# The GEDAP Software Package for Hydraulics Laboratory Data Analysis

M.D. Miles and E.R. Funke
National Research Council
Ottawa, Canada

## 1 Introduction

GEDAP[1] is a general purpose software system for the analysis and management of laboratory data including real-time experiment control and data acquisition functions. It has been designed with particular emphasis on random wave generation and analysis in hydraulics laboratory basins but it can easily be used for other applications as well. GEDAP was originally developed by the NRC Hydraulics Laboratory on EAI and Hewlett Packard computers and has recently been extensively revised to operate on Digital VAX computers under the VMS[2] operating system.

GEDAP is a modular software system which is fully integrated by means of a common data file structure and a common set of support routines. It has been specifically designed to provide the following features:

- A standard data file format so that any GEDAP program is able to process data generated by any other GEDAP program.
- A common file structure which can handle all types of laboratory data efficiently with record lengths up to 32,768 samples per channel.
- A broad set of data analysis programs so that most laboratory projects can be handled with little or no project-specific programming.
- On-line documentation for all programs.
- A consistent user interface so that all programs are easy to use in either interactive or batch mode. (GEDAP applications do not require any programming experience on the part of the user.)
- A fully-integrated interactive graphics capability so that results can be conveniently examined at any stage of the data synthesis or analysis process.
- A mechanism whereby data can be automatically identified and labelled with correct engineering units whenever they are listed or plotted.

[1]GEDAP is a registered mark of the National Research Council of Canada.
[2]VAX and VMS are registered trademarks of Digital Equipment Corporation

- Routines to convert data from model-scale units to full-scale units and vice-versa (Froude Scaling)
- Conversion utilities to move data between GEDAP files and other standard formats such as Lotus 1-2-3[3].
- An extensive set of well-documented subroutine libraries so that new application programs can be developed quickly and efficiently.

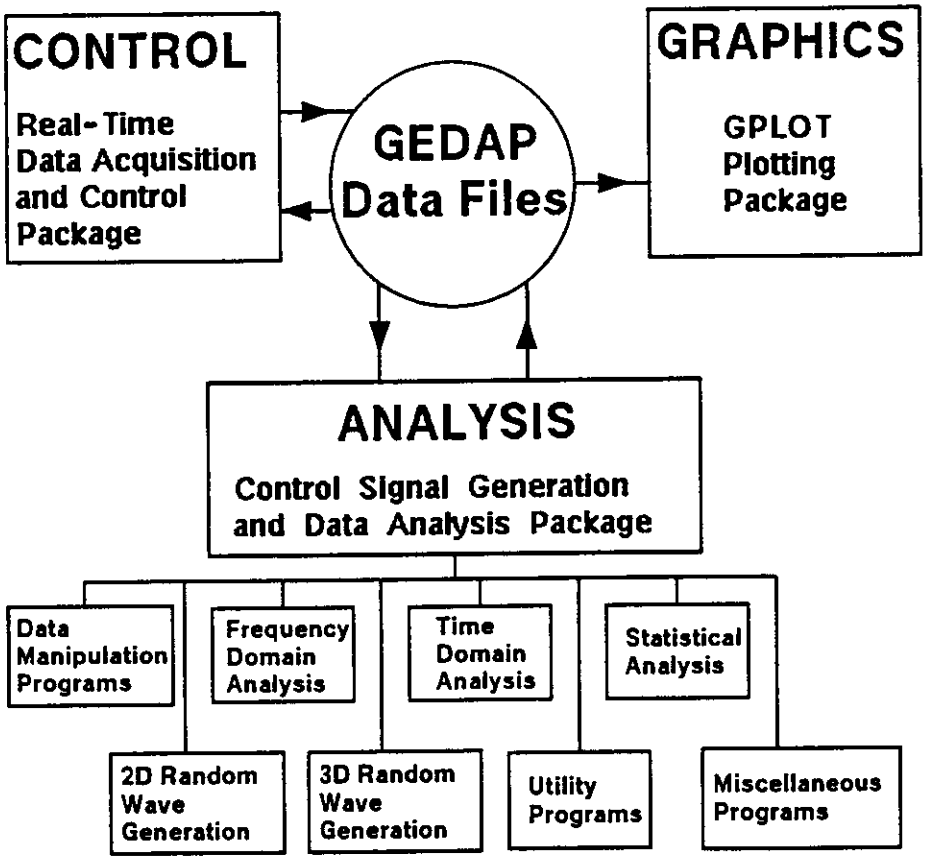


Figure 1: Block diagram of the GEDAP system

A block diagram of the basic GEDAP system is shown in Figure 1. The GEDAP software is organized into three main packages:

CONTROL Performs real-time data acquisition and control functions. This includes sensor calibration, output of wave generator and other control signals, data sampling and real-time display of data during acquisition.

GRAPHICS The GEDAP graphics package is called GPLOT. It performs both interactive and batch mode plotting on a wide variety of devices.

[3]Lotus 1-2-3 is a registered trademark of Lotus Development Corporation.

ANALYSIS The main part of the GEDAP system which handles all of the data analysis functions. It also includes random wave synthesis, control signal generation and various utility programs. The ANALYSIS package contains all GEDAP programs and subroutines except those in the CONTROL and GRAPHICS packages.

This paper describes only the ANALYSIS and GRAPHICS packages. The CONTROL package is described in reference [1].

## 2 A Brief History of GEDAP

The GEDAP software package has been developed over a period of time since 1970 as part of a program at the NRC Hydraulics Laboratory to computerize data acquisition and experiment control functions. During the initial stages of its development, it operated on Electronic Associates Inc. EAI 640 and EAI Pacer computers.

The original disk file oriented structure of the system evolved for the simple reason that the EAI machines had only 16K bytes of memory. As a consequence of this memory limitation, most programming tasks had to be implemented in several stages with the results of each stage being stored on disk for access by the next program. It soon became evident that software development efficiency could be significantly improved by standardizing on the file structure and by designing modular computer programs which could be used by more than one specific application. This also led to the development of a general purpose plotting program which was able to accept the standard data file structure. Because of the fact that the GEDAP file structure carried all pertinent data-definition parameters in a header, the plotting package was able to provide graphic output without operator intervention.

The name GEDAP was derived from "General Experiment Control, Data Acquisition, Data Analysis and Plotting Package". This name was registered in March 1980 as an official mark by the National Research Council of Canada.

In 1976, when the EAI computers reached the end of their useful life, the software system was converted to operate on Hewlett- Packard HP-1000 computers and, in the process, was significantly improved and expanded.

In 1985, the conversion of GEDAP to Digital Equipment VAX computers was started. The initial Phase 1 VAX conversion was confined to the minimum alterations required to make the system operational. A subsequent Phase 2 conversion included major changes to both the data file structure and the software architecture in order to take full advantage of the 32-bit address space and the virtual memory features of the VAX/VMS operating system. Many GEDAP programs were also recoded and generally improved during this process. Phase 2 replaced all previous operational versions of GEDAP as of August, 1988.

GEDAP has also been installed and operated at several Canadian and international organizations in addition to the NRC Hydraulics Laboratory. Since most of these installations were based on previous HP-1000 versions, some may no longer be in service due to hardware obsolescence. Three Canadian university laboratories are now in the process of acquiring the Phase 2 version of GEDAP for operation on VAX workstations. Several European labora-

tories have also shown strong interest in the current version and a system has recently been ordered to support wave generation and analysis for a new dual-flap wave machine at a major ship model testing facility.

## 3 GEDAP Application Programs

The programs in the GEDAP ANALYSIS package are organized in eight categories as follows:

(1) DATA_MANIP — Data Manipulation Programs: These programs perform various functions such as data scaling, resampling to new time spacings, linear transformations, selection of sub-records from longer records, etc.

(2) FREQ_DOMAIN — Frequency Domain Programs: These are data analysis programs which operate mainly in the frequency domain. They usually calculate frequency-domain functions of time-domain data. This category includes various spectral and cross-spectral analysis programs as well as wave reflection analysis programs.

(3) STATISTICS — Statistical Anaysis Programs: These include basic statistics (minimum, maximum, mean, standard deviation), probability density histograms, Goodness-of-Fit tests, etc.

(4) TIME_DOMAIN — Time Domain Programs: These are data analysis programs which operate primarily in the time domain. This group includes zero-crossing analysis of wave records, trend removal, peak detection, time-domain filtering, etc.

(5) UTILITY — General Utility Routines: These include programs to examine the contents of GEDAP files and to perform data conversions between the GEDAP binary file format and various ASCII formats.

(6) WAVE_GEN — 2D Random Wave Generation Programs: These include wave train synthesis for specified wave spectra and wave grouping, episodic wave generation, nonlinear compensations for long and short waves, calculation of wave machine drive signals, etc.

(7) 3D_WAVES — Programs for Generation and Anaysis of 3D Wave Fields: This group includes drive signal generation for segmented wave machines, spreading function generators and various procedures for analysis of 3D wave fields using x-y current meters and arrays of wave elevation probes.

(8) MISCELLANEOUS Miscellaneous Programs: This is a catch-all category for any GEDAP programs which do not fit into any of the previous 7 groups.

New programs are continuously being added to the GEDAP system. A comprehensive On-Line Help facility for GEDAP has been incorporated with the standard VMS Help facility. Thus, information on the GEDAP programs which are currently available can be obtained by simply entering the following command:

```
HELP GEDAP PROGRAMS
```

This will produce a list of GEDAP program categories which can then be used to display single-line definitions of the function of each program. The Help facility is very useful for finding the appropriate program for a particular job but it does not provide any details on the operation of the program itself. Complete documentation on the current version of each program can be obtained by entering the following command:

```
DOCUMENT xxx
```

where xxx is the name of the program. This will create a program documention file named xxx.DOC in the user's directory which can either be printed or examined with an editor.

GEDAP programs can handle both model-scale and full-scale data but analysis is normally performed using only full-scale units in order to avoid confusion. Wave synthesis is also performed in full-scale units and data are converted to model-scale units only at the final stage when the wave machine drive signals are generated.

GEDAP programs also use SI (Metric) units exclusively. If other units such as U.S. Customary are required, then the conversions are usually done only at the last stage when plotting or tabulating results.

## 4 Data File Structure

GEDAP programs use a standard file structure for all input and output data files. This is probably the most important feature of the package since it provides a common protocol by which all of the programs can communicate effectively. Unformatted, sequential Fortran files are used instead of ASCII files so that large sets of numerical data can be handled efficiently. Utilities are provided for listing and modifying the contents of these binary data files.

A GEDAP data file contains 6 records as shown in Figure 2. Record 1 has a fixed length and is used to store the A-Header section. The A-Header contains a fixed set of mandatory parameters which are necessary to define the size and contents of the other five records in the file. Records 2–5 have variable length and used to store four classes of B-Header parameters. Most B-Header parameters are optional. Finally, record 6 is used to store the main data section of the file. This is normally the largest record and it may contain up to 500,000 data points.

GEDAP DATA FILE STRUCTURE

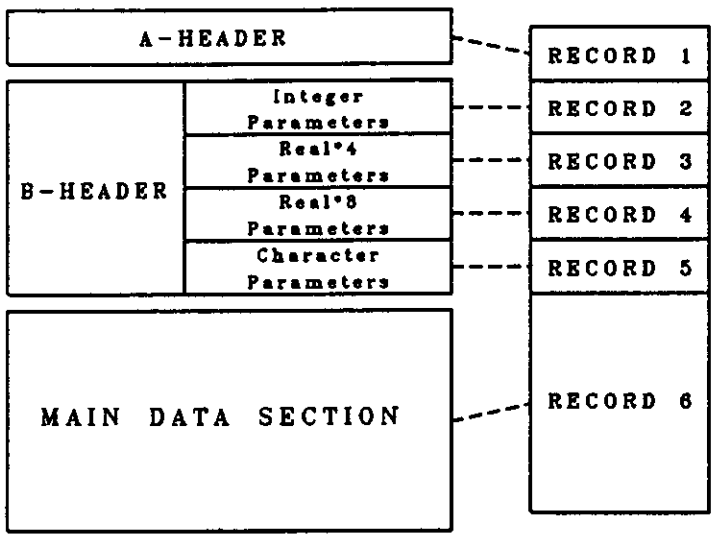


Figure 2: The GEDAP Data File Structure

The A-Header and B-Header parameters perform four important functions:

- They provide a comprehensive description of the data stored in each file.
- They allow GEDAP files to be plotted very easily with all data automatically labelled in proper engineering units.
- They provide a convenient and flexible mechanism for passing ancillary information from one GEDAP program to another.
- They allow project-specific labels to be attached to data files and then automatically propagated through all subsequent stages of analysis. These labels can be easily incorporated when plotting results.

The A-header is used to store certain basic information such as the following:

- the data file type
- the dimensions of each data vector or array
- constants defining any implicit variables such as time or frequency
- text strings defining the data and its physical units
- the name of the GEDAP program which created the data file

The previous HP-1000 version of GEDAP used a fixed length record for all header parameters. This was adequate initially but it gradually led to problems after a few years

because no space could be made available for new parameters except by deleting existing ones. Programming was also rather complicated because parameter values had to be stored and retrieved by their relative position in the header array. Management of the limited header space became increasingly difficult as more and more programs were added to the system and eventually chaos threatened to reign supreme.

The B-header parameters introduced in the VAX version of GEDAP have provided a very effective solution to these problems. B-header parameters are stored and retrieved by name. This means that both the parameter names and their values are stored in the file header. Consequently, the position of a given parameter in the header is not important and may vary from file to file. Although this structure requires an extra 16 bytes of space in the file for each parameter name, it also provides a very flexible data storage mechanism which can handle any number of parameters. In practice, the VAX file headers are not significantly larger than the previous HP-1000 headers because space is only required for the parameters which are actually being used in any particular file. The ability to access header parameters by name has greatly simplified the development and maintenance of GEDAP software.

A GEDAP data file may contain any number of B-header parameters. Furthermore, there is no fixed set of names for the B-header parameters so that new parameters can be easily added to the GEDAP system as required. Temporary names can also be defined for special parameters which are related to a specific project. There are four classes of B-header parameters:

| | |
|---|---|
| **Integer** | Integer values from -2,147,483,648 to +2,147,483,647 |
| **Real*4** | Single precision floating-point values (7-digit accuracy) |
| **Real*8** | Double precision floating-point values (16-digit accuracy) |
| **Character** | 32-character alphanumeric text strings |

By default, all B-header parameters are copied from the input file to the output file when a GEDAP program is run. Each program may also add new parameters. Thus, once they are stored in one data file, B-header parameters are automatically propagated through all subsequent stages of analysis so that they are always available for labelling plots or for use as input data to another program.

B-header parameter names may be up to 16 characters long and may include the underscore character as well as alphanumeric characters. There are seven standard classes of parameters:

| | |
|---|---|
| GENERAL | - General basic parameters such as water depth |
| DAS | Data Acquisition System parameters |
| RWG | Random Wave Generation parameters |
| SPEC | Spectral Analysis parameters |

| | |
|---|---|
| **STAT** | Statistical Analysis parameters |
| **SWG** | Segmented Wave Generator parameters |
| **WAV** | Wave definition and analysis parameters |

Apart from the GENERAL class, the parameter names within each category usually begin with the class name as a prefix. For example, wave parameter names begin with 'WAV_', statistical parameter names begin with 'STAT_', etc. The WAV parameters conform to the standard IAHR definitions [2]. Thus, 'WAV_HM0' is the significant wave height based on spectral moment $m_0$, etc. Definitions of all of the standard parameters are maintained in the on-line Help library and approximately 100 parameters are currently defined.

The structure of the main data section stored in record 6 of a GEDAP file is defined by the file type parameter. There are two basic data file types designated **Vn** and **An**. **Vn** files are used to store n separate data vectors and **An** files are used to store n-dimensional arrays. For example, a V1 file contains one data vector, a V3 file contains three data vectors, an A2 file contains a two-dimensional array, etc. The three most commonly used file types are V1, V2 and A2.

Most GEDAP data files are type V1. A V1 file contains a single data vector $y(x)$ where $x$ is an equally spaced implicit variable such as time or frequency. $N$ discrete values of $y(x)$ are stored in a one dimensional array $Y(j)$ for $j = 1$ to $N$ in the main data section. $Y(j) = y(x)$ at $x = x(j)$ where $x(j) = x_1 + (j-1)\Delta x$. $x_1$ and $\Delta x$ are constants stored as header parameters X1 and DX.

V2 files are normally used to store a set of $N$ $(x, y)$ values which define a function $y(x)$ tabulated at $N$ values of $x$ which are not equally spaced. The $x$ and $y$ values are stored explicitly in vectors $X(j)$ and $Y(j)$.

A2 files are used to store a two-dimensional array $A(x, y)$ where $x$ and $y$ are equally spaced implicit variables. The array values are stored in the main data section in a 2D array $A(i, j)$ for $i = 1$ to $N$ and $j = 1$ to $M$. $A(i, j) = A(x, y)$ at $x = x(i)$ and $y = y(j)$ where $x(i) = x_1 + (i-1)\Delta x$ and $y(j) = y_1 + (j-1)\Delta y$. $N$, $M$, $x_1$, $\Delta x$, $y_1$ and $\Delta y$ are stored as header parameters. For example, an A2 file could be used to store a directional wave spectrum tabulated at $N$ frequencies and $M$ wave angles. In this case, $A(x, y) = S(f, \theta)$, $x = f$ and $y = \theta$.

# 5 Running GEDAP Programs

GEDAP programs can be run in three different modes:

- Interactive Mode
- Command Procedure Mode
- Batch Mode

Each GEDAP program has several control parameters which must be specified by the user. In Interactive mode, these parameters are entered via an interactive dialogue on the terminal in which the program prompts the user to enter each required value. In the other two modes, the control parameters are obtained from a command procedure file which is prepared separately with a text editor prior to running the GEDAP program. The best mode to use depends on the nature of the particular job being done.

The Interactive mode is the easiest to use and it is also the best mode for learning how to run GEDAP programs. It is well suited to small analysis jobs such as checking the quality of measured data. It is also very useful for experimenting and for planning large scale data analysis procedures. It is the most flexible mode and allows results to be easily examined either graphically or numerically at each stage of the analysis process.

The Command Procedure mode is best suited to analysis tasks of moderate size in which several GEDAP programs must be run on multiple sets of data files. It is faster than the Interactive mode for routine analysis tasks because a single command procedure can be designed which runs several different GEDAP programs in succession in order to perform a complete analysis job.

Batch mode is simply an extension of the Command Procedure mode in which the analysis job is run in the background on a VMS batch queue. Batch mode is best suited to large scale analysis jobs which require more time to complete. The main advantage of batch mode is that the user's terminal can be used for other work while analysis jobs run in the background.

## 5.1 Interactive Mode

The data flow in a typical GEDAP program is shown in Figure 3. The program reads in data from one or more GEDAP input files. It then processes the data in some manner and stores the results in one or more GEDAP output files. The names of the input and output files and any other parameters required to control the data processing operation are obtained by means of an interactive dialogue with the user. This interactive dialogue is also referred to as conversational I/O.

A program is run in interactive mode by simply typing its name. The program will then prompt the user to enter each of the required control parameters. All GEDAP programs use a standard procedure for conversational I/O so that the same rules will apply for user input regardless of which program is being run. Numerical values are entered in free format and decimal points are optional. Default values may be selected by pressing RETURN in response to a prompt. If a default value is available, it is always shown in square brackets at the end of the prompt message. The prompt also specifies the required physical units.

Both interactive dialogue and menu-based schemes were considered when designing the user interface for GEDAP. The menu approach has an advantage when the number of control parameters is large and most of them are left at default values. Menus would not be an improvement for most GEDAP programs since the number of control parameters is usually less than 8. The interactive dialogue also acts like a checklist and thus encourages users to think a little bit more about what they are actually doing. The major reason for choosing

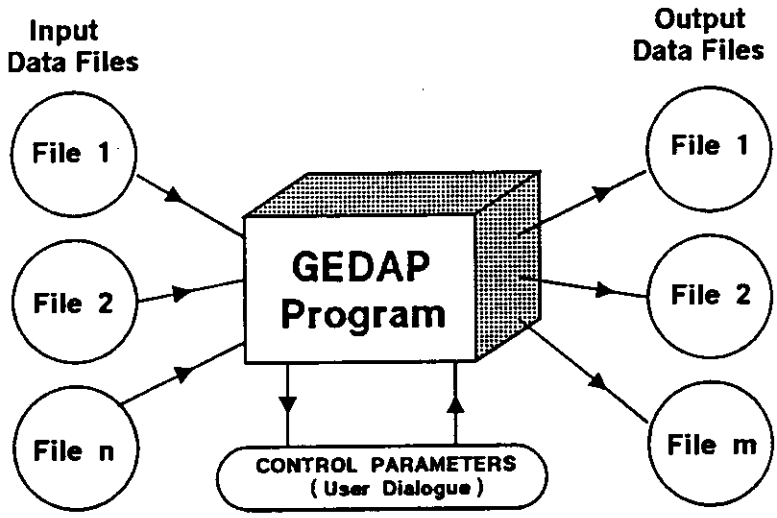


Figure 3: Data flow in a GEDAP program

the dialogue option, however, was to provide compatibility with the command procedure and batch modes which are used for most production work.

All GEDAP programs also use standard procedures for error handling. Some errors are recoverable. For example, if an invalid entry is detected in interactive mode, a warning message is displayed and the prompt is repeated. If a non-recoverable error occurs, a message such as the following is displayed and the program stops.

```
**************************************************
**  Fatal Error in Subroutine WRITE_GEDAP_V2      **
**  Too many points in arrays X and Y.  N = 45000 **
**  which exceeds the maximum limit of 32768      **
**************************************************
```

GEDAP error messages attempt to describe the problem as clearly as possible in plain English; no obscure error codes are used.

## 5.2 Command Procedure and Batch Modes

Command procedures are a convenient way to run a GEDAP program on several sets of input data in cases where most of the control parameters remain the same for all of the input data files. They are also very useful for defining composite analysis procedures in which several different GEDAP programs are run in succession. When a GEDAP program is run from a command procedure, the procedure file must include the command to run the program followed by all of the conversational input data required to control its operation. This is a trivial matter, however, because the required command procedure file can be generated

automatically by simply running the GEDAP program once in interactive mode. The following example shows a typical procedure file named C1.COM which was generated by running program RWSYN:

```
$ RWSYN
1          ! number of program cycles [1]
40         ! model scale factor [1.0]
20         ! wave train recycling period in full-scale minutes
1          ! seed option [1]
1          ! wave synthesis option [1]
SPEC1      ! name of wave spectrum input file [.001]
WAVE1      ! name of first wave record output file [.001]
```

The generated procedure file contains all of the required input values for RWSYN and each value is followed by a comment (starting at the special character !) which defines the particular input variable. C1.COM can therefore be easily edited in order to run program RWSYN with different input values.

GEDAP programs also allow conversational input values to be specified using symbols instead of fixed values. This feature allows a command procedure to be written with input parameters which can be passed into the GEDAP programs which are being run by the procedure. These symbols can be used to specify either numerical values or file names. Logical names can also be used to specify files.

The use of symbols and logical names allows general purpose command procedures to be written for various jobs. Each command procedure may run several different GEDAP programs. Procedure files can also be nested and used in a similar manner to subroutines. Thus, command procedures can be designed to perform comprehensive data analysis tasks by using GEDAP programs and other procedure files as building blocks.

Batch mode is almost identical to the command procedure mode. The only difference is that the command procedure is submitted to a VMS batch queue instead of being executed directly on the user's terminal. This allows other work to be done on the terminal while the command procedure runs in the background.

## 6 Plotting GEDAP Data Files

GEDAP contains an interactive graphics program called GPLOT which is used for plotting data files. GPLOT is a comprehensive two-dimensional plotting package which can handle both type V1 and type V2 files. GPLOT is based on the international GKS standard for computer graphics and consequently supports many different devices including VAX workstations, graphics terminals, laser printers and pen plotters. IBM PC's and compatibles are also supported via Tektronix 4014 terminal emulation software. This section contains only a very brief overview of GPLOT; a full description may be found in [3].

GPLOT is a command-driven program which can run in either interactive or batch mode. Interactive mode is used mainly for checking intermediate results and for small plotting jobs.

Batch mode is used primarily for large production jobs. The GPLOT command files required for batch jobs can be generated automatically while running in interactive mode. This greatly simplifies the task of designing plot layouts for large production runs.

Although it provides a broad range of functions, GPLOT is very easy to use. For example, a GEDAP data file named ABC.001 can be plotted by entering the single command `PLOT ABC`. This will produce a fully labelled plot including x and y data descriptions and physical units using information obtained from the data file header. The plotting scales will be set automatically according to the range of the data.

Some of the features provided by GPLOT are as follows:

- 12 different line patterns for drawing curves. Line widths can also be varied.
- 18 different symbols for plotting discrete data points.
- 23 character fonts including Greek characters. Text strings may be drawn at any angle and any size.
- Dynamic objects such as text strings, boxes and arrows which can be moved around with the mouse when designing plots.
- Choice of full grids or tic marks.
- Automatic generation of plot legends to identify the different symbols or line patterns used.
- An option whereby all commands entered in interactive mode are saved in an output file for subsequent use as a GPLOT command procedure for batch jobs.
- Convenient commands to label plots with parameters retrieved by name from data file headers.
- On-line Help facility for all GPLOT commands

Several of these features are illustrated in the example plot shown in Figure 4.

Figure 5 shows a typical wave analysis production plot generated by a standard GPLOT command file. In this case, one wave record file and two wave spectrum files are plotted in the same frame. All of the information shown on this plot has been retrieved from the data file headers. For example, the TPD value was printed with the following command:

```
PRINT 5.0, 4.0
TPD = {WAV_TPD} sec
```

The first line defines the position at which the text string in the second line is to be printed. GPLOT retrieves the numerical value of parameter `WAV_TPD` from the header and substitutes it for the expression `{WAV_TPD}` before printing the text string. In this case, the resulting text string is `TPD = 1.821 sec`. Formatting is automatic unless specified otherwise. It is thus very easy to include any desired header parameters on a plot.

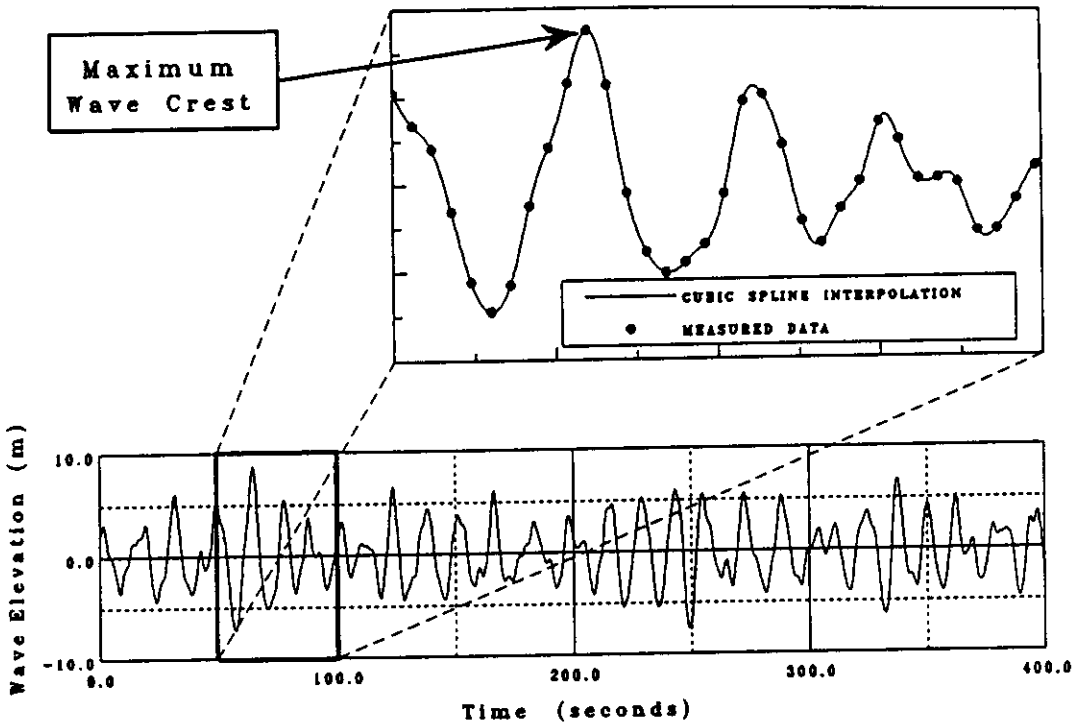


Figure 4: Example of some GPLOT plotting features

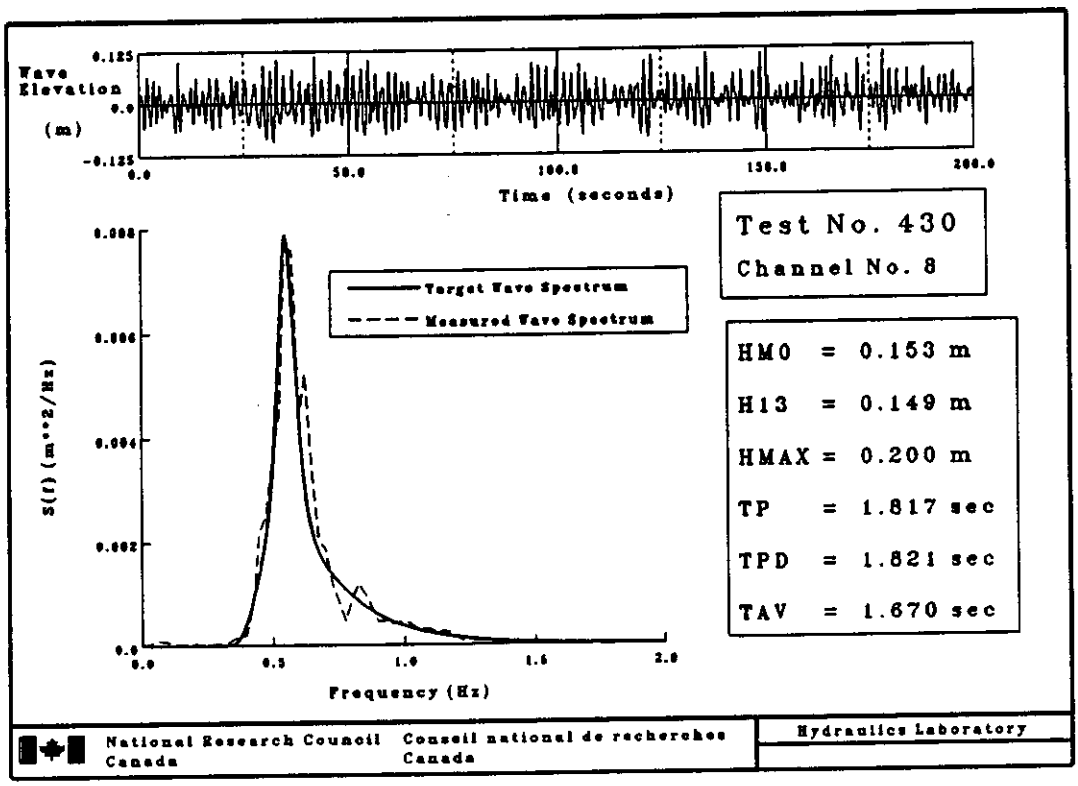


Figure 5: Typical wave analysis plot

# 7 Writing GEDAP Programs

Most data analysis tasks can be performed by using the standard set of GEDAP programs but custom programs will always be required for special applications. GEDAP has been carefully designed as a modular package so that both new standard programs and custom programs for a particular project can be written with minimum effort. The primary difference between a GEDAP program and any other Fortran program is that all of the I/O is handled exclusively by calling support subroutines.

A typical GEDAP program is organized as follows:

- Define constants, variables and arrays
- Open GEDAP
- Perform conversational I/O to obtain the control parameters
- Read the input data file(s)
- Perform data analysis functions
- Write the output data file(s)
- Close GEDAP

Each of the GEDAP-related functions can be accomplished by a single call to a support subroutine. Several program templates are also available to further simplify the writing of custom programs. These templates are skeleton Fortran source files which include all of the GEDAP subroutine calls required to implement the basic program structure shown above.

The GEDAP system library contains several subroutines to handle conversational I/O. Each of these subroutines has a name beginning with the prefix `PAR_` which stands for Prompt-And-Read. Each of these routines displays a prompt message and then returns the response entered by the user. For example, `PAR_REAL` reads a Real*4 input value, `PAR_INTEGER` reads an Integer*4 input value, `PAR_CHAR` reads in a character string, etc. The following example of a call to `PAR_REAL` will display the prompt message `Enter water depth (m):` and then store the user's response in variable DEPTH:

```
CALL PAR_REAL ( 'water depth (m)', DEPTH )
```

Exclusive use of the `PAR_` subroutines provides the following benefits:

- A consistent user interface is provided for all conversational I/O.
- The number of program statements required to implement the user dialogue is reduced by a factor of 8.
- Error handling will function properly in interactive, command procedure and batch modes.

- The procedure files required for batch mode operation will be generated automatically in interactive mode in a manner which is completely transparent to the programmer.

Subroutines are also provided to read and write GEDAP data files. The following example reads a type V1 file and stores the data vector $y(x)$ in array Y:

```
CALL READ_GEDAP_V1 ( 1, N_max, N, DX, X1, Y,
+                    DATA_X, UNITS_X, DATA_Y, UNITS_Y )
```

N is the number of points in array Y and DX and X1 are constants defining the implicit variable x. The last four arguments are Character*16 variables which define the x and y variables and their physical units.

A separate subroutine is used to retrieve any required B-Header parameters by name. For example,

```
CALL GET_REAL_PARAMETER ( 'WAV_FPD', FPD, FOUND )
```

will retrieve a Real*4 header parameter named WAV_FPD and store its value in variable FPD. The argument FOUND is a logical variable which indicates whether or not the requested parameter existed in the input data file.

In addition to the I/O routines, the GEDAP package includes extensive subroutine libraries to perform many other functions. These include digital signal processing, Fourier transforms (FFT), spectral analysis, correlation, linear filtering, interpolation, integration, differentiation, peak detection, sorting, nonlinear regression, statistical analysis and both linear and nonlinear wave mechanics.

## 8 References


1. Crookshank, N.L., *Experiment Control and Data Acquisition Systems at the Hydraulics Laboratory of the National Research Council of Canada*, IAHR Workshop on Instrumentation for Hydraulics laboratories, Canada Centre for Inland Waters, Burlington, Ontario, August, 1989.

2. Darras, M., *IAHR List of Sea State Parameters: A Presentation*, IAHR Seminar on Wave Analysis and Generation in Laboratory Basins, Lausanne, Switzerland, September, 1987, pp. 11–73.

3. Miles, M.D., *User Guide for the GPLOT-GKS Plotting Package*, National Research Council of Canada, Division of Mechanical Engineering, Laboratory Memorandum LM-HY-023, 1988.