A group of programs, called SUMX, *is used for statistically analyzing high-energy physics data by batch-processing techniques.*

Against this background, the paper discusses the first phase of a project directed toward placing the SUMX *user on-line via a display console. On-Line* SUMX *provides a helpful interactive mode of computer use in an inherently difficult application area.*

The experimental environment of the data source is discussed. Presented are functions of component programs as well as the types of statistical analyses performed.

INTERACTIVE GRAPHICS IN DATA PROCESSING

Analysis and display of physics data

by W. C. McGee, H. R. Penafiel, and S. K. Howry

Centers for the study of high-energy physics produce vast quantities of data that must be analyzed. This analysis is frequently accomplished through a class of computer programs known as SUMX, whose principal function is to produce statistical summaries of experimented data. Although intended primarily for use in physics laboratories, SUMX should be useful where large amounts of data are statistically analyzed.

The main purpose of this paper is to describe the operation of phase 1 of a user-interactive version of SUMX, termed On-Line SUMX. Presented is a characterization of the experimental environment in which SUMX has been used. Against this background, proposed ways of putting the SUMX user on line—making SUMX interactive—are outlined. In discussing On-Line SUMX, being developed by the International Business Machines Corporation and the Stanford Linear Accelerator Center (SLAC), emphasis is placed on the operational phase 1. This discussion centers on methods of creating and manipulating the objects of On-Line SUMX—event files, arrays, and text sets. Also given are special implementation features of phase 1. Mentioned briefly are some planned extensions to the on-going development of On-Line SUMX.

Experimental environment

In a typical high-energy physics experiment, a target of known material is bombarded by a beam of energetic particles (e.g., protons or electrons) from an accelerator to produce a secondary beam of particles (e.g., photons and mesons). The secondary beam enters

a chamber containing liquid hydrogen under high pressure (bubble chamber) where the secondary particles interact with the hydrogen atoms, producing other kinds of particles. In addition to participating in nuclear events, the charged particles in the chamber ionize the hydrogen atoms along their trajectories. Reducing the bubble chamber pressure at the proper moment causes the liquid hydrogen to boil (i.e. produce minute bubbles) in the vicinity of the ionized hydrogen atoms along the trajectories of the charged particles, which appear as visible tracks. At the moment pressure is reduced, the chamber is photographed from two or three different directions, thereby making a permanent record of the tracks.

The photographic pairs or triads are scanned either by humans or by an automatic scanning device such as the flying spot scanner. (Physicists call each pair or triad an "event.") Events of interest are then precisely measured, and coordinates of the tracks in each view are recorded. Track coordinates are inputs to a *geometry* program whose function is to reconstruct the event in three dimensions from the two-dimensional data extracted from the photographs. The output of the geometry program, three-dimensional track coordinates, are the inputs to a *kinematics* program that computes the momenta of the particles leaving visible tracks and assigns identities (i.e., masses) to them as well as to any uncharged particles that may have participated in the event. In general, the energy and momentum conservation laws allow a number of alternative mass assignments or hypotheses, and the kinematics program also computes goodness-of-fit to the experimental data for each hypothesis.

The output of the kinematics program is a *data summary tape*, which is a file containing one logical record for each event recorded in the experiment. Each record contains the key variables of an event, such as track coordinates, and particle momenta, masses, and energies. The experimenter is usually interested in knowing the frequency of occurrence of certain types of events or the frequency of observation of particles with certain energies and directions. Reliance on these frequencies requires large numbers of events. It is not unusual for an experiment to involve several hundreds of thousands of events, with the record size for each event ranging from a few dozen to several thousand words.

the SUMX program

The operations performed by the experimenter on a data summary tape are substantially independent of the particular experiment. Because of this, it is possible to provide for these operations in a single generalized program known as SUMX. The input to SUMX is a data summary tape and a set of control cards. By means of control cards, a sample listing of which is given in Figure 1, the user specifies the operations to be carried out. The data summary tape records normally consist of a set of formatted fixed-length FORTRAN variables. The operations provided are principally the formation of one-dimensional frequency distributions of specified variables and the presentation of these distributions in the form of histograms (bar charts), as shown in Figure 2.

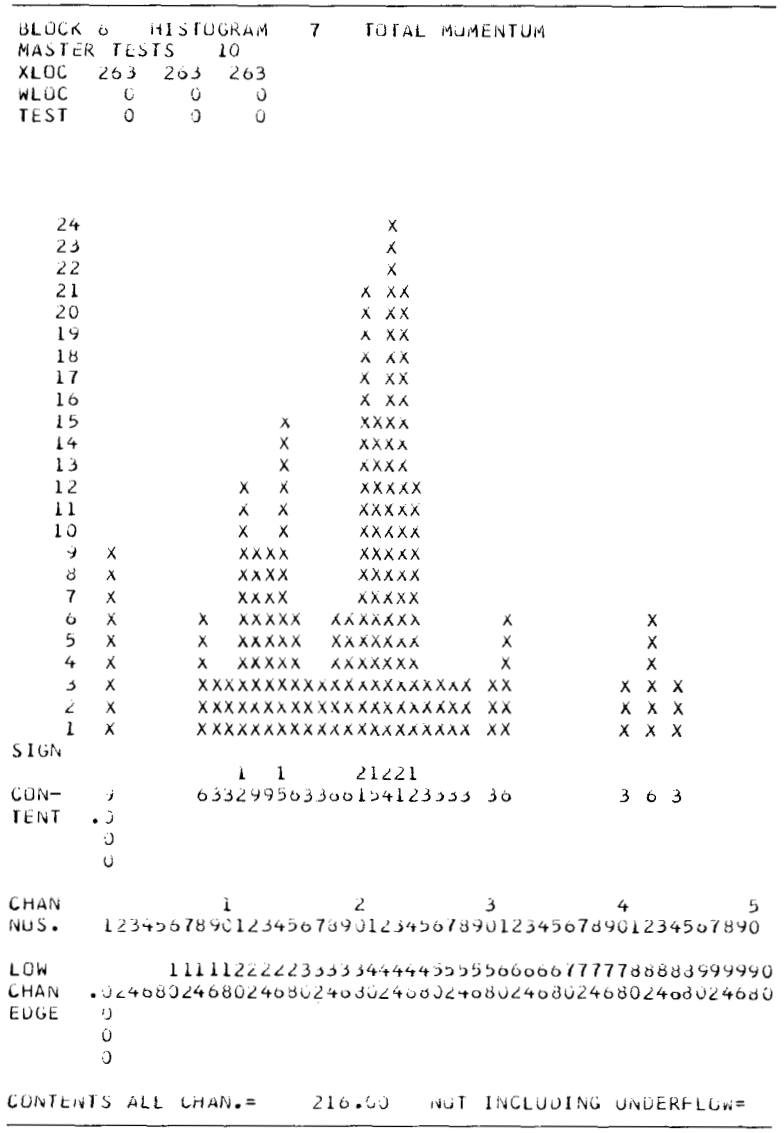
Figure 1 Control-card listing

```
TEXT SET CS2

*NEW PASS
*DISCARD  1
*TAPE
8                       JOHN RETTBERG'S TAPE
*SELECT
TEST        10
            -6          EQU         2
*BLOCK 6
TOTM TOTAL MOMENTUM
40          0.25        0.0         10
263
TOTAL MOMENTUM
50          0.20        0.0         10
263
263
263
*BLOCK7
SCAT XMAGC EXPECTED AGAINST ACTUAL
10                      4
30          20          .2          .2          -3.         -3.
192         174
192         175
193         174
193         175
XMAGC EXPECTED VS. ACTUAL
10                      6
60          20          .1          .4          -3.         -4.
192         174
192         175
193         174
193         175
*ALL DONE
$$$$

END OF TEXT SET...LINE COUNT= 33
```

Provision is made for the conditional inclusion of events based on specified test variables (e.g., event type) and for the weighting of events by specified variables. Other operations include the arithmetic manipulation of previously created distributions, the computation of means and variances, the listing of event data, and the generation of subsets of the data summary tape.

The original version of SUMX was written at the University of California, Lawrence Radiation Laboratory, Berkeley, California.[1] Other versions of the program have been developed at the European Organization for Nuclear Research, Meyrin-Geneva, Switzerland (CERN)[2] and at the University of Chicago, Argonne National Laboratory, Argonne, Illinois. SUMX has been written in a variety of languages, including FAP, FORTRAN II, and FORTRAN IV. It has been run on a variety of computers, including UNIVAC 1107, IBM 7090, CDC 3400 and 6600, and IBM SYSTEM/360.[3] The programs are potentially useful in such applications as questionnaire processing, census

Figure 2 *Histogram printout*



tabulation, and many types of physical and biological experiments where large amounts of data are processed to produce frequency distributions.

Although the SUMX programs allow the user to generate a number of outputs on a single run of the input tape, the user frequently does not know which outputs to request. He often uses the output from one run to suggest things to look for in the next run. Therefore, many runs (and many days) may be required before meaningful conclusions can be drawn from the data.

putting users on-line

To alleviate this problem, it has been proposed that the SUMX

user be placed "on-line" to the computer in order that he may make many output requests during a single session. This proposal is economically feasible because the computer would be shared by several users. The simplest approach would be to provide users with three basic capabilities:

- Defining, through keyboard entries, relevant features of the input file—file location, record length, record format, etc.
- Supplying parameters for a previously programmed input-file transformation.
- Applying the specified transformation to the defined input file.

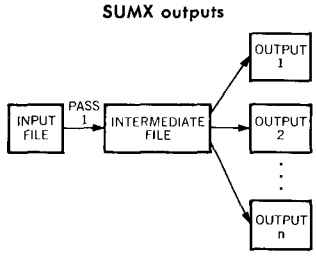
With these capabilities at his disposal, the user would typically proceed as follows:

1. Define an input file.
2. Define a transformation, such as a histogram of variable x between limits *xmin* and *xmax* using an interval dx.
3. Apply the transformation to the input file.
4. Observe the output on the display device.
5. Redefine the transformation, e.g., by changing dx.
6. Repeat steps 3, 4, and 5 as often as necessary.

A potential problem with this approach lies in step 3. The input file normally contains a large number of events, and the time to apply the defined transformation to the file may be of the order of minutes or tens of minutes. Not only does the transformation consume computer time, but also, more importantly, it consumes user time—while the input file is being processed, the user must wait.

Several ideas suggest themselves. Provision could be made (as it is in the existing SUMX) for the user to define multiple transformations and perform them on the input file during a single pass. Thus, the user would have a number of outputs to compare. Another idea is to permit the user to create subsets of the original input file in one pass and use these smaller files on subsequent passes. (This facility is provided in the existing SUMX.) A third idea is to allow the user to retain, in a suitably abstracted form, the results of a pass, and to subsequently manipulate these intermediate results. As an example, a histogram can be retained at the end of a pass in the form of a matrix that can be printed, displayed, or used in arithmetic operations without repeating the pass. This solution can be generalized by providing the user with two kinds of transformations: (1) input-file-to-intermediate-file, and (2) intermediate-file-to-output. In this way, a number of transformations of the input-to-output type can be effectively provided with only a single pass of the input file. The third method is shown schematically in Figure 3.

Figure 3 Generating On-Line SUMX outputs



While the foregoing ideas could help to minimize waiting time, they probably would not eliminate it. Delay seems to be an innate characteristic of those problems that involve the formation of statistical quantities.

On-Line SUMX

To test the feasibility of putting the SUMX user on-line, an On-Line SUMX program for SYSTEM/360 is being developed as part of a joint project between IBM and the Stanford Linear Accelerator Center (SLAC). The project is based on the CERN SUMX program,[2] because many physicists at SLAC are familiar with it and because of the availability of the source program which is being furnished through the courtesy of CERN.

On-Line SUMX is being written in phases in order to obtain user evaluation in parallel with development. We now describe the facilities and special implementation features of the phase 1 program, which is complete, and briefly mention the facilities planned for later phases.

phase 1

The On-Line SUMX user starts by entering control cards through the normal job stream, along with his data summary tape. At an IBM 2250 display console, the user observes a "sign-on" display at the appropriate time indicating that he enter a command. When the command is entered, the program responds by giving an appropriate notification that the command has been carried out. If the program is unable to carry out the command, a reason is similarly given. In either case, the program awaits the next user command. The basic cycle of user command and program response is repeated until the user enters a command that terminates the session. From the user's standpoint, the key feature of this procedure is that the program does nothing except in response to his commands.

Commands may be entered via two sources: the 2250 alphanumeric keyboard and cards submitted with the job deck. Keyboard entry provides the normal interactive method of program use, while card entry facilitates the entry of common command sequences and permits the program to be run unattended (i.e., in the batch mode). The card reader is in control at the start of the program, and commands are read from cards and executed (without pause) until a QUIT or GO TO KB (keyboard) command is encountered. For GO TO KB, control is transferred to the keyboard through which the user may enter subsequent commands. Control remains in the keyboard until the user enters a QUIT command or a GO TO CR (card reader) command. In the latter case, control reverts to the card reader with subsequent commands being taken from cards.

Regardless of their source, commands have the form of alphanumeric strings typically consisting of a verb in the imperative mood followed by one or more parameters. Keyboard commands are displayed as they are entered, and completion of command entry is signaled by the END key. Neither the function keyboard nor the light pen is used in the phase 1 program primarily to obviate the development of separate procedures for command entry from the 2250 console and from the card reader.

The user of On-Line SUMX is concerned with the creation and

manipulation of objects of three types:

- *Event files* contain experimental data with one record for each nuclear event. In the phase 1 program, the user is limited to one event file, which must have the format prescribed by the CERN SUMX program.
- *Arrays* are frequency distributions that are constructed by counting numbers of event file records (events) having variable values lying in certain discrete intervals. In the phase 1 program, distributions of one and two variables are provided.
- *Text sets* are sets of 72-character alphanumeric strings. In the phase 1 program, text sets are used primarily for entering the parameters required by SUMX functions. A 2250 text-set display is shown in Figure 4.

Objects are created and manipulated in an abstraction called the "accumulator." For example, the command READ causes a text set to be read from cards into the accumulator, and the first of possibly many pages of the text set is displayed. The contents of the accumulator are continuously indicated on the screen by displaying: object *type*, i.e., event file, array, or text set; object *name*, a four-character string assigned by the user; and the object *remarks*, a twenty-character annotation string assigned by the user.

A name and a set of remarks are assigned to a new object through the command PUT (name). This command causes the name specified in the command to be attached to the object in the accumulator and the object and its remarks to be filed in direct-access storage. If an object with this name already exists in the file, the command is rejected. The same name may be used to designate different objects successively by issuing the command CLEAR (name) prior to the PUT command. The command MAP may be used to display the name, type, and remarks of each object currently in the file.

The command GET (name) causes the named object to be placed in the accumulator and a picture of the object to be displayed. Figure 4 shows a text set, while Figures 5 and 6 illustrate one- and two-dimensional arrays. In general, the picture of an object extends over several pages, and the GET command displays only the first page. To see a particular page of an object, the command PAGE (number) may be used. Similarly, the commands NEXT and PREV may be used to turn pages forward and back. Objects in the file are arranged in the order in which they are filed by the PUT command and may be retrieved in this order with the GETNEXT command. The resulting display is identical to that produced by the GET command. Objects in the accumulator may be printed by the line printer using the command PRINT.

Arrays are created with the command DO (text-set name). This command performs a SUMX pass on the user event file, using the named text set as parameters. The format of these parameters is the same as that of the control cards for the batch CERN SUMX program. This was done for two reasons: (1) to simplify the use

Figure 4 Text-set display

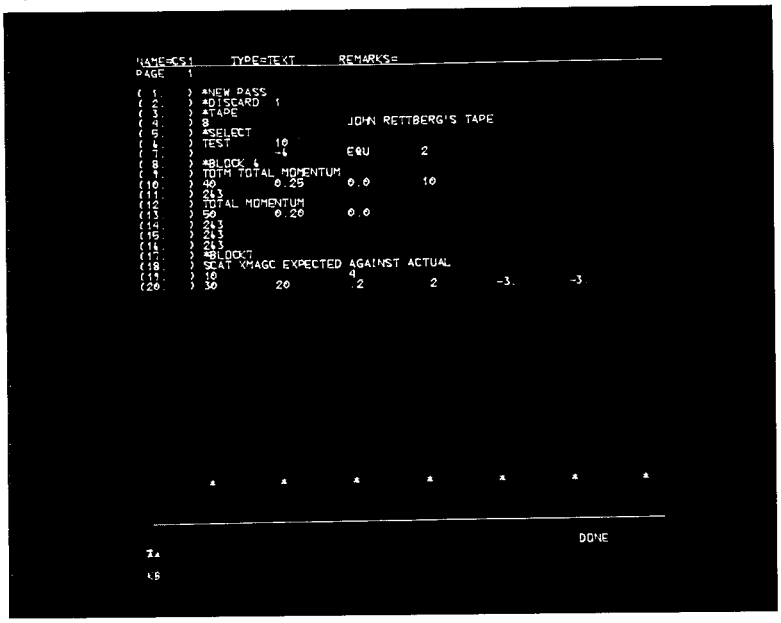
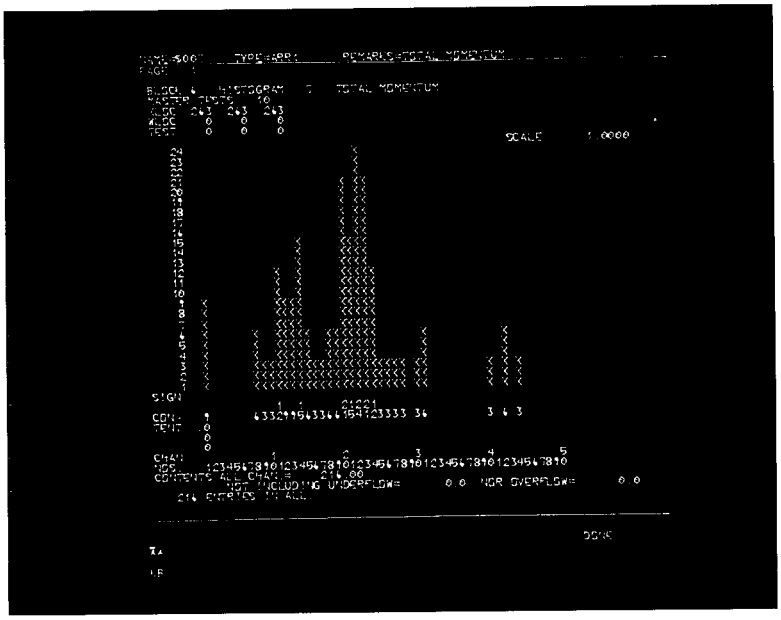
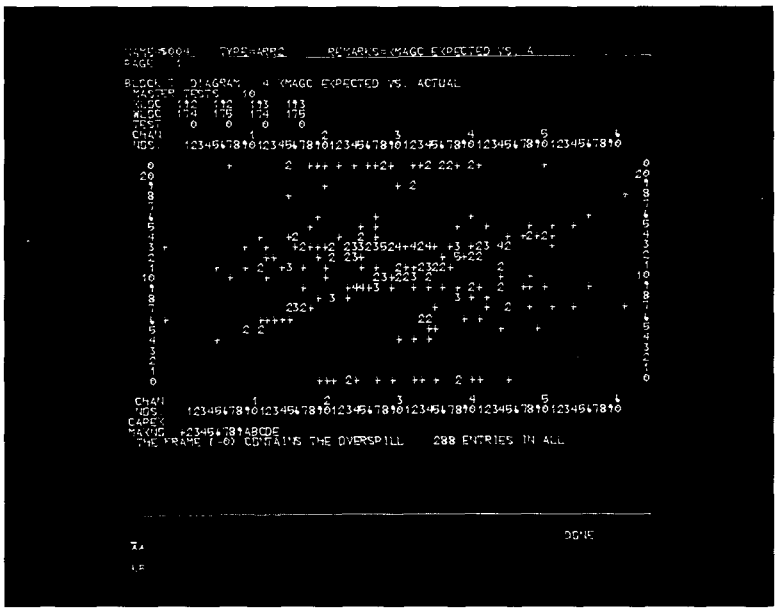


Figure 5 One-dimensional array (histogram) display



of the On-Line SUMX program for persons familiar with CERN SUMX and (2) to utilize as much as possible of the existing CERN SUMX program. Arrays created during a SUMX pass are stored in direct-access storage together with names and remarks specified in the parameter cards.

Figure 6 Two-dimensional display



Text sets are created by reading them into the accumulator by the READ command. A text set in the accumulator may be edited by using the COPY, INTO, and DELETE commands. The command COPY (line number) causes the designated line of text to be copied into a working area on the display screen, where it may be modified by means of the 2250 alphanumeric keyboard. The working area line is placed back in the text by the command INTO (line number). If the line number in the INTO command exists in the text, the new line replaces the existing one; otherwise, the new line is inserted in the text in numerical sequence. The command DELETE (line number) causes the designated line to be deleted from the text.

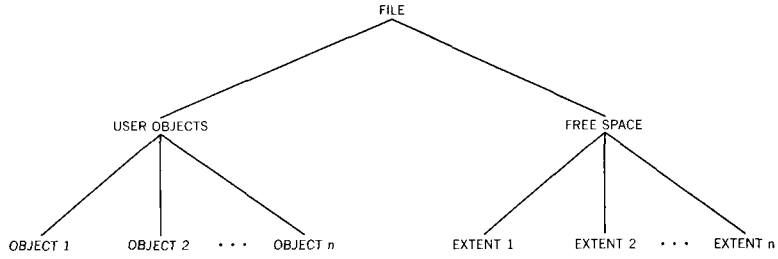
special features

Three aspects of the implementation of phase 1 On-Line SUMX are worthy of special note: (1) a large contribution from the CERN SUMX program to implement the SUMX functions of phase 1, (2) the use of the direct-access file facility of FORTRAN IV to implement the object file, and (3) the use of a table-driven command syntax interpreter.

The SUMX functions of phase 1 of On-Line SUMX, i.e., the operations carried out in response to the DO command, have been implemented using selected routines from the existing CERN SUMX program. Not only was this an efficient way to implement the on-line functions, but it also gives the On-Line SUMX user access to the same functions available to the batch SUMX user. Some 150 routines were extracted from a specific version of CERN SUMX. Minor modifications were required to do the following:

- Make the SUMX main program a closed routine.

Figure 7 File structure for phase 1 On-Line SUMX



- Store the SUMX outputs, such as arrays, in direct-access storage instead of printing them directly.
- Communicate SUMX control card syntax errors and other user errors via the display console rather than the printer.

(Since implementing phase 1 of On-Line SUMX, an updated version of CERN SUMX has become available, and plans call for using the latter in subsequent On-Line SUMX development.)

On-Line SUMX was not implemented simply by combining the 2250 console with an existing batch-SUMX program. The batch program has the wrong structure for on-line use and lacks facilities (e.g., text editing) that the on-line user requires and the batch user does not. On-Line SUMX has an entirely new structure, designed specifically for on-line use, in which significant portions of an existing batch program are imbedded.

In the phase 1 program, the user object file is maintained in direct-access storage. A single data set is used to hold all objects. By means of the DEFINE FILE statement in FORTRAN IV, the data set is partitioned into blocks of fixed length, and each object occupies an integral number of blocks. Objects are read from and written into the data set using the direct-access form of the FORTRAN READ and WRITE statements. A directory is maintained in main storage containing the name and starting block of each object. In addition, each directory entry contains sufficient pointers to enable the file to be structured as a tree. The general tree form of the file structure for phase 1 On-Line SUMX is indicated in Figure 7.

It is expected that the command language will be expanded as a result of physicists' using and evaluating the system. In order to facilitate this expansion, two techniques are included that allow the command interpreter to be driven by an arbitrary command language. The first technique is to read the command language from cards in Backus Normal Form (BNF) and create tables. The second technique uses a generalized "parser," which searches the tables to detect syntax errors in commands as they are entered and to transform valid commands into simple parameter lists for the command interpreter. The command interpreter is merely a collection of routines, one for each command in the language, each of which has its own particular parameter list. Thus, in order

to augment the language, it is necessary only to change the syntax cards and to add an appropriate routine to the command interpreter. The syntax of the phase 1 command language, given in Table 1, is simple and involves no recursive definitions. However, such definitions can be accommodated by the existing table-building and command-parsing routines.

Planned extensions

One of the first extensions planned for On-Line SUMX is the facility for creating and storing a number of event files. This facility would permit the user to create subsets of his principal input file and, hence, reduce the time required to process variables and events of interest. The facility for creating event files is already present in the CERN SUMX program; incorporating it in On-Line SUMX requires only the development of techniques for storing event files in direct-access storage (and possibly also in main storage).

Another planned extension would display scatter plots, i.e., diagrams in which pairs of variables are represented by points in a plane. One method of accomplishing this is with the command PLOT (x-variable, y-variable), which converts the event file in the accumulator into a scatter plot of the specified x-y pair from each record in the event file.

An important adjunct to the scatter plot facility is that for generating contours or closed curves of special significance to the physicist. The command CONTOUR (contour parameter list) would place in the accumulator a picture of the contour specified by the

Table 1 Command language syntax for phase 1 On-Line SUMX

Definitions	
Universal delimiter	The space character.
Word	Any string of characters not containing the universal delimiter.
Terminal set	{READ GETN ENTER PRINT DO PREV NEXT QUIT GET PUT CLEAR COPY INTO DELETE PAGE GO MAP}
Terminal quantities	{id no}
no	Any word whose first character is 0, 1, ···, 9, +, or −. Such a quantity is interpreted as a real number.
id	All other words excepting the terminal set.

Syntax		
⟨instr⟩	:: =	⟨vb0⟩ | ⟨vb1⟩ ⟨id⟩ | ⟨vb2⟩ ⟨no⟩ | ⟨vb3⟩
⟨vb0⟩	:: =	READ | GETN | ENTER | PRINT | PREV | NEXT | QUIT | MAP
⟨vb1⟩	:: =	GET | PUT | CLEAR | DO
⟨vb2⟩	:: =	COPY | INTO | DELETE | PAGE
⟨vb3⟩	:: =	GO ⟨id⟩ | GO ⟨id⟩ ⟨id⟩

supplied parameters. The picture can be saved with a PUT command and later recalled and combined with any one of a number of scatter plot pictures. For example, the sequence

```
PLOT   x, y
ADD    K1
```

would place in the accumulator a scatter plot of variables x and y and superimpose the contour (or any other picture) named $K1$. Another planned picture-manipulation operation includes scaling and translation to permit simultaneous viewing of two or more separately produced pictures.

By the end of a session, the user frequently will have accumulated a set of objects that he wishes to retain and use as a starting point for the next session. To permit this, it is planned to provide a DUMP command, which would save specified objects on a selected output medium (e.g., magnetic tape) in a format suitable for the reloading of direct-access storage at the next session.

The phase 1 program is heavily oriented toward the CERN SUMX user. To make the program more accessible to the general user, two additional facilities are planned. The first would allow a user to introduce his own data-computation program into the system and subsequently call it via the keyboard with a generalized DO command: DO (program name, parameter set name). In this way, the user would not be restricted to the functions provided in the CERN SUMX program or to any other program that might be supplied with the system.

The second facility for making the system more accessible for general use is to provide keyboard commands for performing certain simple data-reduction operations directly, rather than by a preprogrammed processor. Examples of operations that might be provided in this manner are: making file subsets, listing, counting, constructing frequency distributions, finding extremals, and performing arithmetic operations on vectors and matrices. The language for doing this would be an extension of the present language. For example, the command sequence

```
GET      file A
FOR      (logical-expression)
WRITE    (variable list)
PUT      file B
```

would select from file A those records for which a given "logical expression" is true, transform those records into records containing variables specified in the variable list, and store the resulting records as file B.

Concluding remarks

On-Line SUMX is a SYSTEM/360 program that enables the user to control the summarization and display of any data file via the 2250 display console. Although intended primarily for experimental

physicists, the program should be useful to others who must summarize large amounts of data in order to derive meaning from it. The program capitalizes on the unique ability of a CRT display device to present data in forms and at speeds that cannot be attempted on conventional output devices, and thus it provides a useful interactive mode of computer use in an inherently difficult application area.

ACKNOWLEDGMENT

The authors are indebted to Dr. Paul J. Friedl of the IBM Palo Alto Scientific Center for the inspiration for an on-line SUMX program and to Dr. Richard Brown of the University of Illinois for his continuing encouragement. The careful documentation by the authors of the CERN SUMX program is also greatly appreciated.

CITED REFERENCES

1. L. Champomier, FORTRAN *program* SUMX, University of California (Berkeley) Report, UCRL-11222 (April 15, 1964), may be obtained from the author at the University of California, Berkeley, California.
2. J. Zoll, *T. C. Program Library Notes*, SUMX Section, PS/4447 may be obtained from the author at the European Organization for Nuclear Research, Meyrin-Geneva 23, Switzerland.
3. CERN SUMX—*A Data Summarization Program for the* IBM SYSTEM/360, IBM Type III Program 360, D-17.2.006, IBM Program Information Department, Hawthorne, New York.