whereas the logical plotter-level file must be in a format accessible to a wide range of driver programs as well as to the other soft machines (allowing, for example, the editing of generated plots). It therefore must be in character form, and for convenience, in 80-column card images. It then is possible to write for each device a driver program, using the manufacturer's low-level software, which will read the logical plotter file and display the graph(s) contained in it.

The development of a consistent set of modules giving the required interface to the applications programmer is more difficult, because of the variety of transformations which may be applied to the data. However, this may be viewed in terms of coordinate systems. The applications programmer is concerned with the data coordinates, which then must be transformed through some sequence of operations into logical plotter coordinates. If one visualizes the logical plotter as a finite set of integer increments in two orthogonal directions, this defines a 'raster' (a term borrowed from television electronics, in which it refers to the rectangular area of screen illuminated by the TV scan).

The sequence of operations mapping data values into the logical plotter raster may define a set of rasters at different logical levels. By defining a consistent and complete set of types of transformation, it is possible to generalize the software and provide a programmer interface at the level of data values. The applications programmer has the simplified task of selecting the transformation options he requires (or the simpler task of passing through user-supplied options), without the necessity of explicitly coding all transformations and directly calling low-level plotter subroutines.

Plot data are communicated in files which have a 'packet' structure for generality (the files are self-describing) and the packet protocol is handled by a set of subroutines also in the G-PLOT package. The structure of these 'management' modules is integrated with that of the modules more specifically concerned with plotting.

#### RASTER LEVELS

The transformation of data into logical plotter coordinates employs altogether four transformation steps between the 'data space' and 'plotter space' (Fig. 1). This may be illustrated by using a simple example from geochemistry: a set of analyses of samples for copper and lead, of which the logarithms (to base 10) are to be plotted as a simple scatter diagram.

The values of minima and maxima are indicated in Table 1, together with Cu and Pb analyses of a typical sample. The transformation steps and their effects on the limits and on the sample are outlined here and summarized in Table 1.

(1) Mathematical function transformations, independently on  $x$  and  $y$  directions (e.g. logarithmic, square root, inverse, or no transformation) from the data raster (DR) to the transform raster (TR). In the quoted example, the limits for both  $x$  (Cu) and  $y$  (Pb) are to be transformed using base 10 logarithms.

(2) Scaling of  $x$  and  $y$  directions independently, from



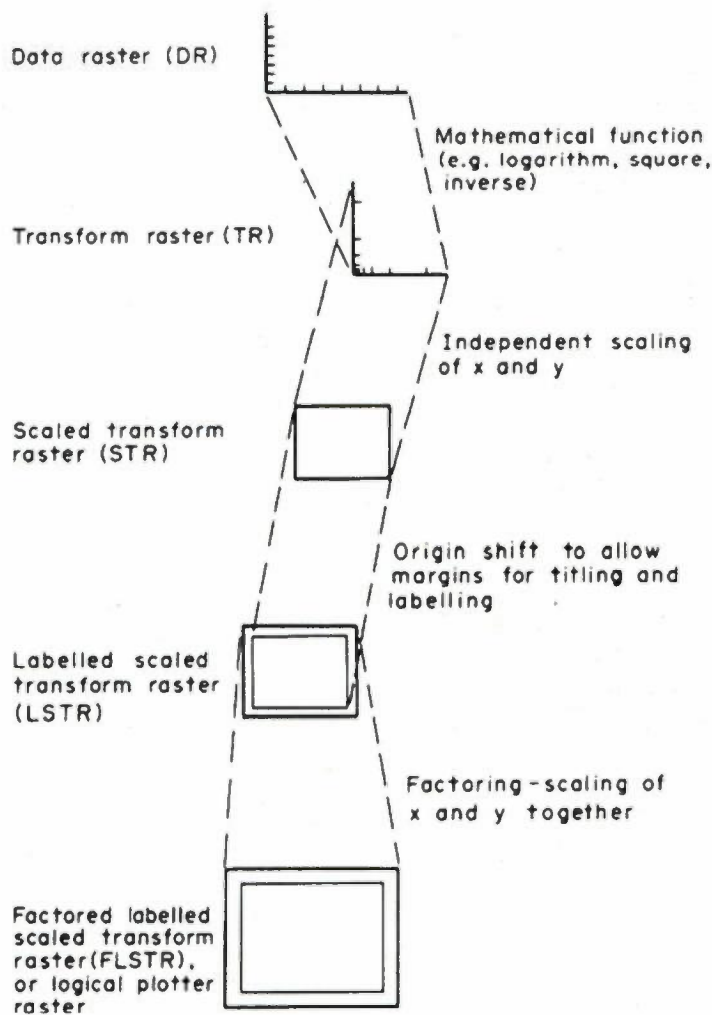


Fig. 1. There are four transformation steps between data level and logical-plotter level.

TR to the scaled transform raster (STR). As a part of this operation, 'round number' limits may be set, either from user-supplied values, or automatically, to replace the 'raw data' limits.

The scaling factor itself may be supplied by the user or may be set automatically by the program to give a default plot size. For example, in this example the logarithm limits may be enclosed by new limits of, say, 1.5 and 3.0 (for Cu) and 2.0 and 3.0 (for Pb); then, to give a data plotting area 600 units by 600 units, these values will need to be multiplied by scaling factors of 40 (Cu) and 60 (Pb) to give ranges of 600 on each coordinate. The coordinates for each sample will be multiplied by the same scaling factors.

(3) Shift of the origin to allow the insertion of labeling and titling around the data-plotting area; the scale

remains the same but the limits are changed, to give the labeled scaled transform raster (LSTR). The size of margin around the plot depends on the particular program which is being used, and is fixed by the program: no user control is necessary. For example, in the example of Table 1, the left and top margins are each of 100 units, and the right and bottom margins are of 50 units.

It is on this raster that text and symbol sizes are defined and the complete plot is assembled.

(4)  $x$  and  $y$  coordinates are scaled together—they are multiplied by a constant to give the factored labeled scaled transform raster (FLSTR) or the logical plotter raster, in whatever plotter units are desired—in the quoted example, a raster of size 1000 by 1000.

In each transformation, it should be noted, G-PLOT modules allow default values which can be overridden by options and parameters supplied by the user, so that it is permissible to generate a plot from a data set about which one has no prior information, or to produce a precisely scaled map with given limits.

#### THE PLOT-FRAME CONCEPT

Apart from the use of a set of fixed raster levels, certain other standards also are adopted for ease of programming and use.

Each process program (the unit of processing available to a G-EXEC user through the EXEC command) produces a single plot frame for a single set of  $x$ ,  $y$ , and where appropriate,  $z$ , coordinates. The overlay of one frame on another, if required, is achieved by user interaction with the soft machine receiving the plot file—editing the generated pictures.

A plot frame in this context may be defined (circulantly!) as that set of logical plotter commands generated by one process program. However, a more useful definition is that a plot frame consists of a single set of logically related graphic components generated from a single data file, using one set of raster transformations (from DR through FLSTR) and a single rectangular plot raster.

The plot frame itself contains different types of information—titles, axes, grids, and data points or lines—and these types of information may be separated usefully as 'subframes'. Although two or more subframes may overlap in the plotter raster, they are all related to the same set of data and thus have a defined logical rela-

Table 1. Raster transformations on typical data set

Raster		Limits		min	max	Incl. margins		Sample values
		Data min	max			min	max	
DR	Cu	53	470					150
	Pb	220	850					440
TR	Cu	1.72	2.67	1.5	3.0			2.18
	Pb	2.34	2.93	2.0	3.0			2.64
STR	Cu			600	1200	500	1250	872
	Pb			1200	1800	1150	1900	1584
LSTR	Cu			100	700	0	750	372
	Pb			50	650	0	750	434
FLSTR	Cu					0	1000	496
	Pb					0	1000	579



tionship. The definition of subframes makes it possible to select desired parts of any number of frames for plotting, by use of appropriate picture editor software in the 'character editor' soft machine (the G-FILE system: Henley and Jeffery, in prep.)

#### PLOTTER DRIVERS

Plot files generated by the G-PLOT software in general are sent to an 'intelligent' receiving machine which identifies the destination in the file header. The file is directed to an appropriate device queue, or to a queue of received files from which it can be collected by the user, for interactive editing or selection of frames or subframes for plotting as described here. In any event, the plot file is passed through a queue to a plotter-driver program which can recognize the G-SEND protocol (Henley and others, 1977) and sends plotting instructions to the device with rescaling if necessary to ensure that the plot frame is not too large for the device.

For interactive editing of plot files, as indicated here, the G-FILE editing system provides a set of basic plot selection and manipulation facilities (select, move, and scale subframes) and used in conjunction with a suitable graphic display terminal, allows the display of plots at intermediate stages during the editing session.

#### FUTURE DEVELOPMENTS

The G-PLOT package, as is the rest of G-EXEC, is designed for ease of upgrading, and should readily provide upward compatibility for tasks such as geological modeling and the iterative generation of long sets of frames to be used for cine sequences.

Three-dimensional applications, also, probably will be handled by the G-PLOT package in G-EXEC rather than

by additional software in the 'front-end' editing system, because of the contrast in power between the two soft machines as well as the principle of logical separation of the different task components. Three-dimensional plots representing progressive rotation, transformation, and scale changes can be generated as above, as sequences of slightly differing frames.

In a different field, that of report generation, the use of G-PLOT in conjunction with report generation programs and an interactive text editor might allow simply the production of text pages which include illustrations, given the suitable devices such as dot-matrix printer-plotters for output of the combined reports.

#### SUMMARY

G-PLOT, in providing a framework for graphics programs which is integrated with a generalized data-handling system, allows a simple set of user commands to generate a wide range of types of graphic output from any of the files to which the user has access in the system, and for display to any available graphic device or for transmission to a local or remote editing soft machine such as G-FILE for a limited set of selection or manipulation procedures before plotting.

#### REFERENCES

- Cubitt, J. M., 1976, An analysis and management system suitable for sedimentological data, in *Quantitative techniques for the analysis of sediments*: Pergamon, Oxford, p. 1-19.
- Henley, S., Jeffery, K. G., Fage, C. J., and Gill, E. M., 1977, Communication of geological information among different soft machines: *Computers & Geosciences*, v. 3, no. 3, p. .
- Jeffery, K. G., and Gill, E. M., 1976a, The geological computer: *Computers & Geosciences*, v. 2, no. 3, p. 347-349.
- Jeffery, K. G., and Gill, E. M., 1976b, The design philosophy of the G-EXEC system: *Computers & Geosciences*, v. 2, no. 3, p. 345-346.