

INTERACTIVE GRAPHICAL SYSTEMS FOR MATHEMATICS, A SURVEY*

by

Lyle B. Smith
Stanford University Computer Science Department and
Stanford Linear Accelerator Center
Stanford, California, 94305

Introduction

Numerical analysis, the mathematics of computer scientists, is sometimes not well-liked by computer science students. However, by putting the mathematics on-line with an interactive graphical system its study can be made more interesting and even fun. Also, the use of numerical analysis by engineers, physicists, and numerical analysts to solve practical problems can be made more effective when an interactive graphical system is available.

In a panel discussion of promising avenues for computer research, Uncapher [1] points out that present systems for on-line computing and problem solving assistance, are not oriented toward the thousands of practicing engineers in the United States. They may be very useful to the computer specialist and the skilled programmer but the casual user is not helped. Recently there has been more effort to make the power of an on-line system available to the casual user as proposed by Uncapher. Some efforts in this area are included in the following discussion of various on-line systems. In particular, the on-line data-fitting system developed by this author is designed for the use of non-specialists in computer science.

Existing and Proposed Systems

Here we list some interactive graphical systems. This list is compiled from a search of the literature and thus may omit some very interesting systems that are in use or under development but have not been described in the computing literature. The systems are grouped according to their applicability. That is, a system that provides the general mathematical capabilities to solve a variety of problems will be classed as a "general purpose" system. On the other hand, a system oriented toward a specific problem area; such as data-fitting by least squares, will be classed as a "special purpose" system. A paper by Ruyle et.al. [2] contains a comparison of AMTRAN, Culler-Fried, the Lincoln Reckoner and the MAP System. References for all the following

* Work supported by U.S. Atomic Energy Commission.

systems are given in [3].

General Purpose Systems

- 1) Culler-Fried at Santa Barbara, California. Culler-Fried at TRW Systems, Redondo Beach, California. (Employs pushbutton programming of mathematical problems through a 96-key console with a storage CRT for display of results)
- 2) TOC at Aiken Computation Laboratory, Harvard University. (Based on Culler-Fried with extensions and modifications)
- 3) AMTRAN at NASA Marshall Space Flight Center in Huntsville, Alabama. (Also based on Culler-Fried. Employs 22⁴ function buttons and two 5-inch CRTs for alphanumeric and graphical display)
- 4) The Lincoln Reckoner at Lincoln Laboratory of Massachusetts, Institute of Technology. (The aim is to provide a system whereby a scientist can reduce his laboratory data without previous programming experience. The system is operation oriented.)
- 5) MAP at Massachusetts Institute of Technology. (Uses a language instead of pushbuttons. Includes high level functions such as integration and linear least squares analysis.)
- 6) NAPSS at Purdue University. (Being developed to "make the computer behave as if it had some of the knowledge, ability and insight of a professional numerical analyst." Employs a language for interactive coding.)
- 7) POSE at Aerospace Corporation, San Bernardino, California (Under development. Similar to NAPSS but less work on comprehensive algorithms.)
- 8) OPS-3 at Massachusetts Institute of Technology. (Includes many standard operators to ease the user's job. A user can write his own compound operators. Simulation is also a primary use of OPS-3. No graphics.)
- 9) JOSS at the RAND Corporation, Santa Monica, California. (The JOSS language permits editing, computing, and typing of mathematical problems. No graphics.)

Special Purpose Systems

- 1) STATPAC at Decision Sciences Laboratory, Bedford, Massachusetts. (Displays a menu of the vocabulary on a CRT. Oriented toward statistical operations.)
- 2) Marchuk and Yershov in USSR. (A proposed system for on-line solution of differential equations.)
- 3) Gear at University of Illinois. (An on-line system for solving ordinary differential equations -- no graphics.)
- 4) Dixon at University of California, Los Angeles, California. (On-line displays used with packaged statistical programs)
- 5) PEG at Stanford Linear Accelerator Center (SLAC), Stanford, California. (PEG is an on-line data fitting system using the method of least squares. The design encourages use by non-programmers as well as programmers.)

The PEG system is designed to be very simple to use and yet retain flexibility and power to handle a wide spectrum of data-fitting problems. PEG has some basic data manipulation capabilities such as point deletion, point correction, subset selection, and a selection of transformations. The system also allows a choice of function for use as an approximating function. The choices include orthogonal polynomials, Fourier approxi-

mations, spline functions, and user defined functions. PEG is coded in Fortran for two reasons: ease of implementation and some degree of machine independence. The system is operating on an IBM 360/91 with an IBM 2250 display unit.

Other Systems of Interest

- 1) DIALOG at IIT Research Institute, Chicago, Illinois. (An on-line algebraic language with graphics).
- 2) MATHLAB at Massachusetts Institute of Technology and the Mitre corporation (on-line assistance in symbolic computations).
- 3) MAGIC PAPER at Bolt, Beranek and Newman Inc., Cambridge, Massachusetts (primarily for on-line symbolic mathematics).
- 4) DISPLAY at System Development Corporation, Santa Monica, California (primarily for graphical examination of stored data -- allows excursions into TINT, an Algol-type interpreter)
- 5) Klerer-May system at Columbia University, Hudson Laboratories, Dobbs Ferry, New York. (Employs modified Flexowriters to achieve "natural" two-dimensional mathematical notation)

Solving a Least Squares Data-Fitting Problem

The PEG system is designed so that it can easily handle a non-linear least-squares data fitting problem with constraints. This is accomplished by minimization of the sum of squares of the residuals. If constraints are present, the code is modified to return a large number for the sum of squares whenever the constraint is violated. In this manner a "wall" is placed around the acceptable region in parameter space and minimization yields the desired least-squares fit for non-linear problems. PEG solves directly the linear least squares problems of fitting by polynomials, Fourier approximations, and splines with fixed joints.

Linear least squares fitting problems can be solved by invocation of a built-in least squares routine or by setting up and solving a matrix equation in NAPSS, OPS-3, the Lincoln Reckoner and MAP. To include non-linear, constrained problems involves varying degrees of additional console programming with these systems. Using the Culler-Fried, JOSS, or TOC systems involves considerable console programming to solve even the linear least squares problem. User-defined console programs could be defined and stored for later re-use.

To solve a non-linear, constrained least squares problem with PEG we first code a Fortran function to evaluate the fitting function. This routine returns an unusually large value if a constraint is violated. PEG then uses the coded function and presents options for lightpen selection on a CRT during the fitting process. Graphical displays of the fit are also presented on the CRT for examination.

Summary

Just as we have various levels of programming languages for coding in a batch environment we have various levels of interactive systems. Each level has its usefulness for particular classes of problems and programmers. These levels are summarized in Table 1 where for each level in batch processing we give an example on-line system.

Batch-Processing	On-line Interactive Processing
Assembler Language	Culler-Fried System (the function buttons provide very basic capabilities)
Procedural Language (e.g., Algol, Fortran)	JOSS (provides on-line algebraic computation)
Using a comprehensive library of subroutines	MAP (has operators backed by comprehensive algorithms)
Using "canned" programs (which only require data)	PEG (given the data, PEG then leads a user through the data-fitting process)

Table 1.
Programming Levels Comparison

- [1] Uncapher, K. The man-machine interface. 1965 Fall Joint Computer Conference, Part II, Spartan Books, Washington, D.C. (1966), p. 88-91.
- [2] Ruyle, A., Brackett, J.W., and Kaplow, R. The status of systems for on-line mathematical assistance. Proceedings -- 1967 ACM National Conference, Thompson Book Company, Washington, D.C., (1967), p. 151-167.
- [3] Smith, L.B. The use of man-machine interaction in data-fitting problems. Ph.D. dissertation, Computer Science Department, Stanford University (to appear).