DEVELOPMENT OF A MICROCOMPUTER BASED SYSTEM FOR PROCESSING OF GROUNDWATER LEVEL DATA

A Thesis

by

NIAZ MOHAMMED

Submitted to the Department of Water Resources Engineering, Bangladesh University of Engineering and Technology, Dhaka, in partial fulfilment of the requirements for the degree

of

MASTER OF SCIENCE IN ENGINEERING (WATER RESOURCES)

January, 1985.

BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF WATER RESOURCES ENGINEERING

January 10,1985

WE HEREBY RECOMMEND THAT THE THESIS PREPARED BY NIAZ MOHAMMED ENTITLED DEVELOPMENT OF A MICROCOMPUTER BASED SYSTEM FOR PROCESSING OF GROUNDWATER LEVEL DATA BE ACCEPTED AS FULFILLING THIS PART OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN ENGINEERING (WATER RESOURCES).

Chairman of the Committee (Dr. J.U. Chowdhury)

Member (Dr. A. Hannan)

m. K. Alam
Member (Dr. M.K. Alam)

SAWasimi
Member (Dr. S.A. Wasimi)

Member (Mir Lutfor Rasul Munsi)

Head of the Department (Dr. A. Hannan)

ABSTRACT

A microcomputer based Data Processing System has been developed. This comprises a Data Base Management System (DBMS) and an application program capable of performing some primary analyses of the data including missing data estimation. The system can store, retrieve,and list weekly groundwater level data and also important characteristics of observation wells. The principal functions of the application program are determination of highest and lowest water levels in a year, computation of annual mean and standard deviation, and assessment of significant harmonics. The system routines are written in FORTRAN. A combination of hierarchical and pointer structure has been used in the Data Base whose basic structural unit is the station-year record. With a view to providing easy access to the system a user's guide has been written.

A real-world application of the system has been done by processing groundwater level data from Mymensingh-Jamalpur area. Weekly data for five years (1979-1983) from ten observation wells have been selected. These data have been analysed with the system. Two methods of missing data estimation, namely: spline-fit interpolation and weighted average interpolation, have been compared. Accuracy of the former method has been found greater. Four methods of determining significant harmonics have been investigated and the method based on plotting of cummulative explained variance against the number of harmonics has been found to be most suitable.

ACKNOWLEDGEMENT

The author gratefully acknowledge his profound gratitude and indebtness to his supervisor. Dr.Jahir Uddin Chowdhury, Associate Professor, Institute of Flood Control and Drainage Research, Bangladesh University of Engineering and Technology for his help, encouragement, guidance and cooperation at all stages of the study. His active interest in this topic and valuable advice throughout the study were of immense help.

The author wishes to express his warm gratitude to other members of his thesis committee; Dr.A.Hannan, Professor and Head, Department of Water Resources Engineering, Dr. M.K.Alam, Associate Professor, Department of Water Resources Engineering, Dr.Saleh Ahmed Wasimi, Assistant Professor, Department of Water Resources Engineering, and Mr.Mir Lutfor Rasul Munsi, Chief Engineer, Hydrology, Bangladesh Water Development Board, for the review of the manuscript and for making valuable comments and suggestions.

The author is beholden to Water Resources Centee, Department of Water Resources Engineering, for providing Microcomputer facility, without which this work would not have been materialized.

The author acknowledges the Cooperation of Bangladesh Water Development Board in data collection for this study.

Finally the author wishes to thank Mr.M.Mofser Ali and Mr.A.K.Azad for their help in typing and drafting the report respectively.

N.M.

TABLE OF CONTENTS

TABLE OF CONTENTS (Contd.)

TABLE OF CONTENTS(Contd.)

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

Notation

A_0, A_i, B_i	Fourier coefficients
C_i	Amplitude of i-th harmonics
D_t	Periodic component in t-th week
L	Total number of surrounding station
m	Number of harmonics
X_t	Groundwater level data of t-th week of a year
S	Station at which data is missing
S_1, S_2, S_L	Surrounding stations
t	time in week
$\bar{X}$	Mean level
Y	Groundwater level data
ϵ_t	Stochastic component in t-th week
ϕ_i	Phase of i-th harmonics

INTRODUCTON

Chapter - I

INTRODUCTION

One of the major assets of any nation is the wealth of data of its natural resources. This is particularly true of data in water resources. The amount of water resources data is usually enormous. Efficient use of this vast amount of information is the key to successful planning of water development projects. A good storage and retrieval system or in other words a Data Base is necessary for the purpose.

A good storage and retrieval system should serve several purposes. Interrelated data should be stored together without harmful or unnecessary redundancy. There should be sufficient safe-guard against the stored data. The system should be able to serve a number of users simultaneously. Retrieval of the data should be easier for the users. There should be provision for continuous modification or updating of the data. The traditional computer-based data banks and file management systems can not meet all of these needs. Researches on this aspect led to the development of so called Data Base Management System (DBMS). It is a package of Computer programs and documentation that allows the users to set up or use a Data Base.

Groundwater is an important water resource of Bangladesh. Development and management of groundwater resources involves extensive processing and analysing of groundwater level data. Groundwater level changes with time and it is a stochastic process having annual periodicity. Bangladesh Water Development

Board (BWDB) maintains a large network of observation wells throughout the country to measure groundwater level at 7-day interval. As a result enormous amount of data is piling up every week. It is impossible to handle this vast information manually and as a result maximum utilization of the whole data is not achieved.

Recently BWDB has set up microcomputer-based data bank. However, it is a very simple storage and retrieval system which consumes large Computer storage. Redundancy and protection of the data has not been considered. Necessity of setting up a DBMS has already been stressed (1). Present study has been aimed in this direction. Considering pragmatic advantages, it has been decided that Microcomputer would be used to develop the DBMS.

Use of groundwater level data in mathematical model studies sometimes requires some preliminary analyses, e.g., determination of statistical parameters, performing harmonic analysis, etc. Estimation of missing data is also a necessary part in any planning and design studies. It has been decided, for greater benefit, that facility for those analyses would be incorporated in the data processing system.

Therefore basic objectives of the present study are

- to implement a microcomputer-based Data Base Management System for processing of weekly groundwater level data;

- to include an application program for missing data estimation, computation of statistical parameters and determination of significant harmonics: and

- to test the system using groundwater level data from Mymensingh-Jamalpur Area.

DATA BASE

Chapter - II

DATA BASE

2.1.0 INTRODUCTION

In this chapter a brief description of Data Base and Data Base Management System is given. Objectives of DBMS are discussed. Some existing DBMS packages in mainframes and microcomputers are mentioned. Data Banks of BWDB are discussed. DBMS program of World Meteorological Organization (WMO) has been evaluated and selected for modification for the present purpose.

2.2.0 DATA BASE MANAGEMENT SYSTEM

2.2.1 What is a Data Base ?

The word "Data Base" has been defined by many authors. Two famous definitions are given below:

i. A Data Base is a collection of interrelated data stored together with controlled redandancy to serve one or more application; the data are stored so that they are independant of the programs which use the data: a common controlled approach is used in adding new data and in modifying and retrieving existing data within a Data Base(2,3). A system is said to contain a collection of Data Base if they are disjoint in structure.

ii. A Data Base consists of all the record occurrences, set occurrences and areas which are controlled by a specific schema. If an installment has multiple Data Bases, there must be a separate schema for each Data Base. Furthermore, the content of different Data Bases is assumed to be disjoint (4).

The main drawback of earlier data processing method is that separate file(s) is created for each application of the same data records. This shortcoming is not present in the Data Base where a common file serve the purpose of multiple application. Hence, a Data Base is the repository of information needed for running certain functions in a body such as comporation, factory, university, or government department. Such a Data Base permits not only the retrieval of data but also continuous modification of data needed for control of operations.

2.2.2 Data Bank Vs. Data Base

A data bank is a collection of data, complete in some sense for the purpose of a particular job. It closely resembles to a tradional file. Each file is created for a single or group or accessing programs. With a Data Base variable amount of data can be used by the same program without recompiling it.

A data bank cannot serve more than one application at the same time, conversely, the intention of a Data Base is to allow the same collection of data to serve multiple application concurrently.

To use data directly from data bank, users must know the exact format of data storage, while to use data from Data Base, users need not know the format of data storage.

Data banks serves simply and solely as a storage and retrieval system without generally giving any consideration of redundancy, data protection,etc., which are the prime factors considered in a Data Base.

2.2.3 What is a Data Base Management System ?

A Data Base Management System is a package of computer programs and documentation required for setting up and using a Data Base. The functions of DBMS program is to insert data in Data Base, to modify and delete existing data in Data Base (in disk), to transfer data from Data Base to application program.

2.2.4 Objectives of DBMS

The benefits and objectives for examining and evaluating a DBMS software are summarized below:

i. Present an accurate model of the data.

ii. Organize data with simplicity.

iii. Provides timely response to queries.

iv. Reduces cost of data management. Disk storage is getting cheaper, and programmers are getting more expensive. A DBMS package reduces the need of both. A DBMS stores data in nonreduntant and compressed form thereby saving disk storage. Another important saving is programming time and effort. With a DBMS solution to simple problems take hours rather than days (5). Complex applications requires one programmer instead of a team (5).

v. Uses only nonredundant data. A data base has sometimes been defined as a nonredundant collection of data items, but in reality some measures of redundancy often exists in order to give improved access time or simpler addressing methods. So in reality a controlled redundancy is applied in a well designed data base.

Uncontrolled redundancy has several disadvantages. First, there is the extra cost of storing multiple copies. Second, and much more serious is that multiple updating operations are necessary to update at least some of the redundant copies. Redundancy is therefore, expensive on files with a large volume of updating or, worse, in which new items are frequently being inserted or deleted. Third, because different copies of the data may be in different stages of up-dating, the system may give inconsistent information.

vi. Safeguards data integrity. A Data Base is often composed of a number of files, records, data items, and interconnections. If something goes wrong - a power failure, for instance a part of the Data Base may be unreadable. The DBMS should check itself and immediately let the user know if it finds something wrong with the system.

vii. Permits access by users. A DBMS user should be able to ask a wide variety of questions about the data. Most Data Base Management System have a query language for asking these questions. This language should be simple and easy to learn.

viii. Provides data protection. The proper protection of data stored in computer system has becomes a deep concern to many people in the last decades and as such considerable amount of work has been done to improve the Data Base from this direction (6,7,8).Protection includes data security (Protection of data against accidentaland deliberate disclosure to unauthorized persons or unauthorized modifications or destruction). Privacy (right of individuals or organization to determine for themselves, when, how, and to what extent information about them is to be transmitted to others).

ix. Provides fail-safe operation. A good DBMS should have the facility to recover data due to accidential loss by power failure or similar other reasons.

x. Interface with the past. A DBMS should be able to accept data from conventional files, and it should be able to output data in file format for use by existing non-data base programs.

xi. Interface with the future. Without a DBMS, even the most trivial changes can require rewriting a lot of programs. Once the rewriting is done,new bugs appear; sometimes causing downtime and lost data. Data Base management systems can make change less painful. Often users can rearrange or add data elements without touching any programs except those which use the new elements.

xii. Permits shared usage of data.

xiii. Offers language flexibility.In addition to the high-level query language, a good DBMS should be capable of interfacing to lower level language like COBOL, PL/1, FORTRAN, BASIC etc.

2.3.0 SOME DBMS SOFTWARES

2.3.1 Mainframes

The number of computer installation using DBMS packages is increasing rapidly over the past years. The national Computer Centre Ltd.(NCC) of U.K. has evaluated experiences with a wide range of these sytems in over forty organization both in the U.K. and in the U.S.A. Six major mainframe-based DBMS package (ADABAS, IDMS, IMS, (DL/1), ROBOT, SYSTEM 2000 TOTAL) were discussed by NCC (9). These packages differs in their package environment, design facilities, data manipulation facilities and performance issues.

All the packages can be installed in an IBM 360/370 mainframe computer except ROBOT which can be installed in a ICL-1900, 2903, 1904 or UNIVAC Computers.

2.3.2 Microcomputers

With the advent of hard disk for microcomputers, microcomputer manufacturers have released network and multi-user operating system and terminal facilities. Some microcomputer based DBMS meet most of the objectives set for a large computer system. Such microcomputer based DBMS is preferable to a main-frame-based one, because microcomputers are cheaper portable involving low maintenance cost.

David Kruglinški (5) has worked with several DBMS package on microcomputers. Some of the well-documented and widely used packages are Conodor Series 20, dBASE II, FMS-80, MDBS III, Peral Level 3 and Data Star. All these package run under CP/M operating system and are generally suitable for report generator, business and information processings.

Vituki of Hungury, on behalf of World Meteorological Organization (WMO) has developed at DBMS for primary processing of hydrological data. The system runs under a TRSDOS Version 2.0a operating system. A TRS-80 Model II or an enhanced model microcoumpter is required to install the package.

2.3.3 Data Banks of BWDB

Bangladesh Water Development Board (BWDB) has already started using Microcomputers for computerization of hydrological information and data. No DBMS package is yet installed but microcomputer-based data banks and File Management System (FMS) have been set up. A number of Data Banks that are already developed are described below:

Table 2.1 Main features of Data Banks of BWDB

Divisions of BWDB	Programming language	Type of Computer	Objective
Flood forcasting Division	FORTRAN	TRS-80 Model 16B	Storage and Retrieval of river stage data, rainfall data.
Surface Water Circle	BASIC & FORTRAN	TRS-80 Model 16B	Storage & retrieva of surface water data
Ground Water Circle	DERPG	TRS-80 Model 16B & 1BM 5281	Storage & retrieva of ground water level data.

Ground Water Circle uses Data Entry Report Program Ceneratur (DERPG) to store) groundwater level data or Observation well characteristics in traditional file. Floopy disk is used as the storage media. Usually no user is permitted to use the data directly from floopy disk in their programs, instead a printed output of the data is supplied to the user, from this printed output users again insert those data in their programs. This causes increased programming time and effort which can be removed by using Data Bases.

2.3.4 Selection of DBMS Software For Present Study

Once the decision to implement a DBMS package has been undertaken the choice of suitable software obviously depends on the type of hardware installed. Experience (9,10) indicates that in most case the choice is made for reasons given below:

i. Needs of a particular organization or company

ii. The size of the Data Base : Data compression, The ability to archive part of the Data Base.

iii. Hardware constraints

iv. Security features : Recovery, Access Control

v. Language interface : Host language supported; Self contained capabilities.

vi. Support : Design and maintenance requirement, Level available from the users.

vii. Cost.

viii. Ease of use

ix. Ability to adapt to change

x. Size of user base.

WMO DBMS software has been selected in this study and is modified for the following reasons:

i. It is a TRS-80 microcomputer based data processing system specially designed for primary processing of hydrological data.

ii. A TRS-80 Model 16B computer, which is now widely used in our country support multiuser operating system (TRS XENIX) and terminal facilities (upto 6 terminals).

iii. The system routines was written in a popular language FORTRAN which can be easily modified to adapt the system for a slightly different purpose.

iv. A TRS-80 Model 16B microcomputer is available in the Water Resources Engineering Department of BUET.

v. BWDB mainly uses TRS-80 microcomputer for processing of hydrological data.

2.4.0 DATA BASE MANAGEMENT SYSTEM OF WMO

2.4.1 Introduction to the Package

WMO DBMS is a TRS-80 microcomputer-based data processing system which can be used for primary processing of diverse hydrological data e.g., river stage data, precipitation data, discharge data.

Its source language is FORTRAN and it can interface to any program written in FORTRAN. The system was originally developed by VITUKI of Hungury for World Meteorological Organization (WMO). A detailed description of the DBMS is given in appendix-A. Hardware requirements, structure, data manipulation facilities of the system, a brief description of source program and modification necessary for the present purpose are described in the following articles.

2.4.2 Package Environment

The system can be installed in a TRS-80 Model II microcomputer. It is also compatible to other enhanced model, viz TRS-80 Model 12, TRS-80 Model 16, TRS-80 Model 16B. Operating system supporting the package is TRSDOS Version 2.0a or any enhanced operating system such as TRSDOS Version 2.0b, TRSDOS Version 4.10 etc.

A 8-bit CPU and a minimum of 64 K memory is required for the system. Single sided and double sided floopy disk as well as hard disk can be used as the storage media.

The system is not portable which means it will not be available to any other computer.

2.4.3 Structure of the DBMS

The following requirement have been identified for the structural development of the system.

i. rapid retrieval of unique record.

ii. ability to dandle records of variable length.

iii. ability to store data in most compact form.

Rapid retrieval is performed by the use of record identifier. A new combination of hierarchical and points structure has been developed by the designer of the DBMS.

2.4.4 Data Manipulation Facilities

The system analyst communicates with the Data Base with the following commands

*I	Insertion of new data record on to Data Base
*M	Modification of existing data record in Data Base files
*D	Deletion of existing data record from Data Base file
*C	Copying data record from one Data Base file to another
*L	Listing data record on Line printer
*P	Listing data record on the screen
*W	Writing new data record onto Data Base file
*E	Exit program with writing of data record from buffer.
*Q	Exit program without writing.

The application programmers or users communicate with the Data Base by compiling their FORTRAN program and linking those programs with the Data Base by the following command

```
L80  PROG-N,USER, MSCLIB-S, TRNLIB-S-E
```

where PROG is the run file saved in the disk

USER is the users' program

MSCLIB & TRNLIB are DBMS programs.

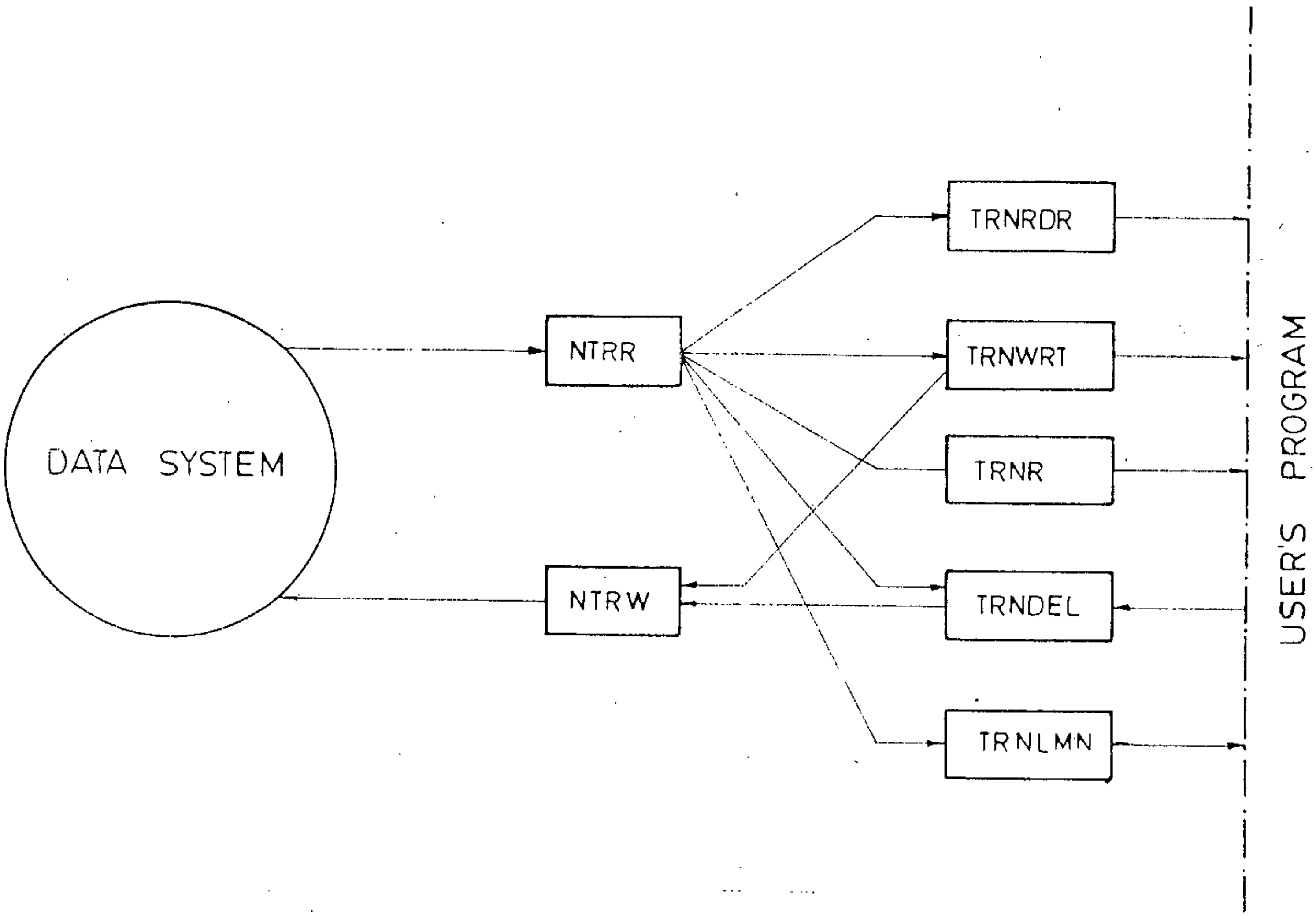
2.4.5 Brief Description of the Program

The whole data processing system consists of 13 programs each of which calls one or more of 107 subprograms. A number of subprograms in grouped in a library. The basis of grouping is the function they perform. The groups are described below:

TRNLIB : This group of subprogram is used to transfers data records from temporary computer storage to data base and vice versa. This group consists of 18 subroutines in particular TRNWRT, TRNR, TRNDEL, TREIMN, TRNERR, TRNHDR, LKLCHK, HKLCHK, KEYSRC, NTRW, NTRR, WSPDEC, WSPCMP, WSPSHF, WSPLBL, WSPKEY, STOKEY, RCLKEY. A user communicates with the Data Base through these subroutines as illustrated in the Fig.2.1.

MSCLIB : This library is composed of 32 micellaneous subprogram. Functions of these subprograms are conversion of data, compression of data, Reestablish data from compacted block, opening disk file, c checking dates etc. The names of subroutines are QFN, IHFN, RCINP, HDBDYR, DCP, DCPI, CMPI, DCPB, NREADK RREAD, HREAD, FOPEN, NPSHF, NDTCHK, MAYDY, MAXDYF, NTSH, SPLINE, FSPLN, QCD, JWRITE, NTSD, NDDF, NNREAD, NTCH, MEAN, RMEAN, ENCD, ENDCR, NENCDE, EDTERR.

MNTLIB : This library comprises 7 subprograms namely MNTINI, MNTDMP, MNTCPY, FLASH, MNTLST, which are used in maintaining Data Base file(s).



~~Fig.~~2.1 CONNECTION OF DATA SYSTEM AND PROGRAMS

STELIB : This library has 7 subprogram viz. STNTER,STNPAR, STNINP, STALST, STADEL, STAMOD, STAWRT. They are used in maintaining only those data Base files which contain basic characteristics of the stations.

STGLIB : This library possesses 13 subprogram namely RCUBIQ, STOBIQ, DCPW, CMPW, DLJW, STGCHK, INLIM, INDTE, INSTP, INSTN, ISTG, MSTG, WSTG, which are used in entering and editing water level data from Keyboard into the data base.

PRCLIB : The subroutines of this library number 14 and named as RCLBIQ, STOBIQ, DCPW, CMPW, DLTW, PRCCHK, INLIM, INDIE, INSTP, INSTN, IPRC, MPRC, WPRC. They are used in entering and editing precipitation data from keyboard into the data base.

PCELIB : The number of subprograms in this library run into 11 and are used in entering and editing rating curve from keyboard to Data Base. The subprograms are WDIR, RDIR, RCDINP, VLDCHK, STNINP, IRC, DRC, LRC, MRC,WRC.

LSTLIB : This library consists of 7 subroutnines viz. INLIM, LDSTN, LASTG, LDDCH, LADCH, LDPRC, LAPRC which are used in listin daily or annual water level data precipitation data and discharge data.

Functions of the programs are given in the following table.

Table 2.2 Purpose of programs of WMO Data Processing System and their calling subprograms (used libraries)

Name of the Program	Purpose of the program	Used libraries
HDBMNT	Initialization, Copying and Dumping of Data Base file(s)	MNTLIB, MSCLIB, TRNLIB
HDBCPY	Copying Data Base file(s)	MNTLIB,MSCUB, TRNLIB
STEDIT	Initialization and editing of basic characteristics of stations	STELIB, MSCLIB, TRNLIB
STAGES	Insertion and edition of water level data	STGLIB, MSCLIB, TRNLIB
PRECIP	Insertion and edition of Precepitation data	PRCLIB, MSCLIB TRNLIB
LSTSTG	Listing of yearly daily table of water level data	STLIB, MSCLIB, TRNLIB
LSTPRC	Listing of yearly daily table of precipitation data	LSTLIB, MSCLIB, TRNLIB
LSTDCD	Listing of yearly table of discharge	LSTLIB,MSCLIB, TRNLIB
LSTDCA	Listing of daily table of discharge	LSTLIB,MSCLIB, TRNLIB
HLPCMD	Helps explaining meanings of available commands of various program	NONE
TRNERR	Aids in detection of error by TRN Subprogram	NONE
TRNBLD	Initiali-ation of common blocks of TRN group of subprograms	NONE

2.4.6 Modification necessary for the Present Purpose

WMO data processing system has been developed for handling of river stage data, precipitation data and river discharge data. It does not have provision for handling of ground water level data. It has the facility to store data at four intervales, viz. 1 day, 60 minutes, 30 minutes and 15 minutes. Groundwater level at an observation well is measured at 7-day interval by BWDB. Hence it is necessary to modify the subprograms used for insertion modification and deletion so that they can handle 7-day interval ground water level data.

The station file management subprograms of WMO DBMS are employed to handle river station characteristics. Since basic characteristics of an ground water observation well differs co considerably, so to process observation well characteristics, the subprograms used for insertion, modification and deletion of station characteristics data requires modification.

Out of 13 programs listed in table 2.2, 5 programs namely PRECIP, LSTSTG, LTSPRC, LSTDCD, LSTDCA, are not included in the present data processing system. Accordingly none of the subprograms of groups PRCLIB, RCELIB, and ISTLIB are called by any program of the present system.

DEVELOPMENT OF GROUNDWATER LEVEL DATA PROCESSING SYSTEM

Chapter - III

DEVELOPMENT OF GROUNDWATER LEVEL DATA PROCESSING SYSTEM

3.1.0 INTRODUCTION

This chapter contains the development of the present groundwater level data processing system in the TRS-80 Model 16B microcomputer of BUET. The development is achieved in two phases. The first is the modification in station file management subprograms and water level file management subprograms of the WMO DBMS programs. The second phase is the development of an application program for performing few primary analysis of the data. A users guide has been prepared.

3.2.0 GROUNDWATER LEVEL DATA OF BWDB

BWDB maintains a large number of observation wells throughout the country. Every well is numbered by means of a one or two lettered prefix indicating the district in which it is located. For example, well number M- 25 refers to the well number 25 in the district of Mymensingh and TA-37 refers to the well number 37 in the district of Tangail. Depth of water table in a well is measured below some fixed measuring point (MP), whose height from the ground surface and above mean sea level (MSL) is known. The measurement is made in F.P.S. unit (feet) upto two digits after decimal point. Measurement is taken at 6 a.m. every monday.

There are two types of observation wells: dug well and piezometric well. Diameter of dug wells varies from 1.5 feet to 6.0 feet of which most of them are 9.75 to 48.00 feet deep. The piezometric well consists of 46 feet to 355 feet long G.I. blind pipe of 1½ inch diameter. Studies (11) reveals that the deep piezometric wells and dug wells measures diffirent piezometric levels and they cannot, therefore, be treated as one homogenous network. Surveys(11) in Bagmara Upazilla, Rajshahi, reported that in the same location piexometric wells measured readings more than one meter higher than those measured by dug wells.

3.3.0 INCLUSION OF FACILITY FOR MISSING DATA ESTIMATION

3.3.1 Methods of estimation

Usually some data remain missing from some station-year record. Estimation of these missing data is required for planning and design studies. Subprograms are developed and incorporated in the application program for estimation of missing data. Two methods have been used for estimating missing data. They are spline interpolation method and weighted-average method. The application program first test the possibility of using the spline interpolation method. If the answer is negative, then it goes for the weighted average method. The two methods are described in the following sections.

3.3.2 Spline-fit interpolation method

Spline-fit interpolation is adopted when at most 2 consecutive data are missing and at least 3 previous and 3 subsequent records are available. Detailed theory of spline function is given in (12). A brief description is given below:

If a function y = f(t) is given by m point (t_1,y_1), (t_2,y_2) ......(t_k, y_k)..... (t_m,y_m) arranged in order of increasing value values of t_i then spline-fit interpolation is accomplised by a polynomial such that the first and second derivative is continuous at each point. A third degree polynomial is selected in this study.

Let Z_1, Z_2 ....Z_k,....Z_m be the values of second derivative at the points. Then at any point (Fig. 3.1)$(t,\check{y})$ in the interval between two consecutive point (t_k,y_k) and (t_{k+1}, y_{k+1}), the second derivative is given by

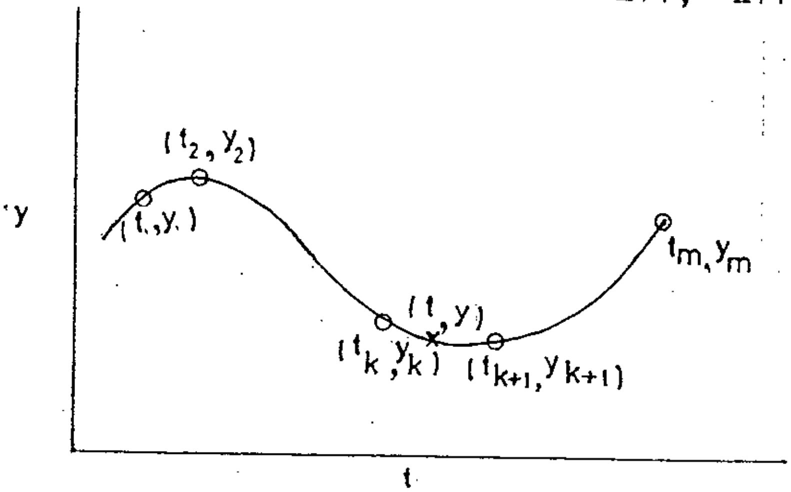


Fig. 3.1 Interpolation of two consecutive point by spline function.

$$y'' = Z_k \cdot \frac{t_{k+1} - t}{d_k} + Z_{k+1} \frac{t - t_k}{d_k} \quad (3.1)$$

where $d_k = t_{k+1} - t_k$

Integrating (3.1) for first derivative,

$$y' = - Z_k \left[(t_{k+1} - t)^2/2d_k \right] + Z_{k+1} \left[(t - t_k)^2/2d_k \right] + C_1 \quad (3.2)$$

Integrating again the equations of the spline-function becomes

$$y = Z_k \left[(t_{k+1} - t)^3/6\ d_k\right] + Z_{k+1} \left[(t - t_k)^3/\ 6\ d_k\right] + C_1\ t + C_2 \quad (3.3)$$

where C_1, C_2 are the constants of integration. They can be evaluated from the fact that the curve passes through the points (t_k, y_k) and (t_{k+1}, y_{k+1}).

From equation (3.3) after substitution.

$$y_k = (Z_k\ d_k^2/6) + C_1\ t_k + C_2$$

$$y_{k+1} = (Z_{k+1}\ d_k^2/6\ ) + C_1\ t_{k+1} + C_2$$

From which

$$C_1 = \left[(y_{k+1} - y_k)/d_k\right] - \left[(Z_{k+1} - Z_k)\ d_k/6\right] \quad (3.4)$$

$$C_2 = \left[(y_k\ t_{k+1} - y_{k+1}\ t_k)/\ d_k\right] - \left[Z_k\ t_{k+1} - Z_k\ t_k)\ d_k/6\right] \quad (3.5)$$

Substituting C_1 and C_2 in equation (3.3) equation of the spline function in the interval $(t_k - t_{k+1})$ becomes

$$y = \left[Z_k\ (t_{k+1} - t)^3/6\ d_k\right] + \left[Z_{k+1}(t - t_k)^3/6\ d_k\right] + \left[(t_{k+1} - t)\right.$$
$$\left.(y_k/d_k - Z_k d_k/6)\right] + \left[(t - t_k)\ (y_{k+1}/d_k - Z_{k+1}\ d_k/6)\right] \quad (3.6)$$

In this equation all variables are known except Z_k and Z_{k+1}; the values of the second derivative at the end points of the interval. One condition which can help to determine these values is that the slope at (t_k, y_k) as determined by equation (3.2) must be the same as that determined by the corresponding formula for the interval $(t_k - t_{k-1})$. When the values of C_1 from (3.4) is substituted in equation (3.2), the equation for the first derivative at the begining of the interval $(t_{k+1} - t_k)$ becomes

$$y' = -\left[Z_k\ (t_{k+1} - t)^2/2d_k\right] + \left[Z_{k+1}(t - t_k)^2/\ 2d_k\right] + \left[(Y_{k+1} - Y_k)/d_k\right]$$
$$- \left[(Z_{k+1} - Z_k)\ d_k/6\right] \quad (3.7)$$

Similary the relation at the end of the proceeding interval $(t_k - t_{k-1})$ is

$$y' = -\left[Z_k\ (t_k - t)^2/\ 2\ d_{k-1}\right] + \left[Z_k\ (t - t_{k-1})^2/2\ d_{k-1}\right]$$
$$+ \left[(y_k - y_{k-1})/d_{k-1}\right] - \left[(Z_k - Z_{k-1})\ d_{k-1}/6\right] \quad (3.8)$$

From conditions of spline function, equation (3.7) and (3.8) must be same and hence

$$y'_k = (-Z_k\ d_k/2) + \left[(y_{k+1} - y_k)/d_k\right] - \left[(Z_{k+1} - Z_k)\ d_k/6\right]$$
$$= (Z_k\ d_{k-1}/2) + \left[(y_k - y_{k-1})/d_{k-1}\right] - \left[(Z_k - Z_{k-1})\ d_{k-1}/6\right] (3.9)$$

An equation like this can also be obtained for each of the internal point , that is k = 2,3.....m-1. So there are (m-2) equation in m unknown second derivative $Z_1, Z_2 \ldots\ldots Z_m$. Two more conditions are imposed in order to determine these quantities completely. In the present study they are taken as: the third derivatives are continuous at the two points (t_2, y_2) and (t_{m-1}, y_{m-1}). Differenting equations (3.1) the third derivative becomes

$$y''' = -Z_k/d_k + Z_{k+1}/d_k \quad \ldots (3.10).$$

From the imposed conditions, values of y''' before and after the point 2 must be the same.

Hence $-Z_1/d_1 + Z_2/d_1 = -Z_2/d_2 + Z_3/d_2$ (3.11)

Similarly at point (m-1)

$$-Z_{m-2}/d_{m-2} + Z_{m-1}/d_{m-2} = -Z_{m-1}/d_{m-1} + Z_m/d_{m-1} \quad (3.12)$$

Equations (3.11) and (3.12) along with (m-2) equations of the type (3.9) constitute m equation in m unknown second derivative $z_1, z_2 \ldots . z_m$. In the present study spline fit interpolation is applied when 3 previous and 3 subsequent data are available. Hence m becomes 6. Total number of equation involving Z and C becomes 18, 2 from third derivative condition, 4 from first derivative condition and 12 from constants of integration C_1 and C_2. These equations for the Z_k are solved by Gauss-Jordan elimination method. When the values of Z_k and Z_{k+1} are substituted in equation (3.6), working equation for interpolation at the point t in the interval $t_k < t < t_{k+1}$ is obtained.

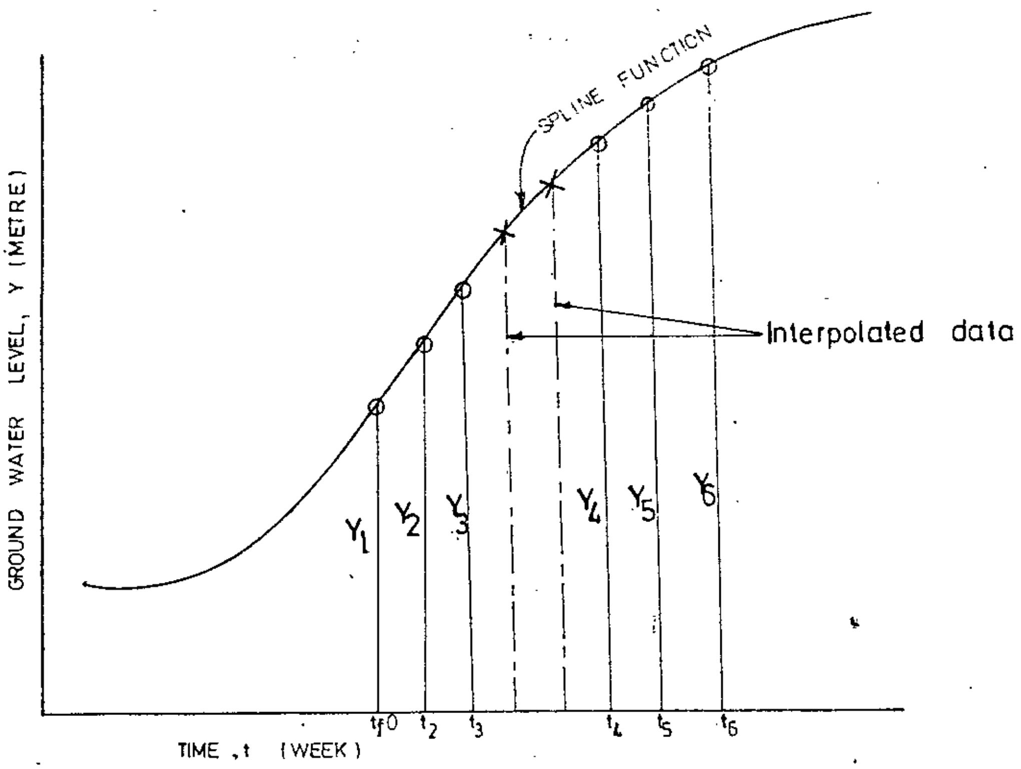


Fig. 3.2 Missing data estimation by spline-fit interpolation

In this case t represents the time in week and y represent corresponding groundwater level in metre. It has been decided at most 2 consecutive data will be interpolated by the spline function. Hence referring to Fig3.2 if n denotes the number of consecutive missing data where n is equal to either 1 or 2, then the points for passing the spline function are determined as follows: Shifting the groundwater level axis to a convenient position

$t_1 = 0$

$t_2 = 1$

$t_3 = 2$

If n = 1

$t_4 = 4$

$t_5 = 5$

$t_6 = 6$,

If n = 2

$t_4 = 5$

$t_5 = 6$

$t_6 = 7$

y_1, y_2, y_3, y_4, y_5, y_6 are the corresponding groundwater level. These values are determined by the computer and missing data between y_3 and y_4 are estimated. Listing of computer program is given in appendix - D.

This method has, however some limitations as given below

i. It is not suitable to estimate more than 2 consecutive missing data.

ii. If any of the 3 previous or 3 afterwards record is missing this method is not applicable.

iii. Application program can retrieve groundwater level data for a particular ~~record~~ station-year record. Hence if any of the first 3 or the last 3 data of a year is missing, this method cannot estimate that missing data.

3.3.3 Weighted-average method

This method is applicable for estimating any number of missing data at a station provided corresponding record (data measured at the same time) for the surrounding wells are present. Missing data at the station S is estimated as the weighted-average of the groundwater level at the surrounding observation wells S_1, S_2, ... S_L. Surrounding wells are selected satisfying the following conditions.

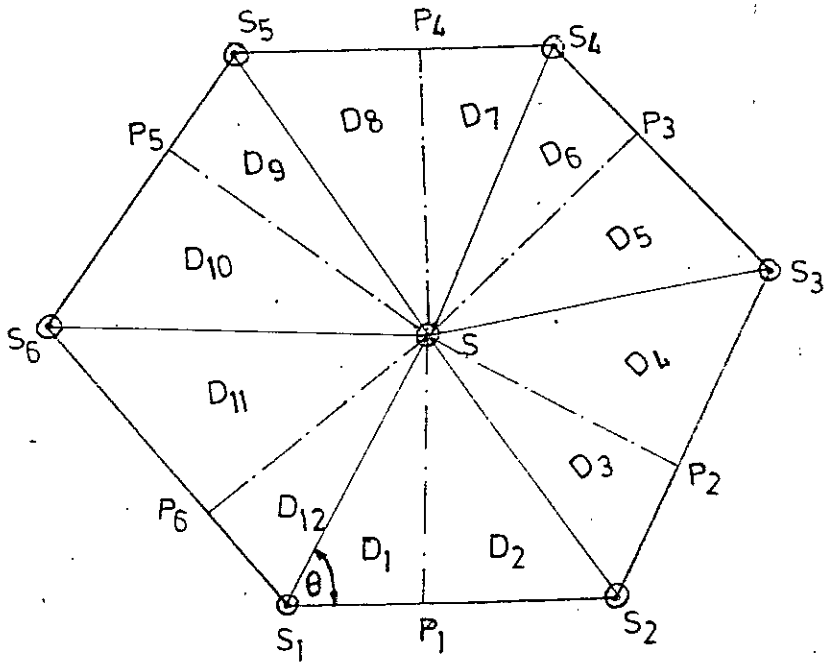


Fig. 3.3 Schematic diagram illustrating weighted-average method.

i. They must be same type of well (dug or piezometer) as the observation well S.

ii. Station S should lie inside the polygon $S_1 S_2 S_3 \ldots S_L$ formed by the sides (Fig. 3.4)

iii. $3 \leqslant L \leqslant 6$ where L is the total number of selected surrounding wells. Four possible configuration are shown in Fig.3.4

iv. Angles $\angle SS_1 S_2$, $\angle SS_2 S_1 \ldots \angle SS_{L-1} S_L$, $\angle SS_L S_{L-1}$ etc should be less than 90°.

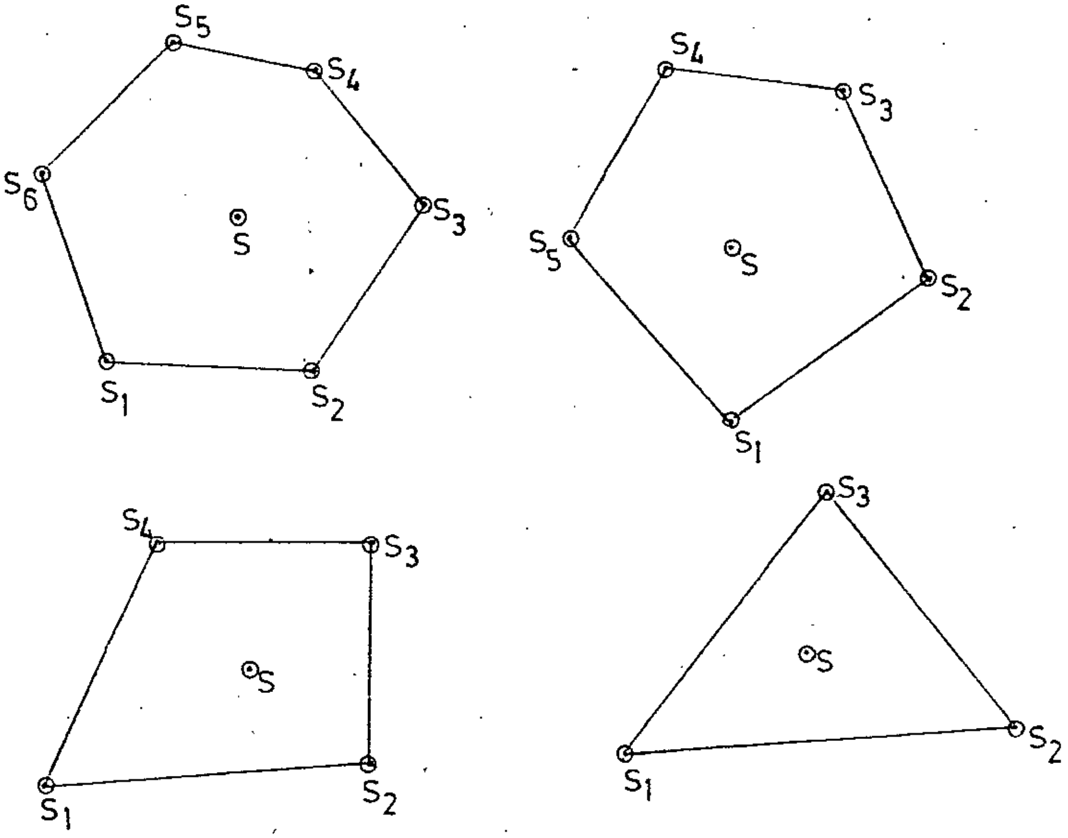


Fig. 3.4 Four possible configuration of surrounding wells

The working equation for estimating missing data is

$$X_i = \frac{\triangle A_1 \ H_{1,i} + \triangle A_2 \ H_{2,i} + \cdot \triangle A_j \ H_{j,i} \cdots + \triangle A_L \ H_{L,i}}{\sum_{j=1}^{L} \triangle A_j} \qquad (3.13)$$

where X_i = estimated groundwater level at ith week of a year at the station S

$H_{j,i}$ = measured data at surrounding well S_j at ith week of the same year.

$\frac{\triangle A_j}{\sum \triangle A_j}$ = are the weighted factor for station S_j which are computed as follows

Let D_1 = area $SS_1 \ P_1$. D_2 = area $SS_2 \ P_1$, D_3 = area $SS_2 \ P_3$ and so on. Where P_1, P_2,... P_L are the perpendiculars drawn from S to sides S_1S_2, S_2S_3, ...$S_{L-1}S_L$ respectively.

Refering to Fig.3.3.

$$\mathrm{Cos}(Q) = \frac{(SS_1^2 + S_1S_2^2 - SS_2^2)}{2 \ SS_1 \ S_1S_2}$$

$$\mathrm{Sin}(Q) = 1 - \mathrm{Cos}^2(Q)$$

$$D_1 = \frac{SS_1 \ \mathrm{Cos}(\theta) \ SS_1 \ \mathrm{Sin}(\theta)}{2}$$

$$D_2 = \frac{SS_1 \ \mathrm{Sin}(\theta) \ (S_1S_2 - SS_1 \ \mathrm{Cos}\ (\theta))}{2}$$

Similarly D_3, D_4 ...D_{2L} can be computed.

Now

$$A_1 = D_1 + D_{2L}$$

$$A_2 = D_2 + D_3$$

and so on.

In the computer subprogram, station code of all the stations for which missing data estimated by using weighted-average method is stored in a one dimensional array. The station codes of the corresponding surrounding wells are stored in a two-dimensional away. The sides SS_1, $SS_2 \ldots SS_L$, S_1S_2, $S_2S_3 \ldots S_{L-1}S_L$ for each station are stored in another two dimensional array. The computer first read in the data for the station S and store it in the one dimensional array X_i. Corresponding data of the surrounding wells are then retrived and stored in a two dimensional array $H_{j,i}$. Weighted-factor are then computed and missing data are estimated.

3.3.4 Accuracy

Effectiveness of the two methods of missing data estimation is tested with the data processing system. Sample results are given in appendix - B.

In order to compare the accuracy of the two methods, on missing data estimation a few data at observation well M-47 were assumed as missing. Then they were separately estimated by the two methods. Results are shown in Table 3.1. It indicates that spline function is more reliable than weighted average method for missing data estimation.

Table 3.1 Comparison of accuracy of two methods of missing data estimation

Observation Well Number : M-47 Year : 1979

Date of observation	Observed data (m)	Spline-fit interpolation		Weighted-average method	
		Estimated data (m)	Error (% of range)	Estimated data (m)	Error (% of range)
5.3.79	5.088	5.125	+.89	5.243	+3.74
12.3.79	4.988	5.041	+1.28	5.186	+4.78

3.4.0 INCLUSION OF SOME PRIMARY ANALYSIS

3.4.1 Computation of statistical parameters

Statistical parameters such as highest and lowest groundwater levels in a year, range of fluctuation in a year etc. are useful information. Provision for determining these parameters is included in the present data processing system. Data in the Data Base are stored w.r.t. a measuring point (MP) at station. Whose reduced level (RL) and height from ground surface (GL) are also stored. After retrieval, the groundwater level data are converted w.r.t. mean sea level (MSL). Statistical parameters of groundwater level data at a station in year are as follows:

i. Highest water level above MSL

ii. Lowest water level above MSL

iii. Mean level

iv. Depth of lowest level below ground surface

v. Range of fluctuation

vi. Standard deviation

vii. Variance

Let X_i, $t = 1,2..N$ denotes a station-year record, XMAX, XMIN being the highest and the lowest groundwater level measured above MSL. Let ELMPSL and ELMPGL be the elevation of the measuring point from MSL and GL respectively.

Depth of lowest level below ground surface,

XMINGL = ELMPSL - ELMPGL - XMIN

The range of fluctuation, RANGE is given by

RANGE = XMAX - XMIN

The mean level,

$$\text{XMEAN} = \frac{1}{N}\sum_{i=1}^{N} X_i$$

Where i is the week number, N is the total number of data in a year.

The standard deviation

$$SD_x = \frac{1}{N}\sum_{i=1}^{N} (X_i - \text{XMEAN})^2$$

Variance,

$$VAR_x = SD_x^2$$

All the above parameters are processed by the computer. Sample results are given in Appendix- B.

3.5.0 MODIFICATIONS IN THE WMO DBMS PROGRAM

3.5.1 General

The present data processing system is a modification of WMO DBMS and equipped with an application program discussed in article 3.6.0. It was already mentioned in article 2.4.6 that

WMO Data Base Management System cannot be directly used for handling of weekly groundwater level data and data relating to observation well characteristics. The modification that is necessary for this purpose was also outlined in that article. A detailed description of the modifications done is given in the following sections.

3.5.2 Modifications in station-file management subprograms

Program STEDIT (of WMO DBMS) can be used to insert, modify, delete and list data relating to river station e.g., station code, name of the river, size of catchment area, distance from the mouth etc. A total of 11 characteristics can be handled by the program. Groundwater observation well characteristics differs considerably from those of river station and accordingly the program is modified so that it can handle following 11 characteristics of observation well.

i. Station code: Station code must consists of three numeric digits indicating the observation well number in a particular district. Thus for this system, station code lies between 001 and 999.

ii. Observation well no.: Observation well no. can be assigned any data consisting of 20 alphanumeric characters including special character and blank. In the present case it indicates the district in which the well is located and the serial number of the observations well in that district. For example Mymensingh-25 stands for observation well number 25 in Mymensingh district.

iii. Location (Village): This can be assigned 20 alphanumeric characters (including blank, the special character). In the present case it indicate the name of the village in which the well is located.

iv. Old station code : Old station code can be assigned a data consisting of 10 alphanumeric character. Sometimes observation well no. is modified. In some case previous number may be useful. Hence old station number should be stored in the Data Base.

v. Type of well: This can be assigned to any data consisting of 10 alphanumeric character. There may be two type of well: DUG & PIEZOMETER.

vi. Availability of records(since): It is an integer variable indicating year of establishment of the observation well.

vii .Latitude: Latitude of the observation well consists of two numbers, an integer number (degree) and a real number (minute).

viii. Longitude: Same as latitude.

ix. Elevation of MP above MSL

x. Height of MP from ground surface

xi. Depth of well

All of the items (ix-xi) can be assigned any real number.

The calling subprograms of STEDIT used in (i) Insertion (ii) Modification (iii) Deletion and (iv) Listing of river station characteristics are STAINP, STAMOD, STADEL, STALST. All these subprograms belonging to STELIB are modified so that observation well characteristices can be inserted, modified deleted and listed easily. By the modified STEDIT program the basic characteristics of observation well are entered in Data Base exactly in the sequence as they are described above. The application program however, list them in a slightly different sequence. Sample results are given in Appendix-B

3.5.3 Modification in water level file management subprograms

Program STAGES (of WMO DBMS) can be used to handle water level data collected at four intervals: 1 day, 60 minutes. 30 minutes and 15 minutes. Calling subprograms of STAGES which are used in (i) selection of interval (ii)insertion (iii) modification of water level data are INSTP, ISTG, MSTG respectively. All these subprograms belonging to STGLIB are modified so that weekly groundwater level can be entered and modified directly. The old STAGES program uses two codes for missing data. -998 is taken as the initial value and -999 is entered as the missing data. A third code, -997 is to be entered for weekly missing data. This code is introduced to the system from consideration of efficiency of the system.

3.5.4 Structure of the system

When the necessary observation, measurement and recording of the data is made, function of the data processing system begins. Data Base files are first created by the DBMS program. Data (basic characteristics of observation well and groundwater level data) are then stored in Data Base files. Any modification of the data, if necessary is made by the DBMS program. Stored data can be retrieved by the system for use in application program for further processings. Schematic work structure of the data processing system is shown in Fig. 3.5.

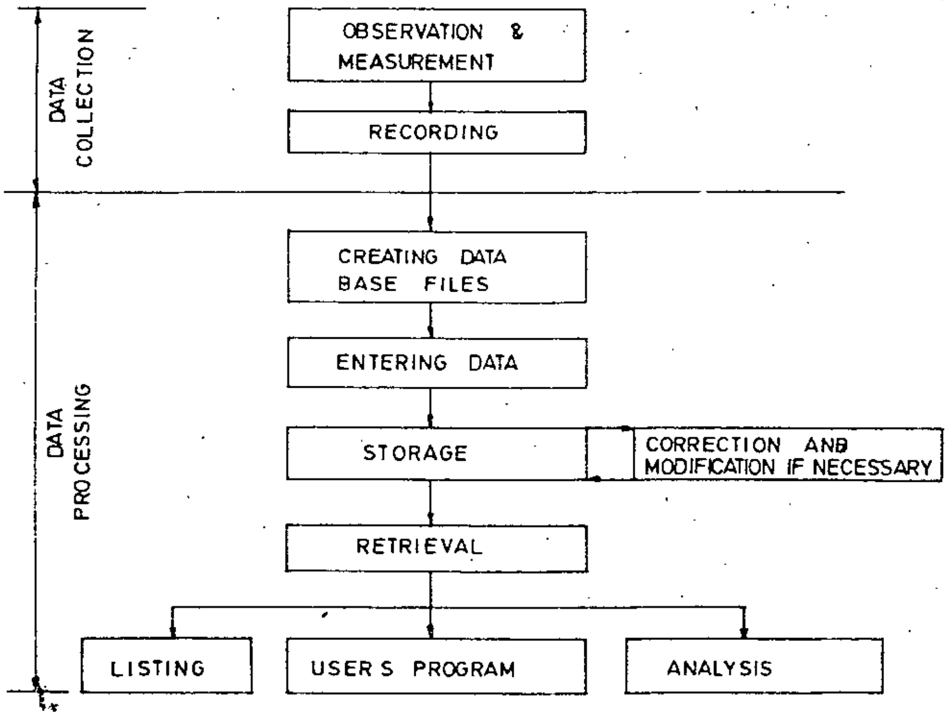


FIG. 3.5 STRUCTURE OF THE DATA PROCESSING SYSTEM

3.6.0 DEVELOPMENT OF THE APPLICATION PROGRAM

3.6.1 General

On the basis of the primary analyses of the data and missing data estimation described in the article 3.30 and 3.40 an application program (Program USER) is developed for the common users of the data processing system. Program USER embodies 14 subprograms, namely DATAIN, LSTSTN, LIST, MISS1, SPLIN, SPLICO, NMISS, MISS2, AREA, SPLSTN, STAT, ADDRESS, PERMIT, HARM and a Block Data subprogram. These subprograms are grouped in a library (USRLIB). Purpose of each of these subprograms, parameters involved, COMMON BLOCKS, called subprograms etc are described in detail in appendix- B . Listing of the source program is presented in appendix- D . A short description of the subprograms is given below.

3.6.2 Subprogram for retrieval of data

Subprogram DATAIN interact with the Data Base through TRN-group of subprogram of the DBMS and read in basic characteristics of observation well and groundwater level data. For a particular observation well and for a particular year. Raw data input to the Data Base are in centimetre (integer) and measured w.r.t. some measuring point. After retrieval from the Data Base the subprogram convert them in metre (real) and with respect to MSL .

3.6.3 Subprogram for missing data estimation

After retrieval of the data from Data Base computer first checks the presence of missing data. If present, control is first transferred to subroutine MISS1 which attempts to estimate missing data first by using spline-fit interpolation. If spline-fit interpolation is aborted to estimate missing data control is transferred. to subroutine MISS2 which estimates missing data by weighted-average method. If missing data estimation be still unsuccessful, primary analyses cannot be performed by the application program. In that case, users can only list data(basic characteristics of observation well and groundwater level data).SPLIN, SPLICO, NMISS are used as the calling subprograms of MISS1 . AREA and DATAIN are used as those of MISS2. Their functions are described in appendix- B

3.6.4 Subprogram for listing station characteristics

The various characteristics of an observation well mentioned in article 3.5.2 is listed by the subprogram LSTSTN. Sample listing is given in appendix- B

3.6.5 Subprogram for listing groundwater level data

Groundwater level data (of a particular station-year record) together with their date of measurement can be listed by the subprogram LIST. This subprogram can list both the row data as it is entered in the Data Base and the refined data (missing data estimated). Sample results are given in appendix-B.

3.6.6 Subprogram for computing statistical parameters

The various statistical parameters described in article 3.4.1 is computed by the subprogram STAT. Sample results are given in appendix- B

3.6.7 Subprograms of determining significant harmonics

The significant harmonics together with their amplitudes and phases is computed by the subprogram HARM. Sample results are given in appendix- B

3.6.8 Flow of processing

Program USER first select a particular station (observation well) and a particular year for which data processing is needed. It then retrieves data (basic characteristics of observation well and groundwater level data) corresponding to that station-year record, estimates missing data if any and displays a menu to the screen. Detailed description of this menu is given in User's guide (appendix- C) . User's can perform their desired processing on the retrieved data using this menu. To perform analysis on data of a new station year record, user's must first exit this menu. Computer will then ask the user to enter the station code of the new observation well and the year of analysis. The flow chart is showing data retrieval and analysis is given in Fig.3.6.

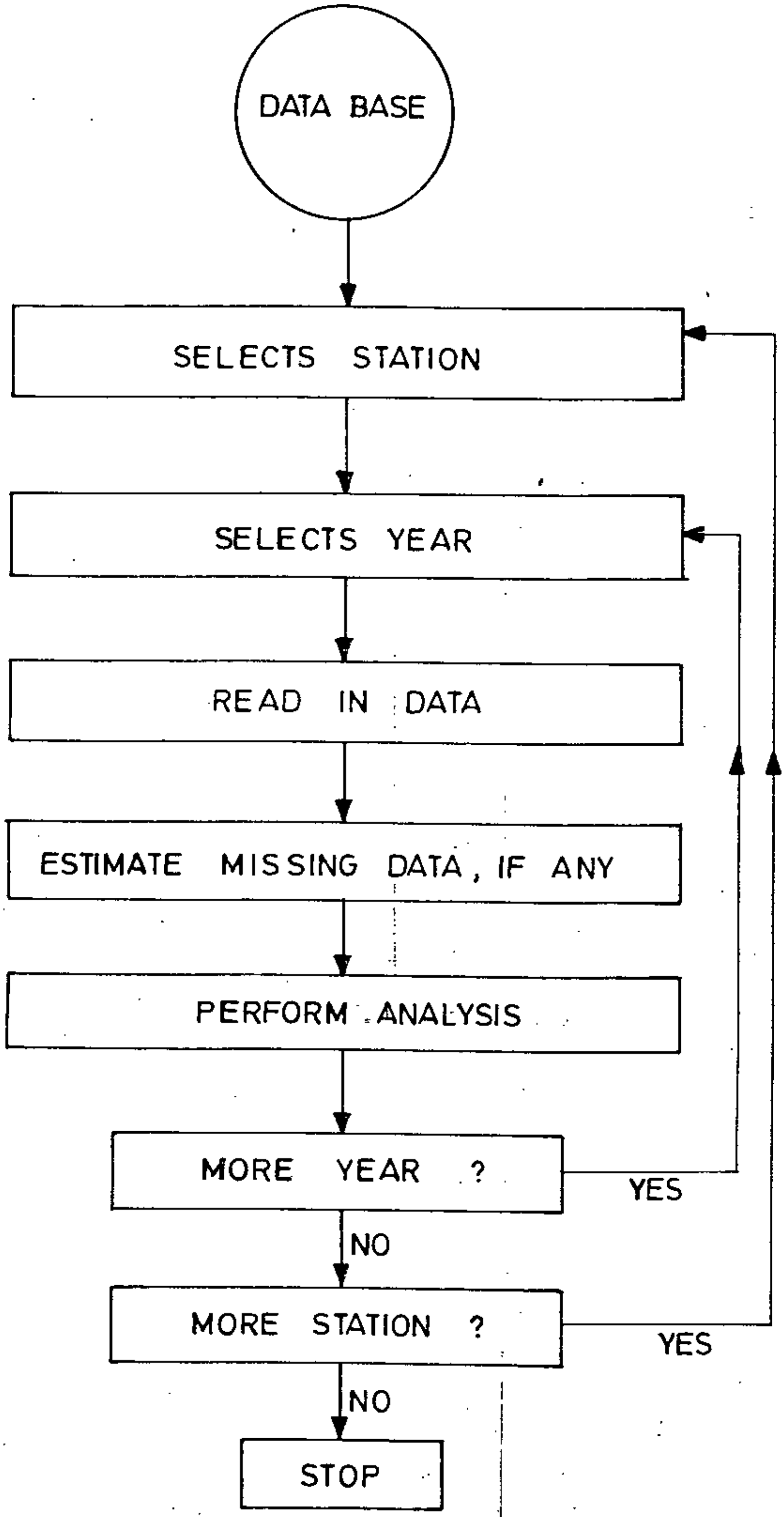


FIG. 3.6 FLOW CHART SHOWING DATA RETRIEVAL AND ANALYSIS

3.7.0 USER'S GUIDE

A guide has been prepared for the user's of the data processing system. To facilatate use of the guide, users are categorized into three classes.

Class-A: Those who will run the application program developed in the present study or run their own application program retrieving data from the Data Base.

Class-B: Those who will enter data in Data Base file, edit it if necessary and then retrieve the data like class A users.

Class-C: Those who will first create the Data Base files and then enter and retrieve data like class B users.

For convenience, the guide is divided into a few sections. Only a user of class C need to consult all the sections;other users will have to read only prescribed sections. The user's guide is given as appendix-C.

Data can be stored in the floppy disk in any systematic way. A procedure is suggested here. One diskette should be used for one year. If all the stations can not be stored in one diskette a second diskette will then be needed. This procedure will minimize the number of diskette. Further it will ease the handling of diskettes, or in other words it will eliminate the operation of locating a diskette that contains a particular station.

3.8.0 SUMMARY

The data processing capability of a data processing system solely depends on application program and without application program a Data Base is quite useless. A data processing system, should, therefore be equipped with large number of multipurpose application programs. The application program developed in the present study will facilitate the users to use data in their application program from Data Base.

Groundwater level data invariably contains missing data and therefore, it is imperative that all application program of diverse user embody some subprograms for missing data estimation.

HARMONIC ANALYSIS OF GROUNDWATER LEVEL DATA

Chapter - IV

HARMONIC ANALYSIS OF GROUNDWATER LEVEL DATA

4.1.0 INTRODUCTION

In this chapter, fluctuation of groundwater level, theory of harmonic analysis and methods of determining significant harmonics have been described. Five years (1979-1983) groundwater level data from ten observation wells in Mymensingh-Jamalpur area have been collected and different methods of determining significant harmonics have been investigated.

4.2.0 FLUCTUATION OF GROUNDWATER LEVEL

Fluctuation of groundwater level is a periodic-stochastic process having annual periodicity. In Bangladesh groundwater level fluctuates mainly due to recharge from rainfall. Besides variations of stream stage and evaporation, tides, atmospheric pressure fluctuation, earthquake, etc. affect groundwater level. Water table rises with the rainfall and starts falling with the cessasion of rainfall at the end of monsoon period. Lowest water level below land surface occurs during the last week of a April and May or in early June. The highest water table occurs during the last week of August and early September. Differences of the highest to lowest water table ranges from 3 to 5 metre in most of the wells although a difference of 1 to 6 metres was observed in some wells (13). The magnitude of fluctuation of water table is the greatest near the river Brahmaputra in Mymensingh-Jamalpur district.

4.3.0 HARMONIC ANALYSIS

4.3.1 Theory

Groundwater level data exhibits annual periodicity and composed of a deterministic component in the form of annual periodic movement in some parameters and a stochastic component which is a random variable. Detailed theory of harmonic analysis is given in (14,15,16,17,18). A short description of the theory is given below:

Let X_t represents the time series consisting of weekly groundwater level data, then

$$X_t = D_t + \sigma_x \epsilon_t \qquad (4.1)$$

where D_t is the periodic component, σ_x is the standard deviation of the sample and ϵ_t is the stochastic component

The periodic component D_t is mathematically described as

$$D_t = A_o + \sum_{i=1}^{m} A_i \operatorname{Cos}(2\pi it) + B_i \operatorname{Sin}(2\pi it) \qquad (4.2)$$

where t = 1,2,3....N with N the basic period of the series

m = Nos. of harmonics and can have any between 1 and N/2 (if N is even) or (N-1)/2 is N is odd.

The f The fourier Co-efficients A_o, A_i, B_i are defined as

$$A_o = \frac{1}{N} \sum_{i=1}^{N} X_t = \bar{X}$$

$$A_i = \frac{2}{N} \sum_{t=1}^{N} (X_t - X) \text{ Cos } (\frac{2\pi it}{N})$$

$$B_i = \frac{2}{N} \sum_{t=1}^{N} (X_t - X) \text{ Sin } (\frac{2\pi it}{N})$$

Series (2) can be written in its equivalent form

$$D_t = \overline{X} + \sum_{i=1}^{m} C_i \text{ Cos } (\frac{2\pi it}{N} + \varphi_i) \qquad (4.4)$$

where C_i = Amplitude of ith harmonics

$$= \sqrt{A_i^2 + B_i^2} \qquad (4.5)$$

φ_i = Phase of ith harmonics

$$= \tan^{-1} (-Bi/Ai) \qquad (4.6)$$

The stochastic compnent ϵ_t may be computed using

$$\epsilon_t = \frac{1}{\sigma_x} (X_t - D_t) \qquad (4.7)$$

4.3.2 Methods of determining Significant Harnomics(m)

Four alternative approaches were suggested by Vujica Yevjevich for finding out number of significant harmonics :

1. First approach selects on as a constant ($6 \leqslant m \leqslant 12$) for all practical purpose with D_t having m first harmonics as significant regardless of the probability of significantee level, N, the length of periodicity.

2. The second approach is to use a lower probability significance level. P_m, used as a constant in this study (P_{min} = 0.10). If the m first harmonics have the explained variance defined as

$$P = \frac{\sum_{j=1}^{m} C_j^2}{2* \text{ VAR } (X_t)} \qquad (4.8)$$

which is less then. P_{min}, no significant harmonics is present in X_t. If this explained variance (P) is greater than P_{min}, m harmonics are used.

3. The third alternative approach is the use beside of P_{min}, also the upper limit P_{max}, computed as $P_{max} = 1- P_{min}$. If $P < P_{min}$, no significant harmonics is found in the squence of D_t values. If $P_{min} < P \leqslant P_{min}$ all m harmonics determined by 2nd approach is significant. If however $P > P_{max}$, only some of the m harmonics are significant. The value of $C_j^2/2$ are then shorted from highest to lowest. Only those harmonics with highest $C_j^2/2$ when summed exceeds P_{max} for the first time are selected.

4. The fourth method is a graphical estimation procedure which uses the breaking point of a comulative periodogram. Equation (4.8) reveals that P is a function of m and as a matter of fact if P(m) is plotted against m, graphs like (Fig.4.1) are obtained. The graph composed of two district parts (i) The peridoc part of first rising of P(m) with m and (ii) The sampling part of the slow rising of P(m) with m.

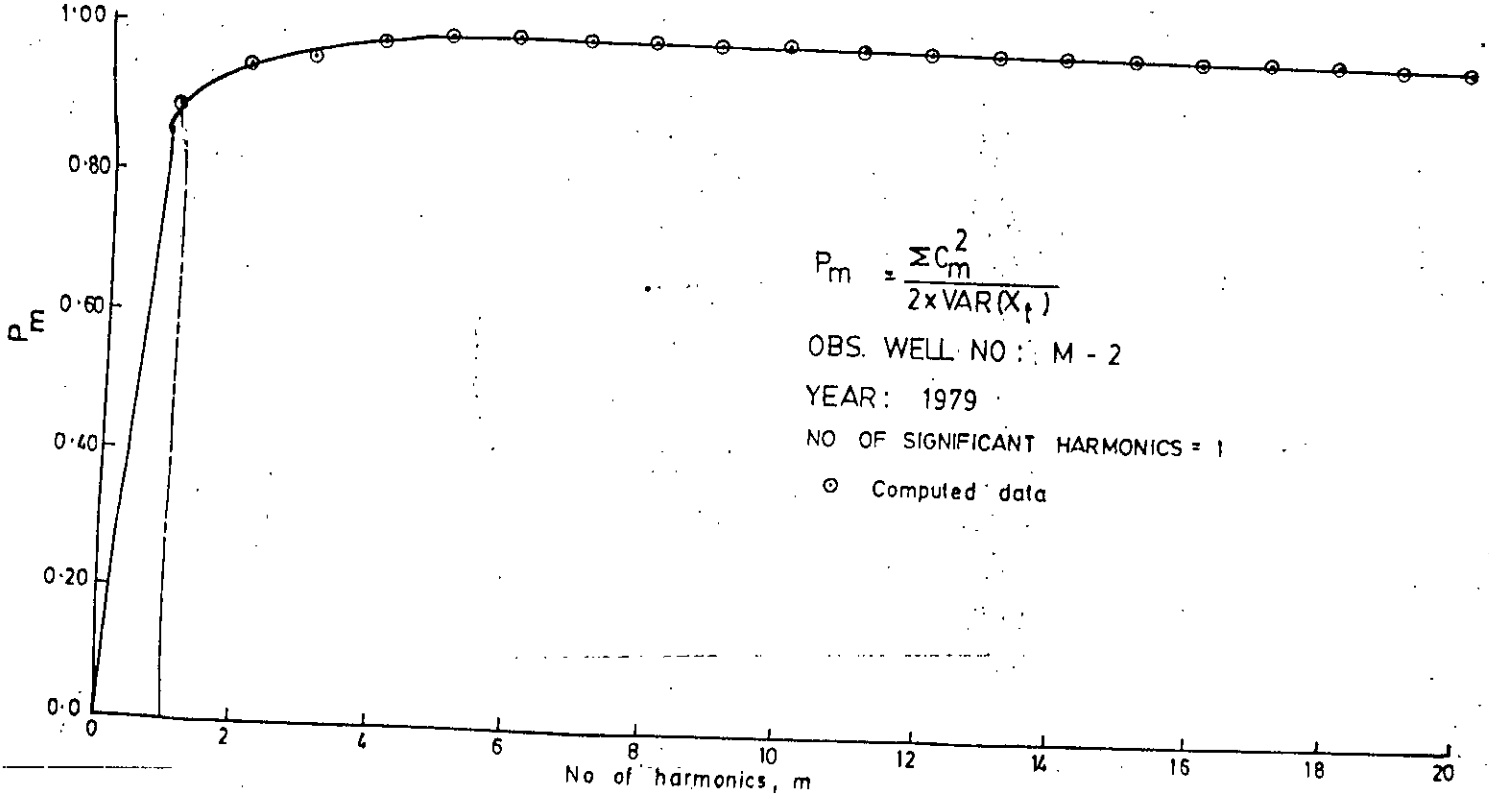


Fig. 4·1 Cummulative Explained Variance (P_m) by the First m Harmonics

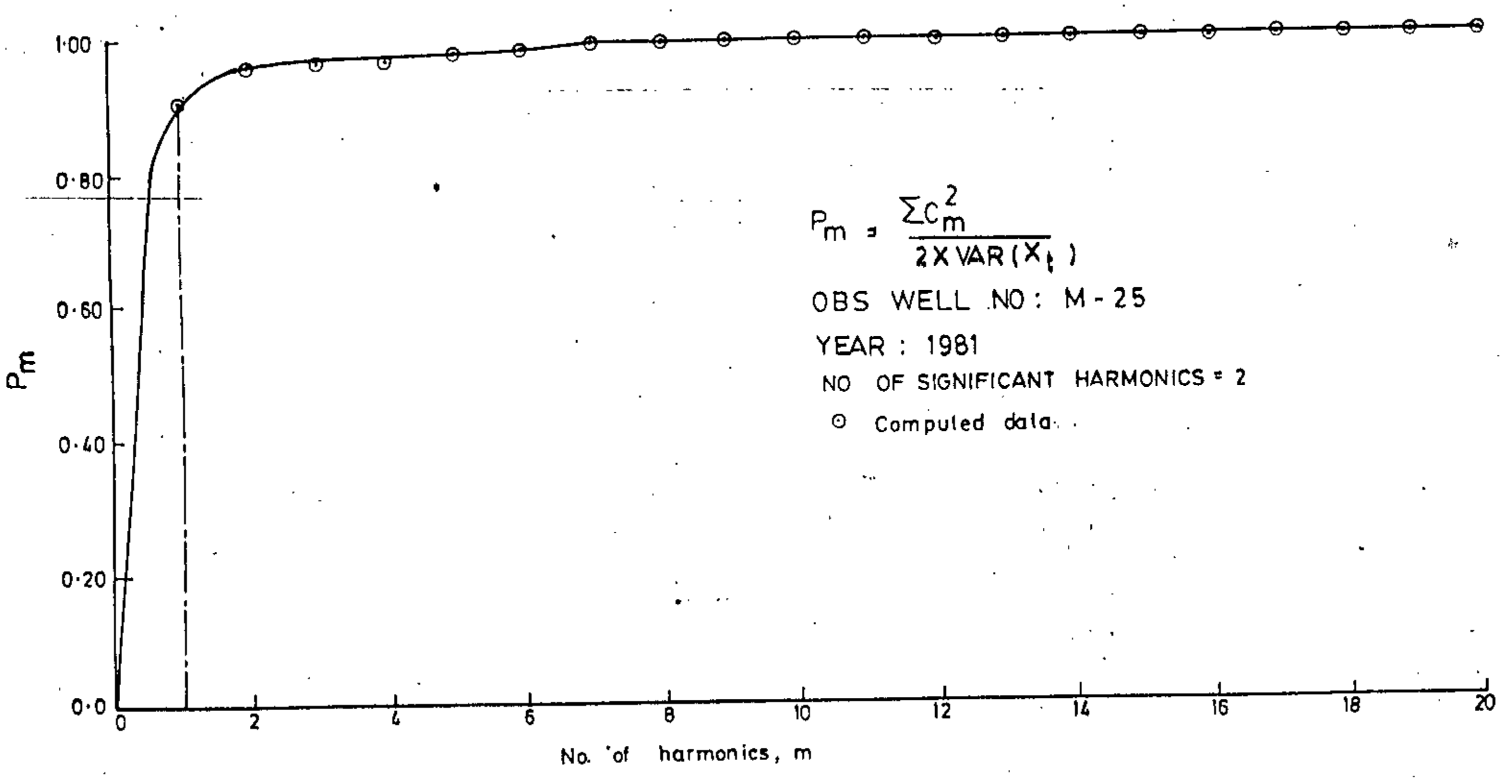

$P_m = \frac{\Sigma C_m^2}{2 \times VAR(X_t)}$
OBS WELL NO: M-25
YEAR : 1981
NO OF SIGNIFICANT HARMONICS = 2
⊙ Computed data
P_m
1·00
0·80
0·60
0·40
0·20
0·0
0
2
4
6
8
10
12
14
16
18
20
No. of harmonics, m

FIG. 4·1 (Contd.)

To estimate the breaking point two approaches are feasible. First, the two parts are approximated by smooth curves that intersects at a point. The intersecting point gives the number of significant harmonics.

In the 2nd approach the two parts are estimated by straight lines, which intersects at a point giving the number of significant harmonics. The procedures are illustrated in Fig.4.1

4.4.0 ANALYSIS OF GROUNDWATER LEVEL DATA FROM MYMENSINGH-JAMALPUR AREA

4.4.1 Description of the study area

In order to demonstrate the applicability of the present data processing system groundwater level data from Mymensingh-Jamalpur area have been collected. A brief hydrogeological description of the area is given below.

The area covers eastern region of Jamalpur and north-western region of Mymensingh. Over the area the old Brahmaputra flows through the eastern region as shown in Fig. 4.2 . The area consists of pleistocene area, low-lying area (depth of flooding 2 m) high and intermediate lands (depth of flooding 0.3m) and very low lying area (depth of flooding 2m). The ground surface elevation in the Mymensingh is found to be 65 feet as a maximum value of elevation and 15 feet as a minimum value of elevation and those of Jamalpur are 70 feet and 20 feet respectively.

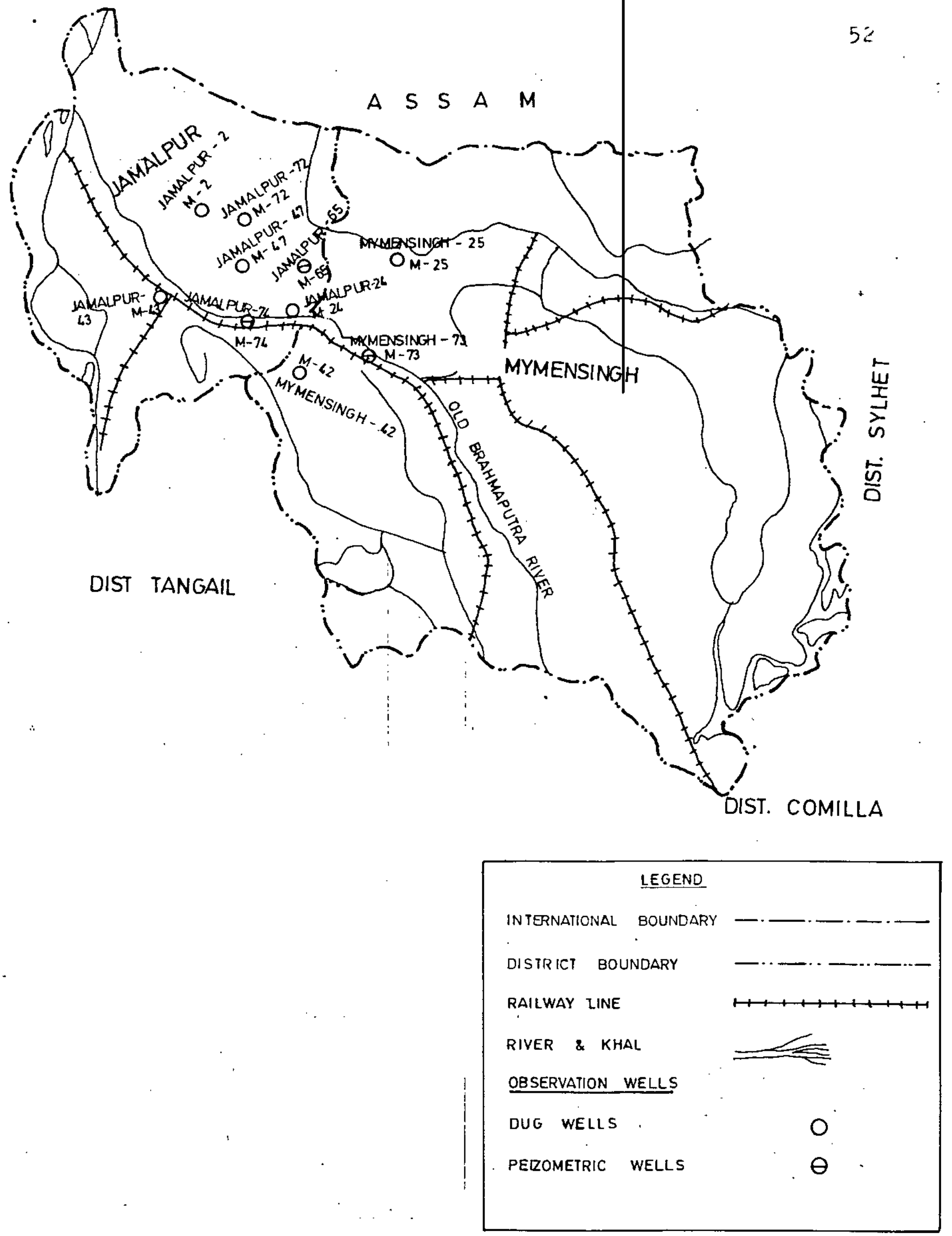


Fig. 4·1 Study area Showing Observation Wells

Rainfall is one of the main recharge components. The part of the accretion to the groundwater occurs mainly due to horizontal inflow from the river old Brahmaputra. In the study area with the cessation of rainfall in the month of November of in the latter part of October, the major scope for recharging of groundwater is stopped, groundwater level begin to fall and the recharged water is being gradually lost by discharge into the river.

The lithological data of the soil shows that water bearing formations contain mainly of medium sands. In the Jamalpur region the formation also contained coarse sand with thickness varying from 100 ft to 180 ft, while in Mymensingh region some fine to medium sand (thickness exceeding 80 feet) also exists in the water level formation.

4.4.2 Collected data

Sample (a station-year record) data was collected from ten observation wells (M-2, M-24, M-25, M-42, M-43, M-47, M-65, M-72, M-73 and M-74) in Mymensingh-Jamalpur area.(Fig. 4.2). Usually a station-year record of weekly groundwater level data contains 52 data. But as groundwater level is measured each monday, so this may cause collection of 53 data in some years. Groundwater level data exhibits annual periodicity. 53 data represents a period (53 week) which is greater than a year. Hence, harmonic analysis is performed on first 52 data. Sample having too many missing data whose estimation is out of scope of the present data processing system is excluded from harmonic analysis. Table 4.1 presents data sample on which harmonic analysis is performed. Sample on which analysis is not performed is marked by X.

Table 4.1 Sample data on which harmonic analysis is performed

STATIONS YEAR	M-2	M-24	M-25	M-42	M-43	M-47	M-65	M-72	M-73	M-74
1979	✓	✓	✓	✓	✓	✓	X	✓	✓	X
1980	✓	✓	✓	✓	✓	✓	X	✓	✓	✓
1981	✓	✓	✓	✓	✓	✓	X	✓	✓	✓
1982	X	X	X	X	X	X	X	X	X	X
1983	X	X	X	X	X	X	X	X	X	X

Out of 10 wells, M-65, M-73, M-74 were peizometer wells and all others were dug wells. From the statistics of samples it has been found that water level in those well in the duration (1979-1983) fluctuated from 2.160 metres to 4.360 metre. Minimum water level fluctuation occured in observations well M-043 in 1981 and maximum fluctuation occured in observations well M-73 in 1979. Maximum water level above mean sea level (MSL) was in observations well M-25 and in 1981 . Minimum level above MSL was .81 metre and in observations well M-74 in 1980.

4.4.3 Comparison of methods of determining significant harmonics

For all samples data on which analysis was performed the number of significant harmonics have been determined by all methods. Typical results for two observation well M-2 and M-25 is given in table 4.2. The table shows that only first harmonics is significant in most case as determined by method 2,3 and 4.

Table 4.2 Typical comparison of number of significant harmonics determined by four methods.

Observation well	Year	Number of significant harmonics			
		Method 1	Method 2	Method 3	Method 4
M-2	1979	6-12	1	1	1
	1980	6-12	1	1	1
	1981	6-12	1	1	1
M-25	1979	6-12	1	1	2
	1980	6-12	1	1	1
	1981	6-12	1	1	2

The periodic component D_t (computed groundwater level) have been computed for all possible number of significant harmonics. Sum of the squares of the residuals have been computed using

$$S_i = \sum_{t=1}^{52} (X_t - D_t)^2$$

These values for the two observations well M-2 and M-25 is given in table 4.3.

Table 4.3 Sum of squares of residuals (S_i, in square metre)

Obs. well	M-2			M-25		
i Year Nos.of harmonics	1979	1980	1981	1979	1980	1981
1	16.6	4.82	6.53	5.42	5.47	3.82
2	44.2	10.70	11.56	2.28	11.50	1.80
3	41.8	8.57	19.98	1.96	9.05	3.47
4	49.9	9.11	18.90	3.17	8.83	3.53
5	40.1	8.93	19.01	4.22	10.48	3.53
6	50.8	8.90	19.10	4.13	10.25	4.53
7	50.2	9.23	19,16	5.11	10.44	4.99
8	50.6	9.32	18.73	4.99	10.58	6.13
9	50.6	9.32	18.74	4.95	10.48	6.02
10	50.7	9.30	18.97	5.27	10.44	5.96
11	50.7	9.27	18.87	5.21	10.91	6.10
12	50.7	9.27	18.83	5.35	10.90	6.08

On the basis of minimum and maximum S_i, observed (X_t) and computed (D_t) groundwater level have been plotted against time, t (weeks). The best and worst fit for these two observations well is shown in Fig. 4.3 and 4.4 respectively. The upper curves (Fig. 4.3a, 4.3c, 4.4a, 4.4c)shows the best fit obtained by different methods. The lower curves (Fig. 4.3b, 4.3d, 4.4b, 4.4d) shows the worst fit obtained. The amplitudes and phases of various harmonics in those station-year is shown in Fig.4.5. Obviously it is difficult to conclude for the number of significant harmonics using figures like 4.3 and 4.4. Table 4.3 shows that values of squares of residuals (S_i) obtained by 6-12 harmonics (method 1) is large compared to those obtained by using 1 or 2 harmonics. Thus method 1 is not applicable for weekly groundwater level data of Mymensingh-Jamalpur area. Table 4.3 further shows that method 2 and method 3 always suggest the first harmonics as significant. However, in some cases (observations well M-25, year 1979; observations well M-35, uear 1981) sum of the squares of residuals obtain using 2 harmanics is less than those obtained by using the first harnomics. Method 4 determines 2 harnomics in thos cases. Method 4 is, therefore, selected and included in the present data processing system. However, from the nature of the curves shown in Fig. 4.3 and 4.4 as well as from the values of some of squares of residuals (Table 4.3) it can be inferred that the first harmonics can be taken as significant for all practical purposes.

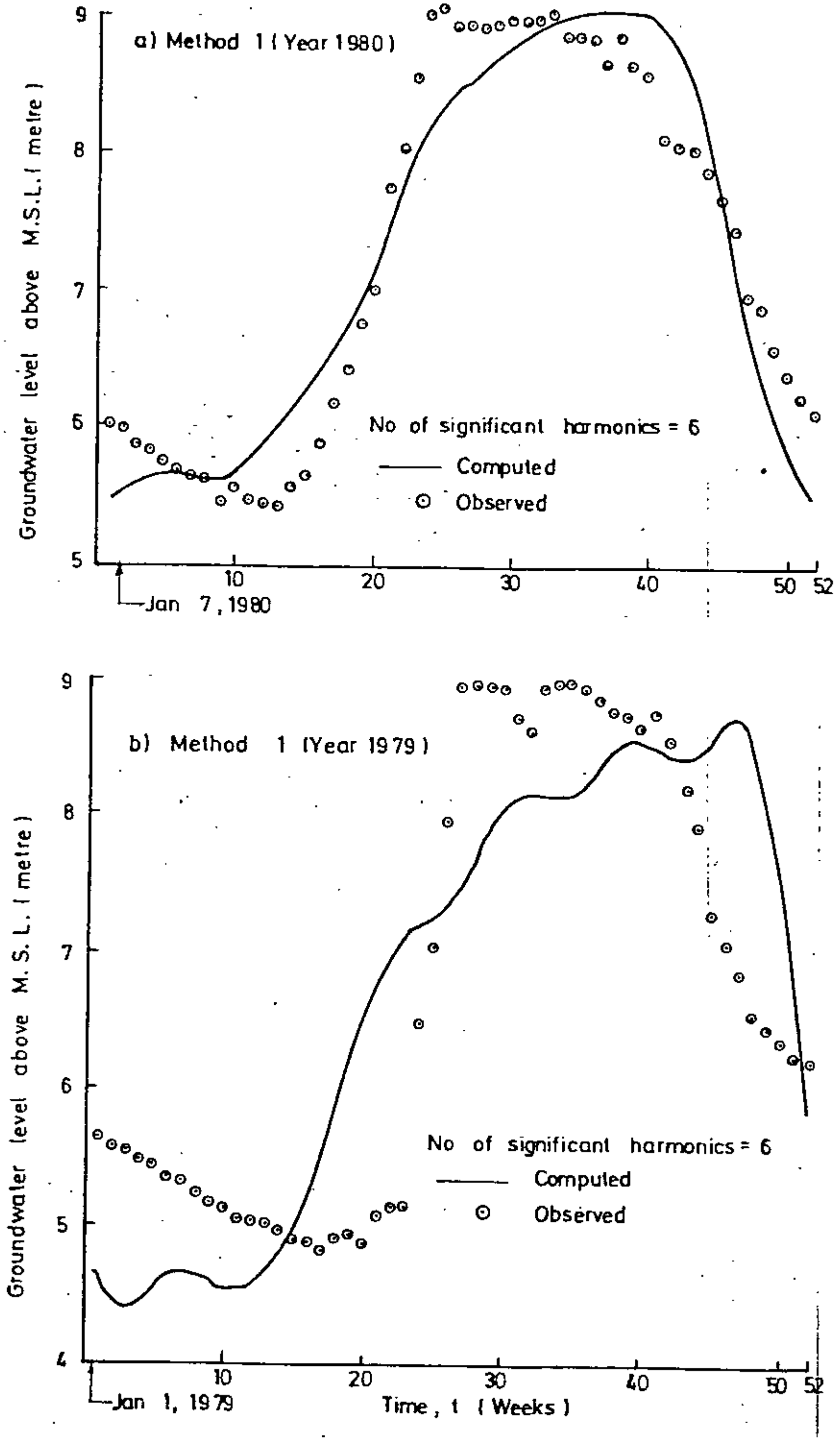


Fig. 4·3 Typical Comparison of Computed and Observed Variation of Groundwater level at Observation Well M - 2

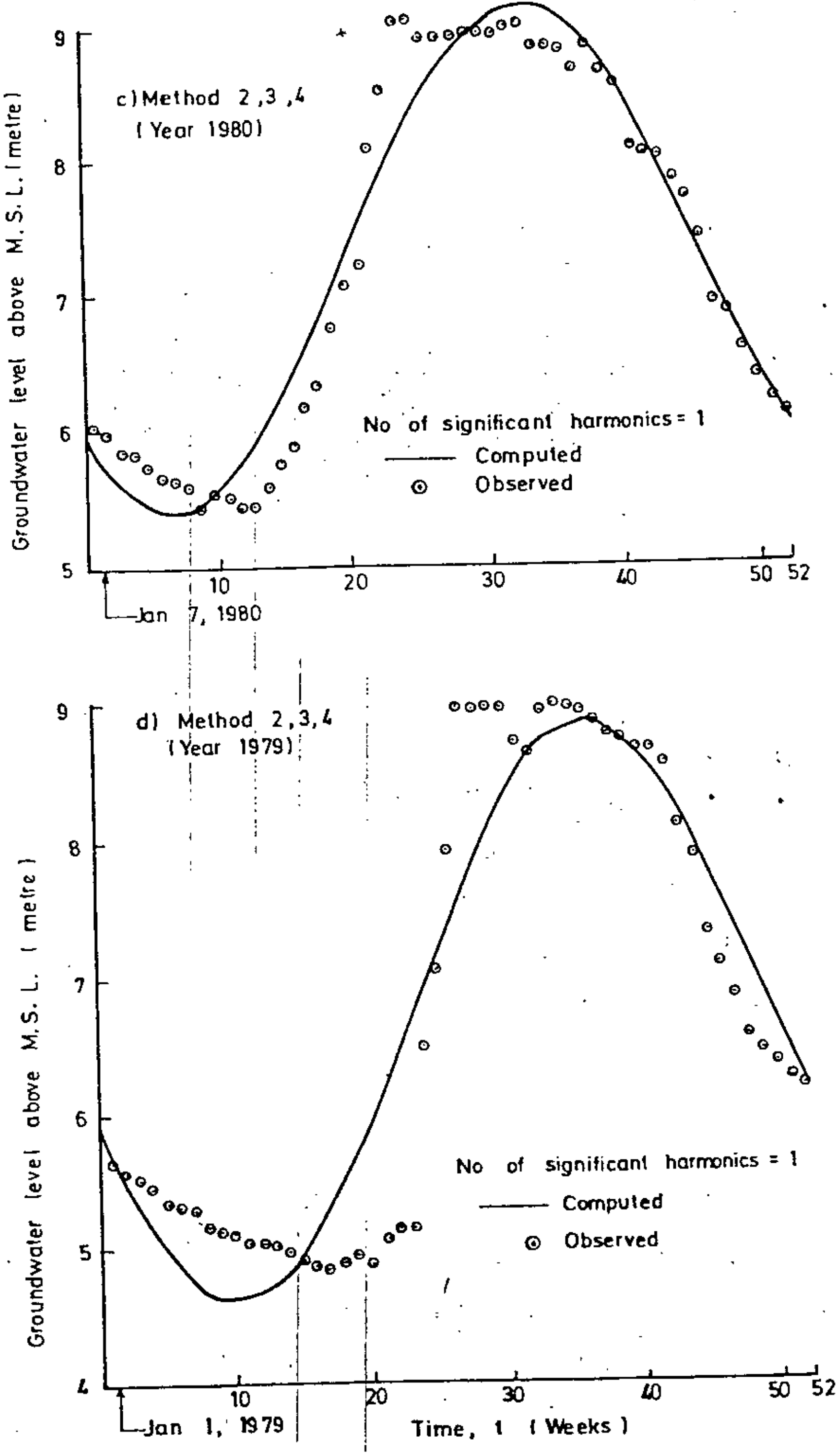

c) Method 2,3,4
(Year 1980)
No of significant harmonics = 1
Computed
⊙ Observed
Groundwater level above M.S.L. (metre)
5
6
7
8
9
10
20
30
40
50
52
Jan 7, 1980
d) Method 2,3,4
(Year 1979)
No of significant harmonics = 1
Computed
⊙ Observed
Groundwater level above M.S.L. (metre)
4
5
6
7
8
9
10
20
30
40
50
52
Jan 1, 1979
Time, t (Weeks)

Fig. 4.3 (Contd.)

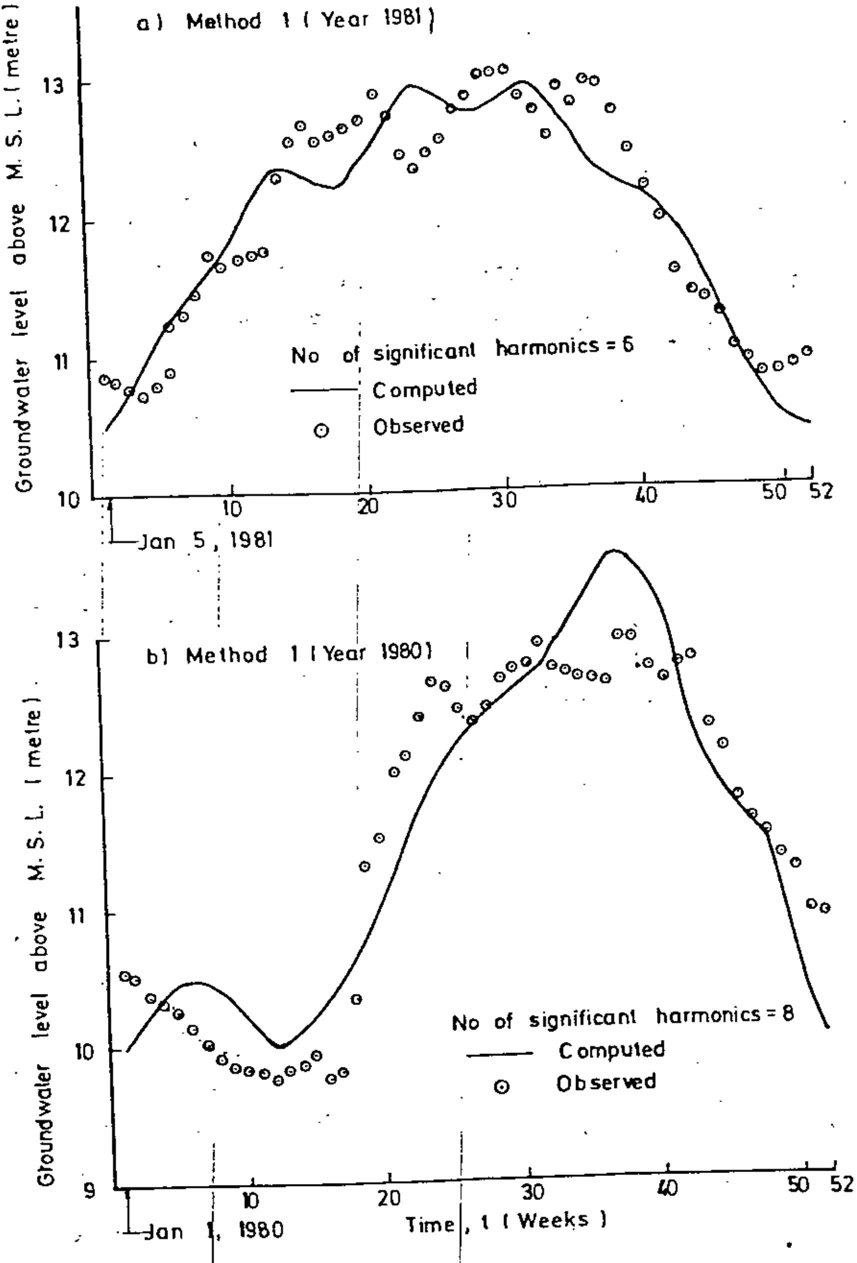
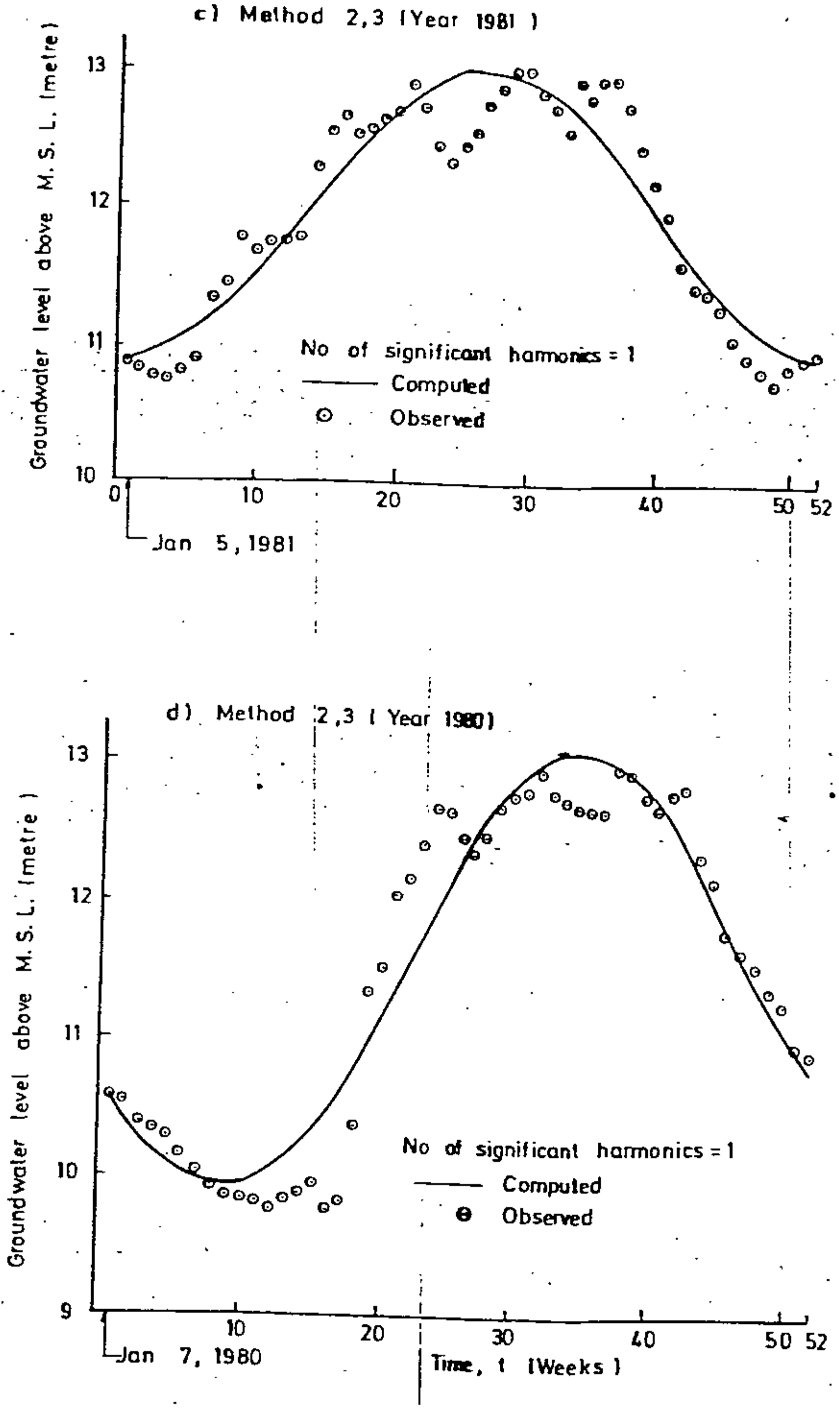


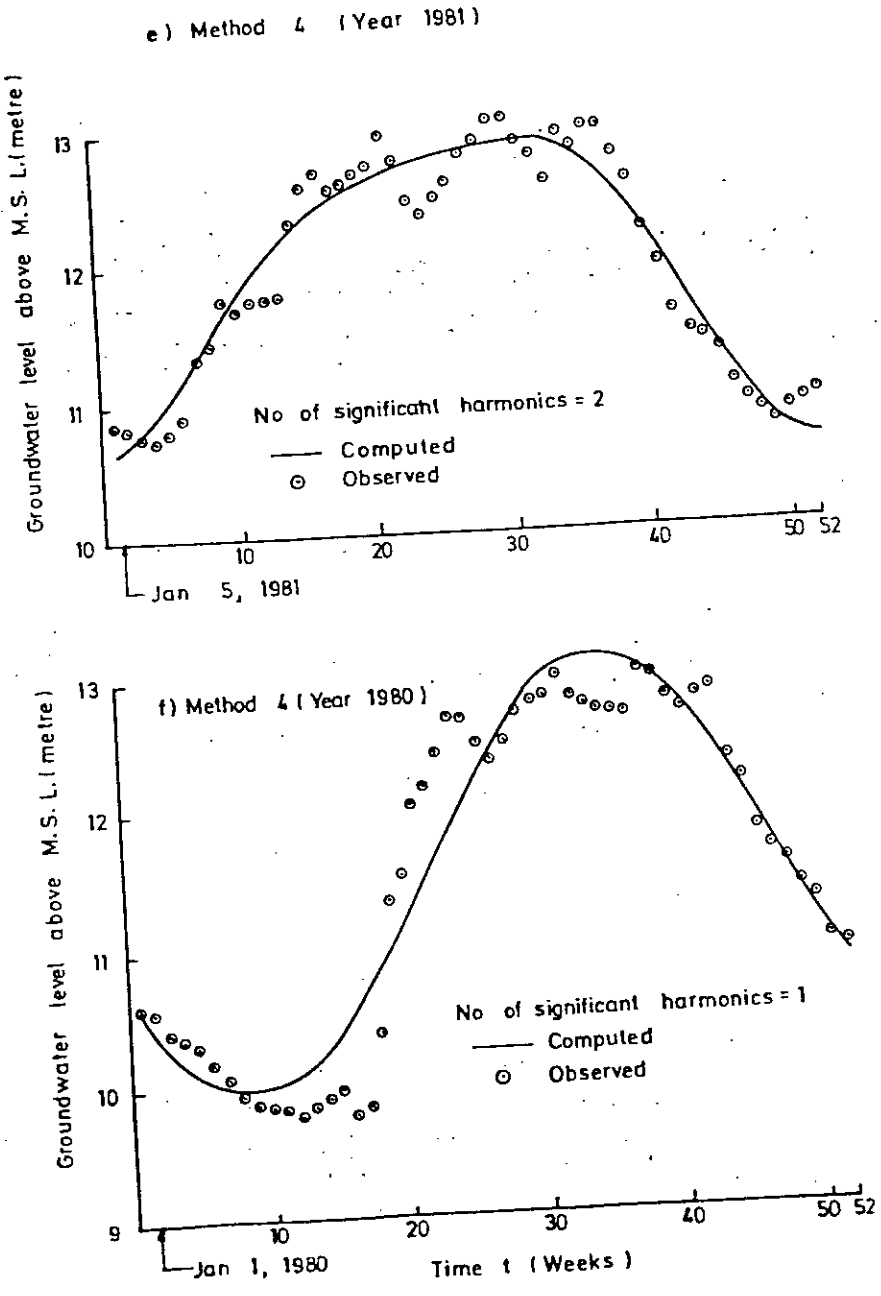
Fig. 4.4 Typical Comparison of Computed and Observed Variation of Groundwater level at Observation Well M-25



c) Method 2,3 (Year 1981)

d) Method 2,3 (Year 1980)

Fig. 4.4 (Contd.)



e) Method 4 (Year 1981)

f) Method 4 (Year 1980)

Fig. 4·4 (Contd.)

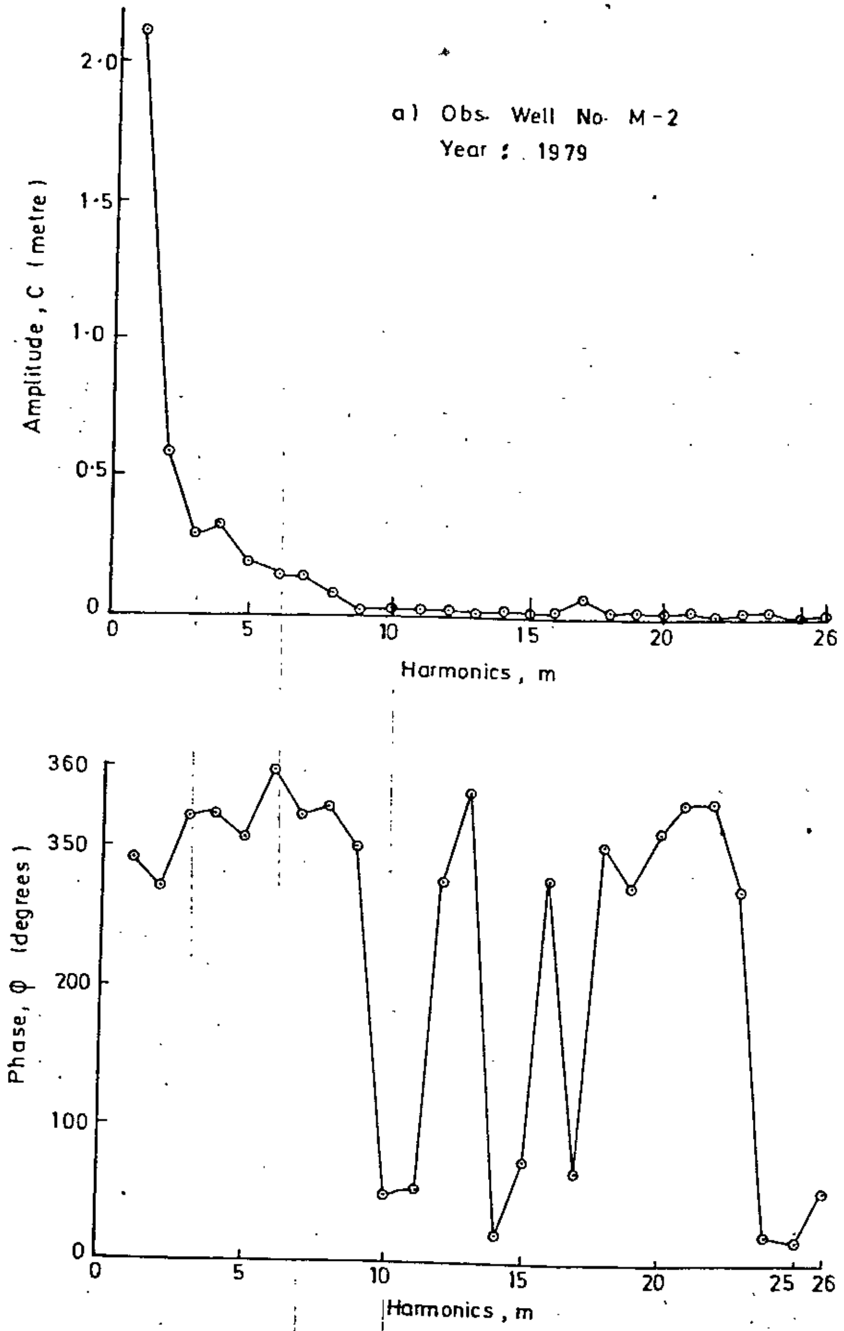


Fig. 4.5 Computed amplitudes and phases of various harmonics

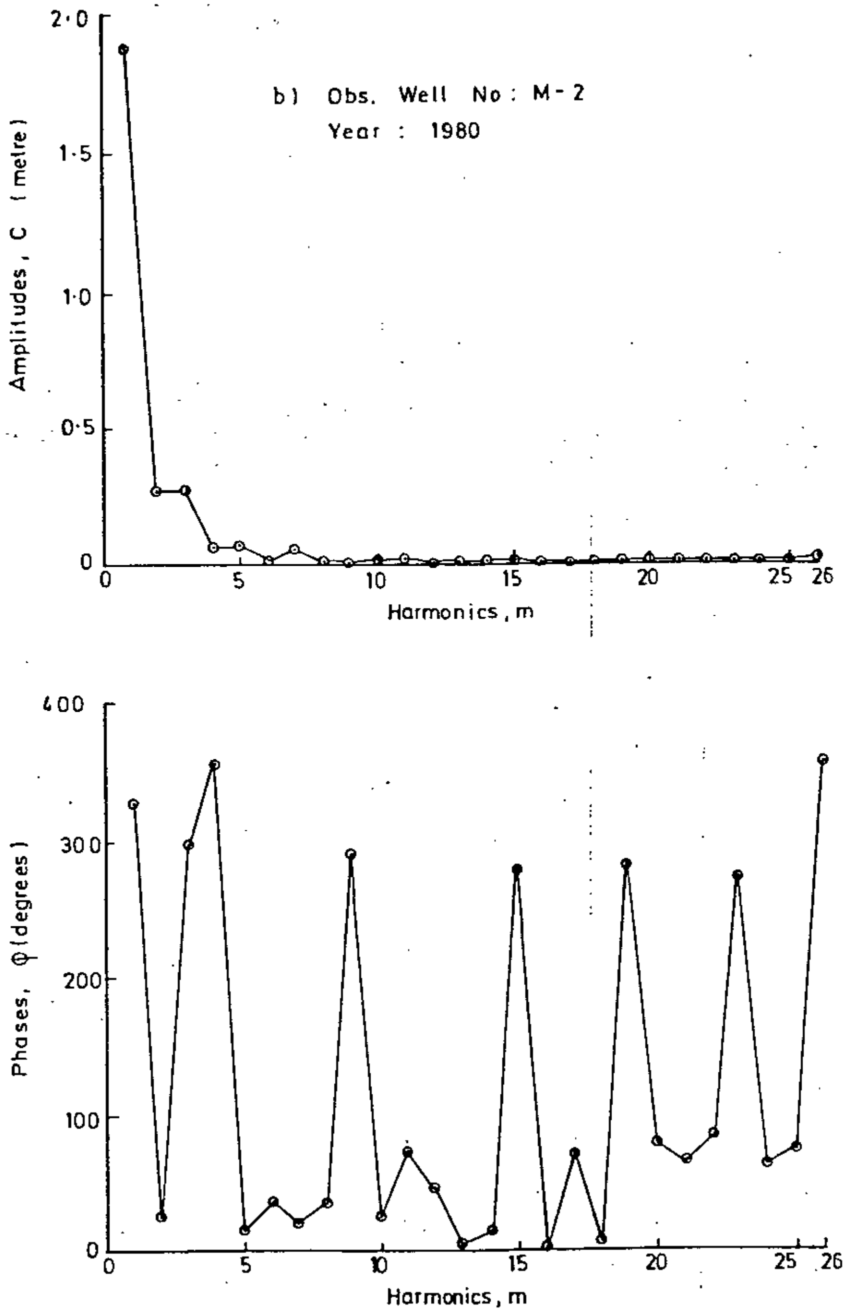
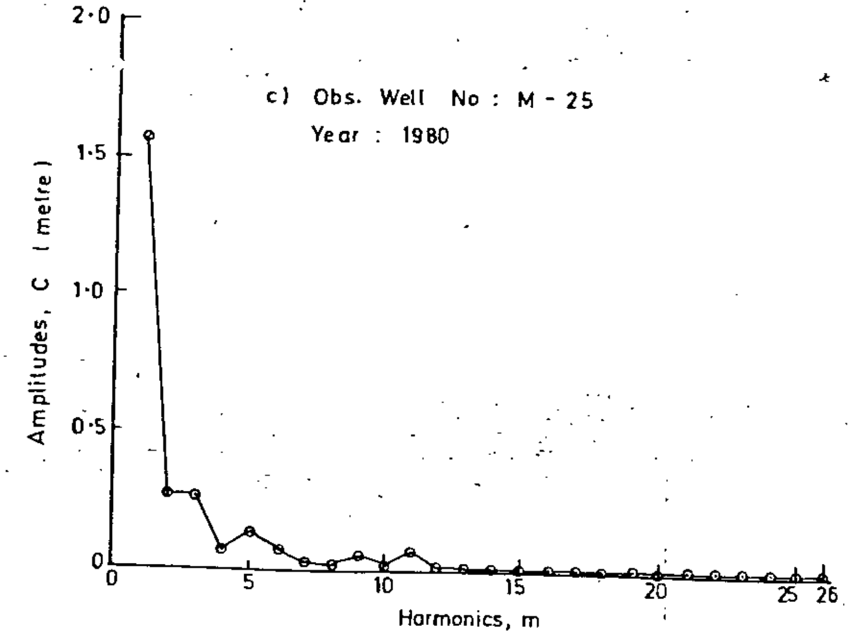

2·0
1·5
1·0
0·5
0
b) Obs. Well No : M-2
Year : 1980
Amplitudes, C (metre)
0
5
10
15
20
25
26
Harmonics, m
400
300
200
100
0
Phases, φ (degrees)
0
5
10
15
20
25
26
Harmonics, m

Fig. 4·5 (Contd.)


c) Obs. Well No : M - 25
Year : 1980
Amplitudes, C (metre)
2·0
1·5
1·0
0·5
0
0
5
10
15
20
25
26
Harmonics, m

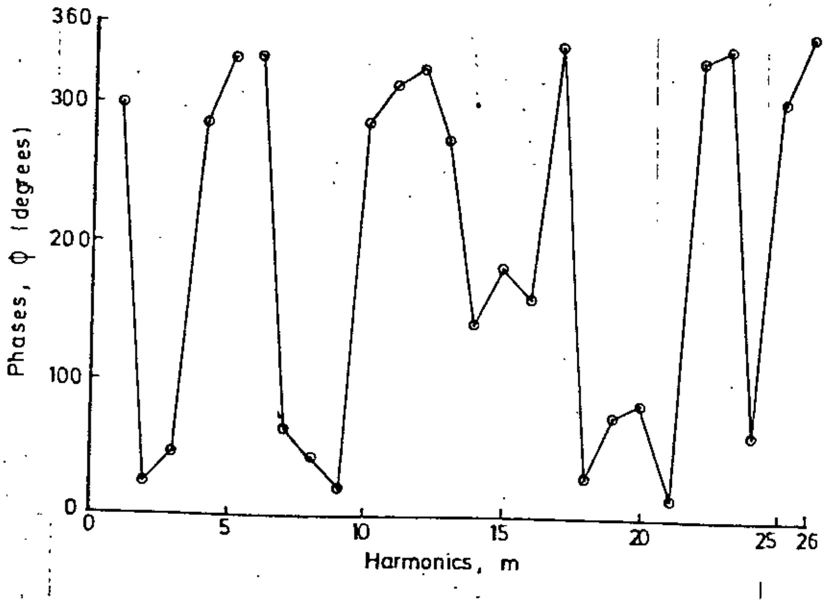

Phases, φ (degrees)
360
300
200
100
0
0
5
10
15
20
25
26
Harmonics, m

Fig. 4·5 (Contd)

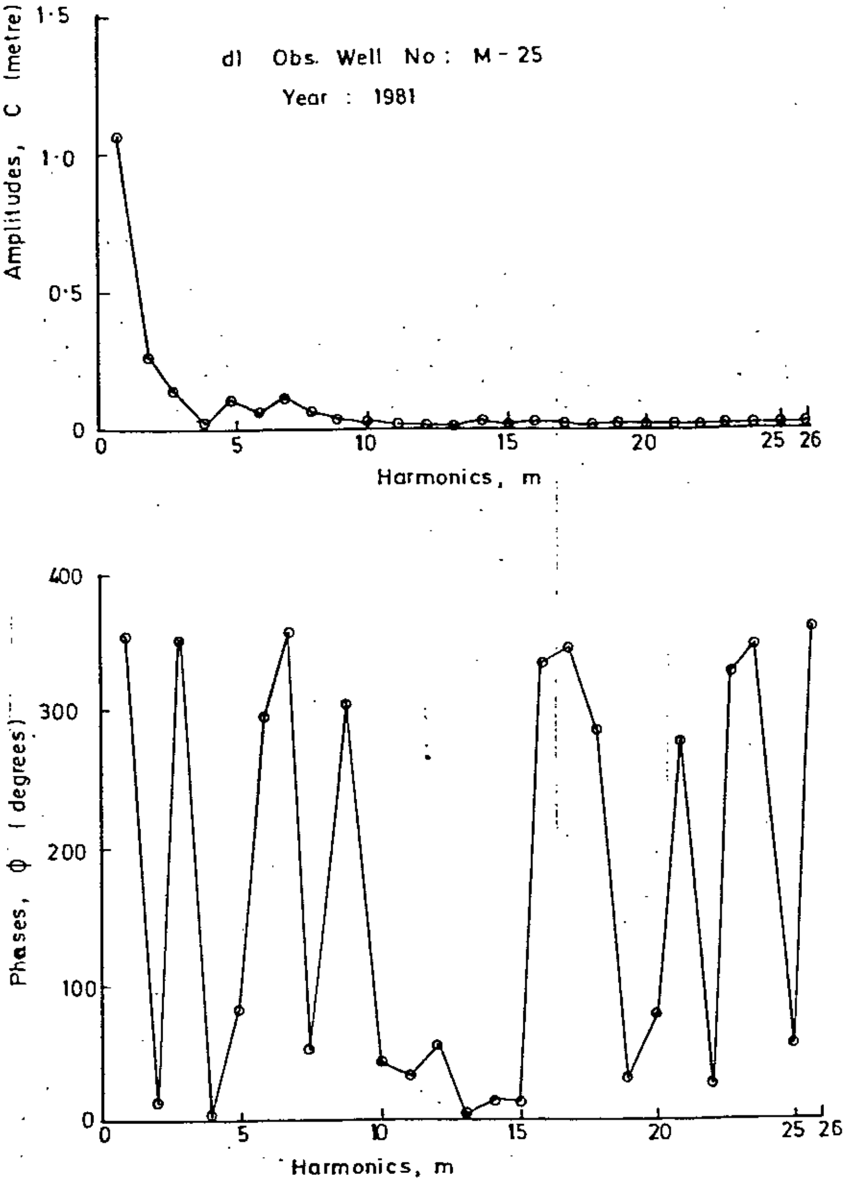

d) Obs. Well No : M - 25
Year : 1981
Amplitudes, C (metre)
1·5
1·0
0·5
0
0
5
10
15
20
25
26
Harmonics, m
Phases, ϕ (degrees)
400
300
200
100
0
0
5
10
15
20
25
26
Harmonics, m

Fig. 4·5 (Contd.)

4.5.0 CONCLUSION

Method 4 (graphical estimation method) has been selected and included in the data processing system for determination of number of significant harmonics. For all practical purpose involving analysis of weekly groundwater level data, the first harmonics may be taken as significant.

DISCUSSION

Chapter - V

DISCUSSIONS

The data processing system developed in this study is essentially a Data Base Management System (DBMS) coupled with an application program. This DBMS is a modification of that of the World Meteorological Organization (WMO). The original DBMS had been developed for surface water data. Modifications was necessary to make it compatible with handling of groundwater level data collected by Bangladesh Water Development Board. This DBMS can be installed in a TRS-80 microcomputer. A TRS-80 model II or any enhanced model such as TRS-80 model 12 TRS-80 model 16, TRS-80 model 16B may be used for the purpose. The system is compatible with several versions of TRSDOS (Tandy Rario Shack Disk Operating System) and TRSDOS-II operating system such as TRSDOS version 2.0a TRSDOS version 2.0b, TRSDOS version 3.0, TRSDOS-II version 4.10 etc. A memory of only 64^k is required for the system. Almost all TRS-80 microcomputers available in our country including those of BWDB and BETS have a memory capacity greater than equal to 64^k.

The source language of system program and subprograms is FORTRAN. Although programming is more complicated in FORTRAN, results can be obtained in shorter execution time than would be required for a program written in BASIC. Further FORTRAN is extrenly efficient for coding complex mathematical and scientific algorithms which are increasingly used during data analysis.

Basic structural unit of the Data Base is a station year record. A station-year record contains the entire set of observation for a particular year. This unit has proved to be efficient for hydrological data (4)

An important development in the present study is the inclusion of facilities for some primary analysis of the data including missing data estimation. Basic analysis comprises determination of highest level, lowest level and range, computation of mean and standard deviation and determination of significant harmonics together with their amplitudes and phases. These parameters are computed for one year at station. Users have given the option so that they can retrieve data with or without the analysis. Inclusion of facility of missing data estimation and primary analysis will greatly increase the practical utility of the system. It will save a great deal of time and programing effort of the user.

A difficulty was encountined during developing the application program due to storage limitation of the computer. Attempt to perform all the analysis using the same retrived data was not successful. This difficulty has been removed by making a compromise.The analysis involving determination of significant harmonics is to be done by retrieving the data again. However, the process of estimating the missing data is not to be repated.

Another important feature of the application program is that it works in interaction with the user. At every step of operation computer displays a message on the screen so that proper response can be easily entered by the user. Once the program is started it will not stop untill and unless requested by the user. Users can perform processings on as may station-year data as recorded provided the data were previously stored in the Data Base. While performing processings on a particular station-year data the program displays a menu on the screen showing several options for data processing e.g., listing of station characteristics, listing of groundwater level data, computation of statistical parameters etc. Just by pressing a single key users can choose any option or even they can terminate processing for the current station-year data without performing any of the processings listed in the menu. Further, a particular option can be selected as many times as desired.

The DBMS programs also works in interactive mode. Insertion modification, deletion of data in data base is performed in interactive mode. This eliminates the possibility of making mistakes during data entry. It also make the operation easier for the user by reducing the effort for learning the system.

The DBMS is most useful when it runs under multiuser operating system. The present system is in single user environment. However when multiuser operating system will be installed in the computer, the data processing system can also be implemented in that system. Present work may be extended in this direction in further studies.

The present system has been designed for data entry in centimetre and it accepts only integer number. The unit of data measurement which BWDB presently uses is metre upto two digit after decimal point. However, the system is flexible enough and can easily be adjusted for data entry in other units say millimetre. After retrieving the data from Data Base the application program again converts it to metre dividing by a factor. Which is obviously 100. If the input unit be changed (say millimetre) then only this factor needs be changed (1000 in this case) in the application program.

Two methods of estimating missing data have been used in the system. They are spline-fit interpolation and weighted-average interpolation. The former method is an interpolation process along time axis only while the latter method involves interpolation along a space (surface) only. The spline-fit method has been found to be better as show in table 3.1 . However if an abrupt fluctuation occurs during missing period due to stochastic reason, this method cannot take into account of this factor. In that case weighted-average method is more reliable. However, such a sudden rise or drop in groundwater the methods except the first one give almost identical result. Second and third method always give the same result and shows the first harmonics as significant in Mymensingh-Jamalpur area. Number of significant harmonics obtained by the fourth method is less than three. When the sum of squares of residuals are compared, minimum values are found in the case of the last method as shown in table 4.2. Hence this method is selected and included in the data processing system.

level is not frequent. Hence for a small number of missing data spline function intepolation can be applied without causing an appreciable error. The weighted-average method is unable to estimate the missing data of a station when data of its neibouring stations is not stored. For these reasons this data processing system is designed in such a way that to estimate missing data, spline function is attempted first. If this method is not applicable due to absence of consecutively three previons and three subsequent data, then the weighted average method is used.

Experience (14) studying periodicities of daily flow and daily precipitation and other hydrologic series reveals that individual harmonics beynod the sixth harmonics add relatively little additional explanation to the variance of the observed data It is shown from the Fig. 4.5 that amplitudes of individual harmonics beyond the third harmonics is less than 5% of that of the first harmonics. Obviously those harmonics are very insignificant in compared to first few harmonics.

Four methods of determining number of significant harmonics have been investigated. They are (i) selection of an arbitary number between 6 and 12; (ii) using a lower probability significance level equal to 0.1; (iii) using both lower and upper significance level (0.1 & 0.9); and (iv) plotting cummulative explained variance against number of harmonics. It has been found that all

CONCLUSION

Chapter - VI

CONCLUSION

The conclusions of this study are

i. If a user need to perform a number of processings, subprograms for the processings should be linked to the DBMS by a single program. This will permit all the processings to be performed on the same retrieved data.

ii. It is better to save the data on a separate disk file after missing data estimation is completed. This is because almost all analysis involve missing data estimation.

iii. The spline-fit interpolation shows greater accuracy compared to the weighted-average interpolation method of missing data estimation.

iv. The method based on plotting of commulative explained variance has been found appropriate for determining number of significant harmonics.

v. Out of the 26 harmonics the first harmonics may be taken as significant for all practical purposes involving analysis of weekly groundwater level data.

REFERENCES

REFERENCES

1. Master Plan Organization (MPO) (1984). Surfacewater availability Vol.II, National Water Plan Project. Ministry of Irrigation, Water Development and Flood Control.

2. Martin James(1976), Principles of Data Base Management, PRENTICE-HALL, INC., England Cliffs, New Jersey.

3. CODASYL (Conference of data Description Language)(1976)in in Principles of Data Base Management by James Martin.

4. WMO (1983) Application of Microcomputer for Primary Processing of Hydrological Data. A DBMS Package developed by VITUKI, of Hungary.

5. Kruglinski David (1983); Data Base Management System (A guides to management software).

6. Martin L. Kersten, (1983). Privacy and Security in information systems using programming language features. Information systems, Volume 8, No.2, 1983.

7. H. REX Harton (1981): Data Base Security-System Architecture. Information System Vol.6,No.1, 1981.

8. Harry Katzan, Jr. (1973) Computer data Security (Pratt Institute).

9. Davis Brian (1977). The selection of DATA Base software NCC (National Capating Centre Ltd. U.K.) Publications

10. Davis Brian (1980) DATA BASE in perspective NCC Publications.

11. Master Plan Organization (MPO) (1984) Groundwater availability volume III, National Water Plan Project. Ministry of Irrigation, Water Development and Flood Control.

12. Pennington H. Ralph (1970) Introductory Computer Methods and Numerical Analysis, Publisher MacMillion Company.

13. BWDB (1977) Groundwater Investigation in Dhaka,Tangail, Mymensingh, Sylhet, Comilla, Noakhali, Chittagong and Chittagong Hill Tracts, BWDB supply paper 395.

14. Yevjevich Vujicha (1972) Stochastic Process in Hydrology, Water Resources Publications, Fort Collins, Colorado,U.S.A.

15. Yevjevich Vujicha (1972) Structural Analysis Hydrologic time series. Hydrology Paper, Colorado State University, U.S.A.

16. Chatfield C (1975) The analysis of time series: An introduction Chapman and Hall Ltd., U.S.A.

17. Shen Hsieh Wen (1976) Stochastic Approaches to Water resources H.W.Shen,P.O. Box 606, Fort Collins, Colorado, U.S.A.

18. Gillett Billy,E. (1974) Statistical analysis of Water Temperature residuals, Water Resources Research (Vol.10, December, 1974) No.6.

APPENDIX - A

WMO DBMS

1./ GENERAL VIEW OF THE DATA BASE

1.1/INTRODUCTION

A data set can be regarded as a Data Base only if the triple system of fondamental criteria can be met i.e. the data system should have:
- structure
- exhaustivity
- non redundancy

The Data Base Management System (DBMS) should be able to meet all present and future demands of the User(s). Experience shows that to fulfill these requirements five fundamental operations must be performed by the DBMS:
- INITialization
- ADDition or Subtraction
- MODification
- SELection
- CLEANing

INIT operation opens the files of the Data Base and sets the starting value necessary for the further development of the data structure.
ADD operation carries out the insertion of new data records into the Data Base as well as their final deletion from the Data Base.
MOD operation modifies the contents of a Data Base data record
SEL operation selects the subset(s) of the whole information set which are required by a particular user or user application.
CLEAN operation removes those dummy areas of the Data Base created during the course of Data Base operations which may not be accessed by the fundamental operations listed above.

These operations can be executed either by independent programs (e.g. HDBMNT for 'INIT' or 'CLEAN') or by optional subprograms embedded in the processing or User's programs (e.g. TRN subprogram group for 'ADD' or 'MOD' or 'SEL').

The FORTRAN programming language was selected for the preparation of system programs and subprograms for the following reasons:
- whilst programming is more complicated with FORTRAN, results can be obtained in shorter execution times than would be required for a program written in BASIC.
- FORTRAN is extremely efficient for coding complex mathematical and scientific algorithms which are increasingly used during data analysis.

The following chapters describe how the triple system of fundamental criteria can be met.

1.2/THE STRUCTURE OF THE DATA SYSTEM

The following requirements have been identified for the structural development of the data system:

i) rapid retrieval of unique records

ii) ability to handle records of variable length

iii) ability to store data in the most compact form

Rapid retrieval is performed by the use of record identifiers. A new combination of hierarchical and pointer structures has been developed by the designer of present DBMS. The method of key-pointer tables had to be discarded, because sequential searching is unavoidable for hydrological time series data (even if the table searches are executed in memory). Moreover, the key-pointer tables would require permanent sorting. In this new design it is the series of addresses (i.e. the sequence of pointers) established by the record identifiers themselves which has been utilized. Traditional key-pointer methods result in either superfluously large, unutilizable storage areas in the system or the application of highly complex functions for the calculation of addresses.

Preconditions for the selected method of addressing are:

- numeric record identifiers are needed for simple manegement
- record identifiers should not enter the Data Base as keys but must be split up into digits and stored in a one-dimensional array, e.g. the triple group of keys:

10109	1983	21
station code	year	code of data type

enters the Data Base as a one-dimensional array (BLOCK):
/1,0,1,0,9,1,9,8,3,2,1/, where each digit occupies one word.

1.3/WRITING TO THE DATA BASE

The DBMS subroutine used for writing to the Data Base first obtains the record identifier described above and after checking its validity starts to establish the storage address, beginning with the first element of BLOCK array, in the following way:

- a record consisting of ten words is opened at the beginning of the Data Base file (following its header record). The starting address of the current free area of the file is then written into the position corresponding to the value of BLOCK(1) of the record This pointer indicates the starting address of the next address record.
- a new record of ten words is opened at the above starting address where the new address record is written. The starting location of the next address record is stored in the appropriate position (corresponding to the value of BLOCK(2)) of this new record and so on.
- after this series of operations, BLOCK(N) (i.e. the last element of the record identifier) is reached and N address records will have been written.
- the Nth record is twice the size of the other records i.e. twenty words. The first ten words of the final record are used for the last address record and the starting address of the data record is stored in the appropriate position (corresponding to the value of BLOCK(N)). The length of the data record (LZ) is stored in the same position of the second half of the final record.
- the data record itself is written to the Data Base.
- since the data has filled some part of the free storage area, the address of the next available storage location is written into the file header record to update the free storage area pointer in preparation for the next write operation.

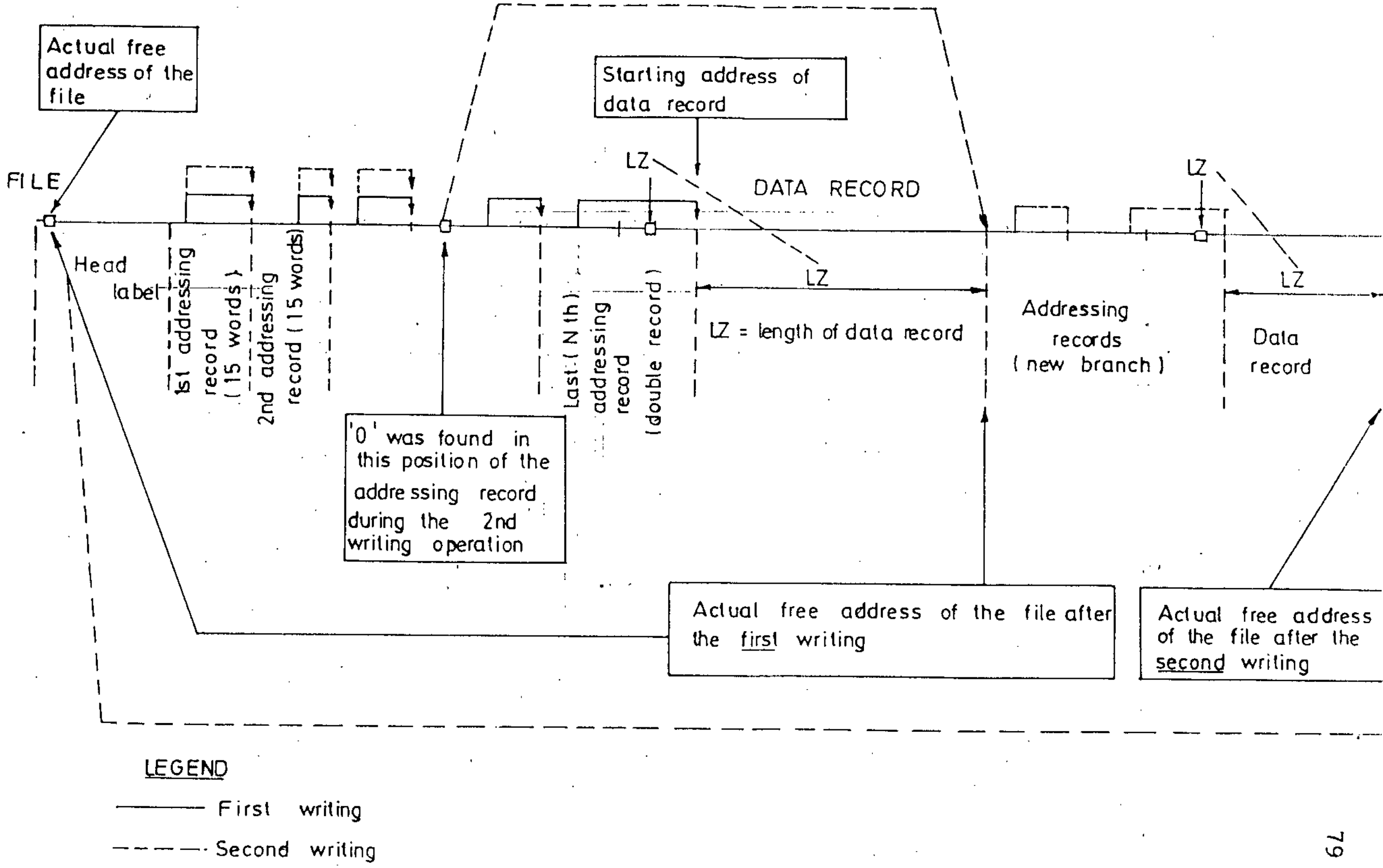
The next write operation attempts to use the sequence of pointers described so far. If this sequence (or chain) of pointers is interrupted i.e. the subroutine finds <zero> at the position corresponding to the value of BLOCK(I) of the Ith address record, the subroutine stores the starting address of the current free area of the file in this position and opens a new sub-sequence of address records. The last step of this operation is to write the new data record to the Data Base and to update the free storage area pointer in the file header record.

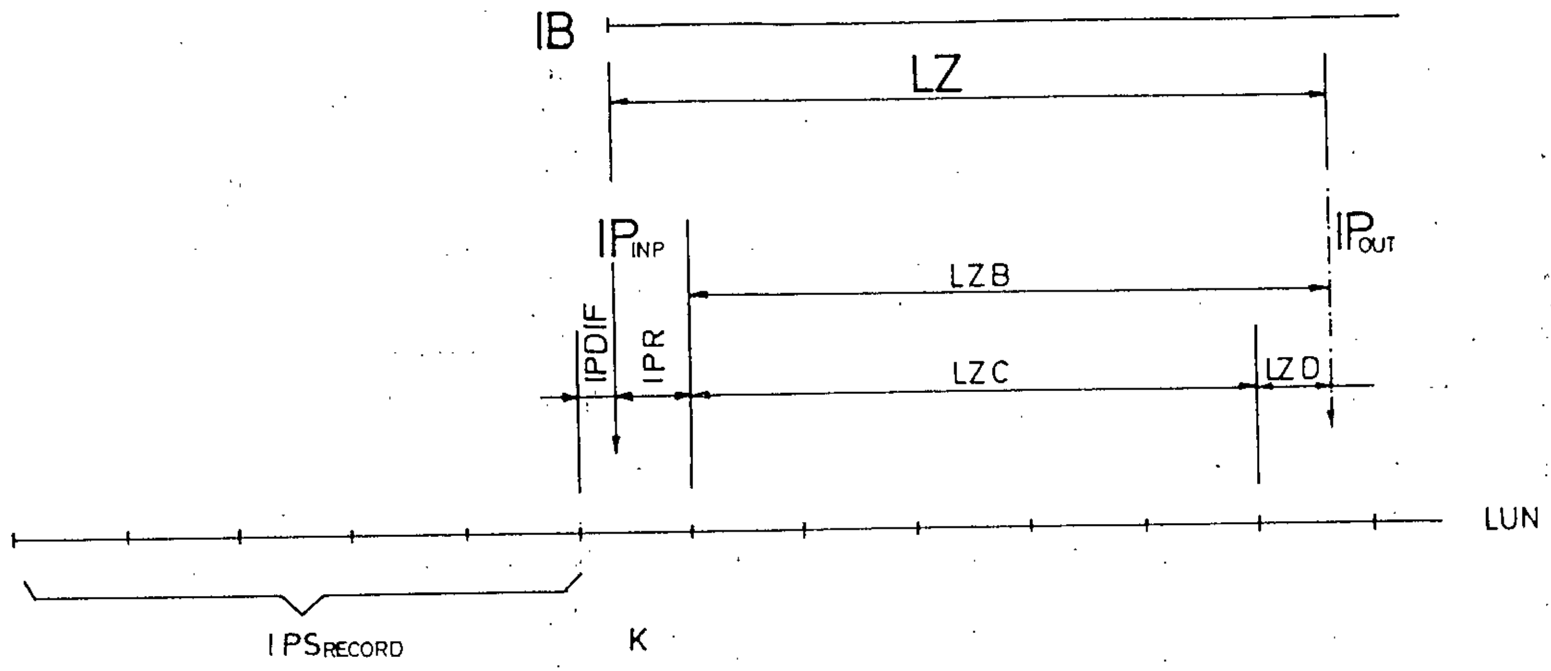
Studying the above operations it is apparent that address and data records are handled invisibly to the User which provides some level of system security. It should also be noticed that the longer are the data records, the smaller is the area used for addressing (i.e. as much data as possible should be inserted into one data record).

On the other hand, there are some disadvantages associated with the use of very long records. These disadvantages relocate to the number of I/O operations required.

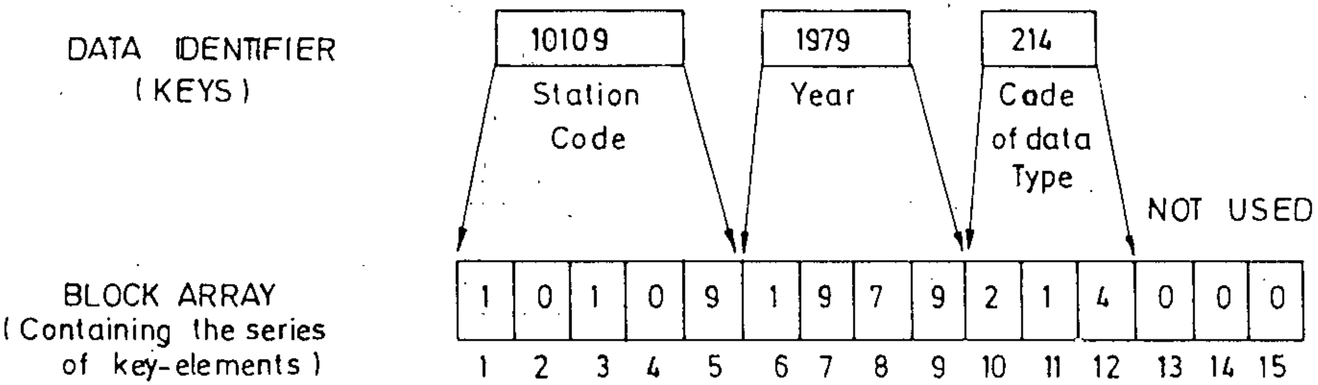
Taking into consideration all these factors, the STATION-YEAR RECORD has been selected as the basic structural unit of the Hydrological Data Base. A STATION-YEAR RECORD contains the entire set of observations for a particular station in a particular year. This unit has proved to be efficient for hydrological data.

When handling data in annual blocks the ratio between the storage area required for address and data records is approximatly 1 to 10. This ratio is much more efficient than for conventional key-pointer systems.

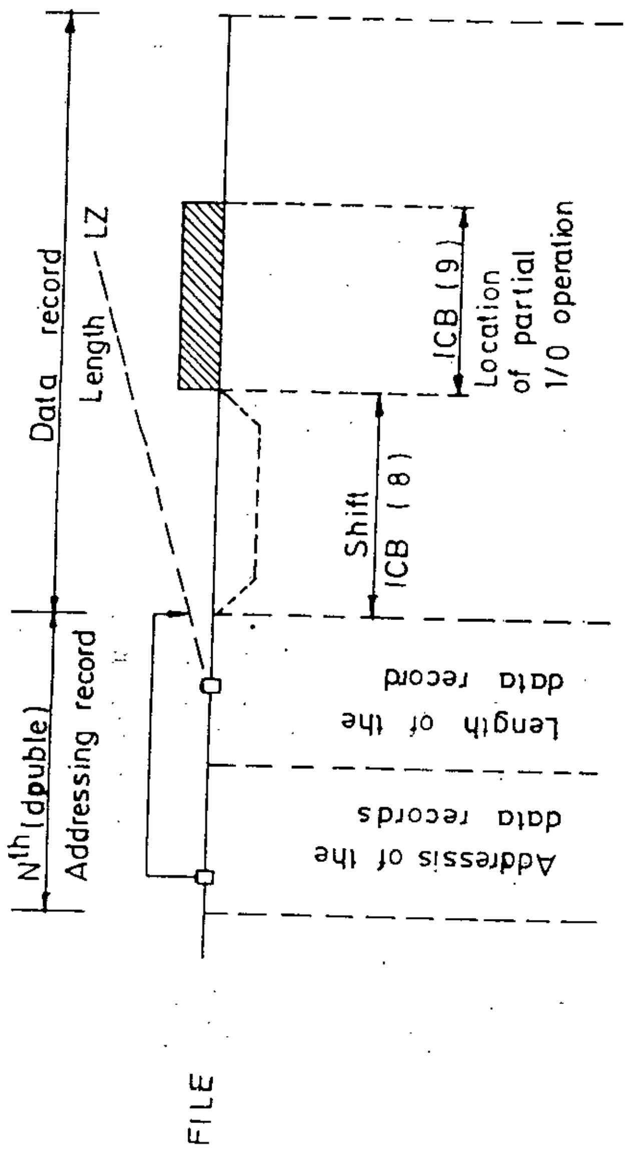

Actual free address of the file
FILE
Head label
1st addressing record (15 words)
2nd addressing record (15 words)
'0' was found in this position of the addressing record during the 2nd writing operation
Last (Nth) addressing record (double record)
Starting address of data record
LZ
DATA RECORD
LZ
LZ = length of data record
Addressing records (new branch)
LZ
LZ
Data record
Actual free address of the file after the first writing
Actual free address of the file after the second writing
LEGEND
First writing
Second writing

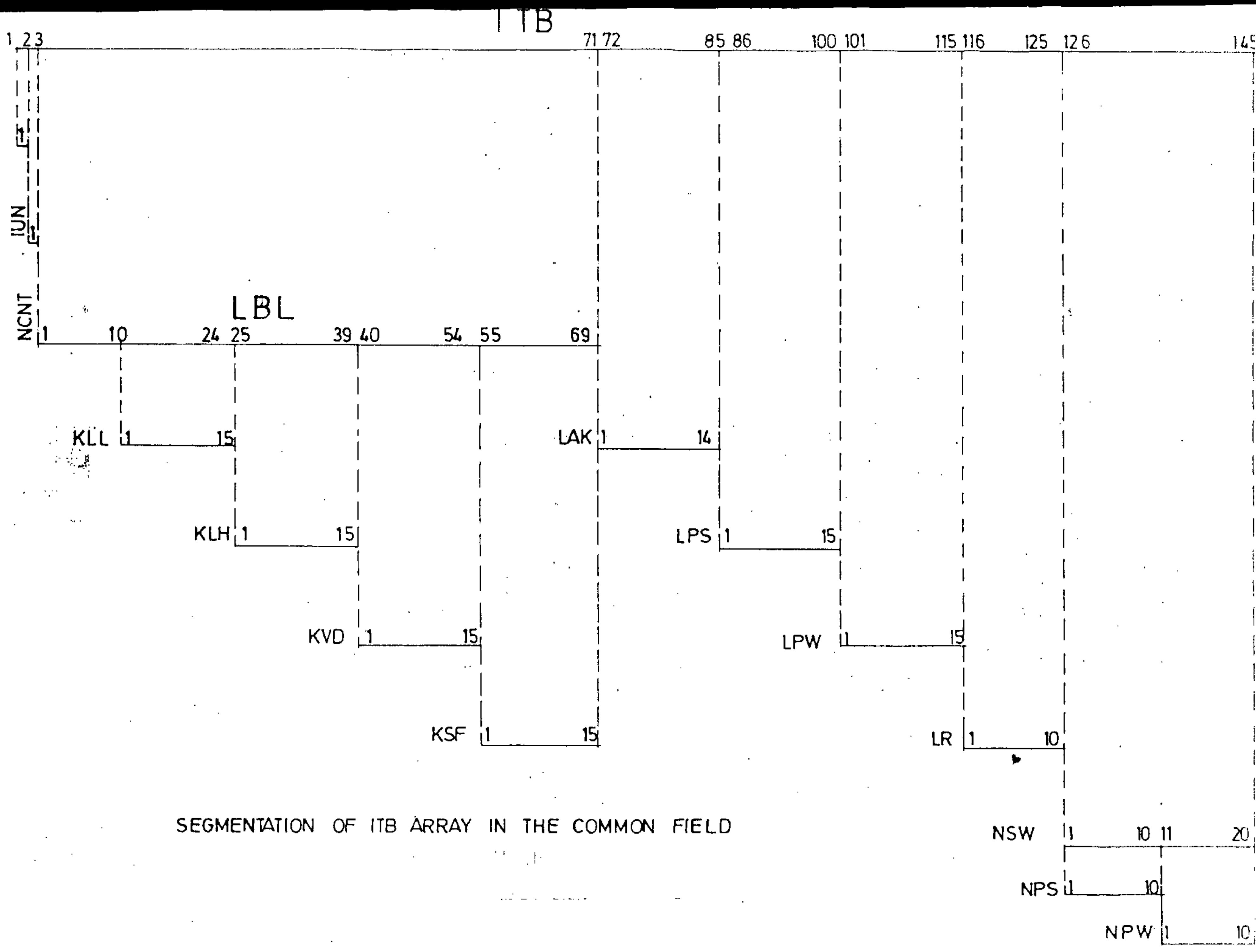


SIMULATION OF WORD ADDRESSING



EXAMPLE: if I = 7 → block(I) = 9


Data record
Length LZ
ICB (9)
Location of partial I/O operation
Shift ICB (8)
Nth (double) Addressing record
Length of the data record
Addressis of the data records
FILE



SEGMENTATION OF ITB ARRAY IN THE COMMON FIELD

1.4/READING FROM THE DATA BASE

The reading procedure also starts with an analysis of the BLOCK array containing key(s) split up to digits, and proceeds as follow:

- the subroutine first reads the first address record of the file and examines the word corresponding to the value of BLOCK(1) to find the starting location of the second address record.
- then reads the second address record and provides the starting location of the next address record and so on.
- when BLOCK(N) is reached the subroutine reads the Nth record with twenty words which contains the last address record (i.e. the address of the data record) and the length of the data record.
- finally, the data record is transferred into memory.

If the subroutine finds <zero> instead of an address in the appropriate position of an address record, it means that the requested data record does not exist in the Data Base (i.e. the necessary branch of address records has not yet been established).

1.5/DELETION FROM THE DATA BASE

The operation of deletion is easily understood from a knowledge of the writing and reading operations:

- a trial read is made using the identifier of the record to be deleted.
- starting from the last address record and moving backward through the chain of pointers, all the address records are examined.
- the first address record which contains more than one address must be written to the Data Base after reset to <zero> that position which corresponds to the value of the appropriate word of BLOCK array (i.e. the subroutine closes the only way to the data record).
- the total length of this data record and the length of the address record(s) increas the amount of dummy area of the Data Base file.

Before describing the subroutine used to select Data Base records, it is necessary to introduce the special procedures used to optimise the performance of subroutines which handle the data transfer.

i) The subprograms of the DBMS (TRN-group) have been prepared using the technique of structured programming. By this means the efficiencies inherent in using uniformed Data Base structure can be utilized to the maximum extent. The principle of this method is that the program modules also have a similar structure and the meaning of particular variables is the same in each modules. In this way both program control data and variables needed in several modules during execution of the main program, can be reduced into a single common data field.

All the information necessary for the Data Base file being processed can be stored in this COMMON field and there is no need to read it from the file before each particular operation.

ii) The Data Base uses a single logical unit i.e. single logical file but it was considered useful from a physical point of view to separate this file into smaller sub-units i.e. into physical files.

On the one hand this was determined by the need for data security and on the other hand by the consideration of file size.

Since operations are executed consecutively, it appeared at first that information contained in the COMMON field could be continuosly exchanged as different files were accessed. However, since the DBMS does not usually operate with more than three physical files simultaneounously, it was decided to create a COMMON field capable of storing the information for three physical files opened in parallel. The length of this COMMON field is 3 * 145 = 435 words and its detailed content is described below (see Section TRNBLD).

If the program has to use more than three physical files, a subroutine decides, by mean of a counter, which of the three sub-fields of the COMMON field will be used for the file control data of the incoming file (see Source program list of WSPLBL in TRNLIB/FOR). When a new file is accessed its file control data are loaded into the sub-field having the highest counter value, as this file has probably been open for the longest period.

iii) Based on practical experience, another optimizing operation has been introduced handling storage of the address record pointer chain in memory.

It was explained when describing I/O operations that the subroutines first read the address records. For the analysis of hydrological data some particular types of data retrieval are common. (e.g. observations of several years at a single station or observations of a particular year at several stations are required). In such cases some part of the record identifiers change, but the rest remains constant. Therefore, it is unnecessary to start reading from the very beginning of the address records for each I/O operation. The pointer chain (i.e. the starting addresses of the address records) and the elements (or digits) of the record identifier are stored in suitable memory buffers. The record identifier of the new request may then be compared, and searches on the Data Base need only be made for the element(s) of the record identifier which have changed.

Thus, the I/O subroutines have rapid access to the data records (after the necessary validity checking) if it is found that only the last few positions of the record identifier have changed.

iv) Because the Data Base is set up as a series of physical files, it must be pre-determined what range of key values is valid for each file. This problem has been solved by storing the upper and lower limits for each of the keys in the header record of the relevant file.
Thus, all of the TRN subroutines can check these limits to establish, whether a record identifier is valid for the physical file in question. The User will always see the record identifiers divided into coded key sub-fields (e.g. station code, year, code of data type), when executing any of the main programs.

It frequently happens that the upper and lower limits of different coded sub-groups are equal e.g.:

```
          min. YEAR  ---> 1 9 0 0
          max. YEAR  ---> 1 9 9 9
                          x x
```

It is unnecessary in this case to establish new address records for the elements having equal min. and max. values as this would result in superfluous repetition of the address record(s), wasting storage space on the Data Base. Such element(s) of the record identifiers therefore are not used in the addressing system. However, they are present in the key limits of the file and thus can be taken into account in the checking procedure. For this reason the DBMS recognises two levels of addressing:

- theoretical (which is apparent to the User)
- real (which is used internally by the DBMS)

1.7/THE SELECTION OPERATION

It has been shown that files are initialized with their upper and lower key limits and that they are defined in the system during execution by information in the COMMON sub-field. It is possible to change the contents of the COMMON sub-field by a special subroutine which is able to adjust both of the file limits. In the course of the selection operation the relevant subroutine can access only the records in the range of these adjusted limits. If both the upper and lower limits of the file are set to the key values for a particular station, the subroutine can not "see" the coding digits of the other stations. Therefore it is able to select the requested key sequence.

The selection operation is terminated when the subroutine is not able to generate new key elements (i.e. the subroutine reached the current upper limit).

1.8/THE MODIFICATION OPERATION

In such cases the start position and length of the modified data is specified in a control block (ICB) which is passed through to the Data Base access subroutines. Only that part of the data record defined by the control block is overwritten by modified data. The starting position effectively defines a shift in the starting address for the write operation compared to the starting address that would be used for operation on a complete data record.

2./ TRNLIB Subprograms of the Data Base Management System

2.1 NTRR and NTRW .

Purpose of the subprogram
Simulation of word addressing access
on a Direct Access file.
Transfer from/into the file into/from the memory.
NTRR for READING
NTRW for WRITING

Parameters
IB Buffer of the transfer

- OUT -

COMMON field(s)

```
/CNTR/   LUN    Logical unit number of the file
         IPS    Sector pointer
         IPW    Word pointer in the sector
                INP = Starting address of the
                         transfer on the file.
                OUT = Address of the word following
                         the zone to be transmitted on
                         the file.
         LZ     Number of words to be transmitted
                (Length of the record)
         IS     Working buffer
         LSPS   Last IPS
                value of IPS in preceding call
         LSLUN  Last LUN
                value of LUN in preceding call
         NWFG   Writing flag
                =0 no change in IS during last call
                =1 content of IS has been changed
                   during last call
                   IS is to write back before operation
```

Called subprogram(s) NONE

```
 Purpose of the subprogram
      Transfer of the data record from the memory
      area 'IB' into the file under the control of 'ICB'
      block.
      The data record is defined by 'KEYS' on the file.
      W R I T I N G
 Parameters
      ICB        Control block
      ICB (1)    Logical unit number of the file          - INP -
          (2)    Length of data record to be              - INP
                 transferred from the memory area 'IB'
                 into the file
          (3)    Number of key elements to be taken       - INP -
                 into consideration from 'KEYS' array
          (4)    NOT USED
          (5)    Error condition                          - OUT -
                 =0     Normal return - NO ERROR
                 .GT.0  Abnormal return - ERROR
                 =1     Logical unit number is erroneous
                        ICB(1).LT.6 or ICB(1).GT.10
                 =2     Number of key elements to be used
                        is not corresponding to the
                        addressing level of the file.
                        ICB(3).NE.LBL(8)
                 =3     Non numerical character in 'KEYS'
                        array or the requested key elements
                        are out of the file limits.
                 =4
                 =5     The description of the record length
                        is erroneous
                        ICB(2).LE.0
                 =6     Attempt to file overflow
                 =7
                 =8     In case of a partial transfer
                        /ICB(8).GT.0/ the description of the
                        zone to be transferred is erroneous
                        ICB(8)-1+ICB(9).GT.ICB(2), or
                        ICB(8).LT.0 or ICB(9).LT.0
                 =9     Attemt to overwrite an existing data
                        record on the file in prohibited case
                        /ICB(6).NE.0/
          (6)    Overwriting of an existing data record - INP -
                 on the file
                 =0     Permitted
                 .NE.0  Prohibited
          (7)    NOT USED
          (8)    Description of the transfer conditions  - INP -
                 =0      Complete transfer of data record
                         is requested. ICB(2) words are to be
                         transferred from the memory area 'IB'.
                 .LT.0   ERROR /See ICB(5)=8/
                 .GT.0   Partial transfer of the data record
                         is requested. The data record on the
                         file is to be written (or over
                         written) only from the ICB(8)-th
                         word. The transfer starts from
                         IB(1) in the memory.
                         ICB(2) must contain the total
```

```
                          length of the data record on the
                          file.
              (9)     Length of the zone of data record to    - INP -
                      be transferred in case of partial
                      transfer.
                      .LE.0 ERROR  /See ICB(5)=8/
                      .GT.0 Number of words to be transferred
                            from the memory area 'IB' into the
                            data record on the file.
         KEYS         Array of key elements                   - INP -
         IB           Buffer of data to be transferred from   - INP -
                      the memory into the file
     COMMON field(s)
         /CNTR/   See description in Section      'NTR'
         /CTRN/   See description in Section      'TRNBLD'
     Called subprogram(s)
          NTRR
          NTRW
```

2.3 TRNR(ICB,KEYS,IB)

Purpose of the subprogram

Transfer of the data record from the file into memory area 'IB' under the control of 'ICB' block.
The data record is defined by 'KEYS' on the file.
READING

Parameters

```
ICB        Control block
ICB (1)    Logical unit number of the file                  - INP -
    (2)    Length of data record to be                      - OUT -
           transferred into the memory (in words)
    (3)    Number of key elements to be taken               - INP -
           into consideration from 'KEYS' array
    (4)    NOT USED
    (5)    Error condition                                  - OUT -
           =0     Normal return - NO ERROR
           .GT.0  Abnormal return -  ERROR
           =1     Logical unit number is erroneous
                  ICB(1).LT.6 or ICB(1).GT.10
           =2     Number of key elements to be used
                  is not corresponding to the
                  addressing level of the file.
                  ICB(3).NE.LBL(8)
           =3     Non numerical character in 'KEYS'
                  array
           =4     The data record defined by 'KEYS' is
                  not existing on the file.
           =5
           =6
           =7
           =8     In case of a partial transfer
                  /ICB(8).GT.0/ the description of the
                  zone to be transferred is erroneous
                  ICB(8)-1+ICB(9).GT.LZ, or
                  ICB(8).LT.0 or ICB(9).LT.0
           =9
    (6)    NOT USED
    (7)    NOT USED
    (8)    Description of the transfer conditions         - INP -
           =0      Complete transfer of data record
                   is requested. Length of data record
                   is given in 'ICB(2)'.
           .LT.0 ERROR /See ICB(5)=8/
           .GT.0 Partial transfer of the data record
                   is requested. The data record zone
                   is to be transferred into
                   'IB' array from the ICB(8)-th word of
                   data record. ICB(2) contains the total
                   length of the data record on the
                   file.
    (9)    Length of the zone of data record to          - INP -
           be transferred in case of partial
           transfer.
           =0     KEY-READING
                  No transfer of data is to be
                  executed. The subprogram only checks
                  the presence of the data record on the
```

```
                           file. ICB(2) contains the total length
                           of the data record on the file.
                     .LT.0 ERROR  /See ICB(5)=8/
                     .GT.0 Number of words to be transferred
                           from the data record (i.e. from the
                           ICB(8)-th word of the record )
                           ICB(2) contains the total length of
                           the data record.
       KEYS          Array of key elements                   - INP -
       IB            Buffer of data to be transferred into   - OUT -
                     the memory.
   COMMON field(s)
       /CNTR/   See description in Section      'NTR'
       /CTRN/   See description in Section      'TRNBLD'
   Called subprogram(s)
        NTRR
```

2.4 TRNRDR(ICB,KEYS,IB)

Purpose of the subprogram

-Transfer of the data record from the file into memory area 'IB' under the control of 'ICB' block. (Same as TRNR if ICB(7)=0.)

-Simulation of sequential reading from the file. The subprogram creates automatically new key elements taking into consideration the key limits of the file existing in 'CTRN' COMMON field /'ITB'/.

READING

Parameters

```
ICB       Control block
ICB (1)   Logical unit number of the file              - INP -
    (2)   Length of data record to be                  - OUT -
          transferred into the memory (in words)
    (3)   Number of key elements to be taken           - INP -
          into consideration from 'KEYS' array
    (4)   Sequential number of the position in         - OUT -
          'KEYS' array (from the left) where
          the last change of the key elements was
          executed by the subprogram while
          creating a new key.
    (5)   Error condition                              - OUT -
          =0     Normal return - NO ERROR
          .GT.0  Abnormal return -  ERROR
          =1     Logical unit number is erroneous
                 ICB(1).LT.6 or ICB(1).GT.10
          =2     Number of key elements to be used
                 is not corresponding to the
                 addressing level of the file.
                 ICB(3).NE.LBL(8)
          =3     Non numerical character in 'KEYS'
                 array
          =4     The data record defined by 'KEYS' is
                 not existing on the file in case
                 ICB(7)=0 at calling time, otherwise
                 NOT USED.
          =5
          =6
          =7     End of file, no more key elements
                 are to be created.
                 /ICB(4)=ICB(3)/ and 'KEYS' array
                 contains the upper key limits of
                 the file.
          =8     In case of a partial transfer
                 /ICB(8).GT.0/ the description of the
                 zone to be transferred is erroneous
                 ICB(8)-1+ICB(9).GT.LZ, or
                 ICB(8).LT.0 or ICB(9).LT.0
    (6)   NOT USED
    (7)   Description of the starting condition         - INP -
          =0      Performs the same as TRNR.
          .GT.0   The search is to be started
                  with the first valid key elements
                  following the received ones.
    (8)   Description of the transfer conditions        - INP -
          =0      Complete transfer of data record
                  is requested. Length of data record
                  is given in 'ICB(2)'.
```

```
                        .LT.0 ERROR /See ICB(5)=8/
                        .GT.0 Partial transfer of the data record
                              is requested. The data record zone
                              is to be transferred into
                              'IB' array from the ICB(8)-th word of
                              data record. ICB(2) contains the total
                              length of the data record on the
                              file.
                   (9)  Length of the zone of data record to   - INP -
                        be transferred in case of partial
                        transfer.
                        =0    KEY-READING
                              No transfer of data is to be
                              executed. The subprogram gives only
                              the valid key elements in increasing
                              order from the file (KEYS array).
                              ICB(2) contains the total length
                              of the data record found.
                        .LT.0 ERROR  /See ICB(5)=8/
                        .GT.0 Number of words to be transferred
                              from the data record (i.e. from the
                              ICB(8)-th word of the record).
                              ICB(2) contains the total length of
                              the data record.
              KEYS      Array of key elements
                   IF ICB(7)=0 The same as for TRNR
                   IF ICB(7).GT.0
                        INP   Key elements for the start of the search
                        OUT   Key elements of the data record
                              found on the file.
              IB        Buffer of data to be transferred into  - OUT -
                        the memory.
       COMMON field(s)
              /CNTR/   See description in Section     'NTR'.
              /CTRN/   See description in Section     'TRNBLD'
       Called subprogram(s)
               NTRR
```

2.5 TRNDEL(ICB,KEYS)

```
    Purpose of the subprogram
         Deletion of existing data record from the file under t
         control of 'ICB' block.
    Parameters
         ICB           Control block
         ICB (1)       Logical unit number of the file         - INP -
             (2)       NOT USED
             (3)       Number of the key elements to be taken- INP -
                       into consideration from 'KEYS' array.
             (4)       NOT USED
             (5)       Error conditions                        - OUT -
                       =0     Normal return - NO ERROR
                       .GT.0 Abnormal return -  ERROR
                       =1     Logical unit number is erroneous
                              ICB(1).LT.6 or ICB(1).GT.10
                       =2     Number of key elements to be used
                              is not corresponding to the
                              addressing levels of the file.
                       =3     Non numerical element in 'KEYS'
                              array or the key elements are out
                              of the file limits.
                       =4     The requested data record (defined
                              by 'KEYS') is not existing on the
                              file.

        KEYS           Array of key elements to be deleted    - INP -
                       from the file.
    COMMON field(s)
         /CNTR/        See description in Section         'NTR'
         /CTRN/        See description in section         'TRNBLD'
    Called subprogram(s)
          NTRR
          NTRW
```

2.6 TRNLMN(ICB,KEYS)

```
   Purpose of the subprogram
        Reading information from the file Head Label.
        Modification of the file limits in CTRN-COMMON field
        for execution.
  Parameters
        ICB         Control block
        ICB (1)     Logical unit number of the file              - INP -
            (2)     NOT USED
            (3)     Number of key elements to be taken
                    into consideration from 'KEYS' array
                    - Reading                                    - OUT -
                    - Modification                               - INP -
            (4)     Request code                                 - INP -
                    =0    Free the file from 'ITB' field.
                    .LT.0 Read
                          =-1 Lower limit of the file
                          =-2 Upper limit of the file
                    .GT.0 Modify file limits in 'ITB' field
                          =1  Lower limit
                          =2  Upper limit
           (5)     Error condition                               - OUT -
                   =0     Normal return - NO ERROR
                   .GT.0  Abnormal return - ERROR
                   =1     Logical unit number is erroneous
                          ICB(1).LT.6 or ICB(1).GT.10
                   =2     Number of key elements to be used
                          is not corresponding to the
                          addressing level of the file.
                          ICB(3).NE.LBL(8)
                   =3     Non numerical character in 'KEYS'
                          array or the requested key elements
                          are out of the file limits.
                          In case of modification only.
                  =4
                  =5
                  =6
                  =7
                  =8
                  =9
                  =10     The request code is erroneous
                          ICB(4).LT.-2 or ICB(4).GT.2
      KEYS        Array of key elements
                  - Reading                                      - OUT -
                    if ICB(4)=-1  'KEYS' array contains
                    the Lower limit of the file
                    if ICB(4)=-2  'KEYS' array contains
                    the Upper limit of the file.
                  - Modification                                 - INP -
                    if ICB(4)=1   'KEYS' array contains
                    Lower limits for modification,
                    if ICB(4)=2   'KEYS' array contains
                    Upper limits for modification.
                  if ICB(4)=0   'KEYS' array is not used
 COMMON field(s)
      /CNTR/   See description in Section      'NTR'
      /CTRN/   See description in Section      'TRNBLD'
 Called subprogram(s)        NTRR
```

2.7 TRNERR(ICB,KEYS)

Purpose of the subprogram

In case of TRN error it displays 'ICB' and 'KEYS' arrays.

Parameters

ICB	Control block to be listed
KEYS	Array of key elements to be listed

COMMON field(s) NONE

Called subprogram(s) NONE

2.8 STOKEY(K,N,KEY,KEYS)

Purpose of the subprogram

Splits KEY content in digits and stores them from KEYS(K) in N word(s).

Parameters

K	Starting word index in KEYS to be stored	- INP -
N	Number of split digit(s) to store	- INP -
KEY	Variable to split and store in KEYS	- INP -
KEYS	Array to be stored	- INP,OUT -

COMMON field(s) NONE

Called subprogram(s) NONE

2.9 RCLKEY(K,N,KEY,KEYS)

Purpose of the subprogram

Reads N element(s) from KEYS array starting with KEYS(K) and stores decimal value in KEY

Parameters

K	Starting word index in KEYS	- INP -
N	Number of element(s) to read from KEYS	- INP -
KEY	Variable containing decimal value	- OUT -
KEYS	Array to be read	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

Purpose of the subprogram

BLOCK DATA subprogram for the TRN group of subprograms
This subprogram must be at the end of the main program calling one of the TRN subprograms.

COMMON field(s)

/CNTR/ See description in Section 'NTR'

/CTRN/ LUNL Lower limit of the logical unit number to be used.
(LUNL=6)

LUNH Upper limit of the logical unit number to be used.
(LUNH=10)

NCX Starting value for the test of the file to be released from 'ITB' field.
(NCX=-1)

LTF Subfield length in 'ITB' field (in words).
(LTF=117)

LTB Length of 'ITB' field (in words).
(LTB=351)

IXZ Starting value for backward searching of the free subfield in 'ITB' field.
(IXZ=234)

ITB COMMON field area for the storage of information concerning the files used.
Number of subfields: 3

KEY Array of key elements used during the program execution.(It contains the actually used key elements.)

NDI Auxiliary table for the minimum test of key elements requested.

NDS Auxiliary table for the maximum test of the key elements requested.

3./ SUPPLEMENTARY INFORMATION

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 1981
(or equivalent)

This package has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ MSCLIB Miscellaneos subprograms

This library contains all subprogram wich can be used by any program of the Data Processing System or by any new User's program.

1.1 REAL FUNCTION QFN(ICB,KEYS,IDAT,NSTG)

Purpose of the subprogram

Conversion of water level given in NSTG into discharge
Converted discharge will be given in M3/S
Error codes:

QFN = -991.0	requested day is between two validity periods of the RC-directory
QFN = -992.0	requested day is out of the range of validity periods
QFN = -993.0	requested water level is out of the range of RC

Parameters

ICB	Control block containing information for Rating Curve file (see TRNLIB/TXT)	- INP -
KEYS	Array of data identifier containing station code for rating curve search	- INP -
IDAT	Array containing date of requested day IDAT(1) ---> day (2) ---> month (3) ---> year	- INP -
NSTG	Water level in CM for wich discharge is to be converted	- INP -

COMMON field(s)

/CRCI/ See description in Section RCINP

Called subprogram(s)

RCINP
FSPLN

1.2 INTEGER FUNCTION IHFN(ICB,KEYS,IDAT,Q)

Purpose of the subprogram

Conversion of discharge given in Q into water level
Converted water level will be given in CM
Error codes:

IHFN = -991	requested day is between two validity periods of the RC-directory
IHFN = -992	requested day is out of the range of validity periods
IHFN = -993	requested discharge is out of the range of RC

Parameters

ICB	Control block containing information for Rating Curve file (see TRNLIB/TXT)	- INP -
KEYS	Array of data identifier containing station code for Rating Curve search	- INP -
IDAT	Array containing date of requested day IDAT(1) ---> day (2) ---> month (3) ---> year	- INP -
Q	Discharge given in M3/S for wich water level is to be converted	- INP -

COMMON field(s)

/CRCI/ See description in Section RCINP

Called subprogram(s)

RCINP
FSPLN

1.3 RCINP(ICB,KEYS,IDAT)

```
Purpose of the subprogram
     Search of valid rating curve on the Rating Curve file
     for a given day
Parameters
     ICB     Control block containing information           - INP -
             for the Rating Curve file (see TRNLIB/TXT)
     KEYS    Array of data identifier containing            - INP -
             station code for Rating Curve search
     IDAT    Array containing date of requested day         - INP -
             IDAT(1) ---> day
                 (2) ---> month
                 (3) ---> year
COMMON field(s)
     /CRCI/  IPV    number of line in IBT containing
                    validity period found for the given day
             IBT    two-dimensional array containing 10 lines
                    of the RC-directory
             NRC    number of rating curve found
                    for the given day
             RC     array containing the rating curve found
                    for the given day
             ERR    error codes of the search
                    =    0.0 no error in RC-search
                    = -991.0 see Sections @FN, IHFN
                    = -992.0 see Sections @FN, IHFN
Called subprogram(s)
     TRNRDR
     TRNERR
     STOKEY
     NTSD
```

1.4 HDBDYR(ICB,KEYS,IB,IQ,N,ITIME,IDATE,NFG)

```
Purpose of the subprogram
     Read daily data from the Data Base file for a given d
Parameters
     ICB     Control block containing information        - INP -
             for the Data Base file (see TRNLIB/TXT)
     KEYS    Array of data identifier containing         - INP -
             station code, year and data type code
             for data search
     IB      Array containing data read from Data Base - OUT -
             for the given day
     IQ      Byte array containing Quality Code(s)       - OUT -
             of data read for the given day
     N       Number of data in IB and IQ arrays          - OUT -
     ITIME   Time of observation if daily data was       - OUT -
             found for the given day
             byte left:  hour
             byte right: minute
             UNCHANGED if more than one data exist
             for the given day ( N > 1 )
     IDATE   Array containing date of requested day      - INP -
             IDATE(1) --> day
                  (2) --> month
                  (3) --> year
     NFG     Type of compression used on                 - INP -
             the Data Base file while inserting data
             = 1 ---> water level    (diff method)
             = 0 ---> precipitation   (repetation meth.
COMMON field(s)      NONE
Called subprogram(s)
     TRNR
     TRNERR
     STOKEY
     DCPI
     DCPB
```

```
1.5   DCP(ICB,KEYS,IB,MONTH,NDAY,N,JBI,JBQ,NFG)

      Purpose of the subprogram
           Reestablish data from compacted block
      Parameters
           ICB     Control block containing information          - INP -
                   for Data Base file (see TRNLIB/TXT)
           KEYS    Array of data identifier containing           - INP -
                   station code, year and data type code
                   for data search
           IB      Array contaning daily data of a year     - INP,OUT -
                   INP: data record read from Data Base
                   OUT: reestablished data series of the given year
           MONTH   Month                                         - INP -
                   = 0 reestablishing of yearly data is requested
                   > 0 reestablishing of daily data is requested
                       for the month MONTH
           NDAY    Day                                           - INP -
                   = 0 reestablishing of yearly data is requested
                   > 0 reestablishing of daily data is requested
                       for the day NDAY
           N       Number of data in JBI and JBQ arrays          - OUT -
                   if reestablishing of daily data is requested
                   ( N = 24, 48 or 96 )
           JBI     Array containing reestablished daily data - OUT -
           JBQ     Byte array containing Quality Codes for      - OUT -
                   daily data
           NFG     Type of compression used on                   - INP -
                   the Data Base file while data input
                   = 1 ---> water level
                   = 0 ---> precipitation
      COMMON field(s)           NONE
      Called subprogram(s)
           TRNRDR
           TRNERR
           STOKEY
           DCPI
           DCPB
```

1.6 DCPI(NCB,IBC,IBD)

```
Purpose of the subprogram
     Reestablishing compacted data block (INTEGER)
Parameters
     NCB     Control block of reestablishing
     NCB(1)  Length of IBD array in words                      - INP -
             NOTE: the subprogram uses the IBD(LENGTH+1)
             as a working area
        (2)  Current number of reestablished data              - OUT -
        (3)  Length of compacted block in words                - INP -
             The compacted block starts from IBC(1)
        (4)  Type of compression in IBC array                  - INP -
        (5)  Filler value to use in output series              - INP -
             ( see corresponding internal code
               Section CMPI )
        (6)  Initial value to use in output series             - INP -
        (7)  Code of missing data                              - INP -
     IBC     Array of compacted data                           - INP -
     IBD     Array of reestablished data                       - OUT -
COMMON field(s)      NONE
Called subprogram(s) NONE
```

1.7 CMPI(NCB,IBC,IBD).

Purpose of the subprogram

Compression of data series (INTEGER)

Parameters

NCB	Control block of the compression Has the same meaning as in Section DCPI except for NCB(2) --->	- INP -
	NCB(3) --->	- OUT -
IBC	Array of compacted data	- OUT -
IBD	Array of data series to be compacted	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

Method used for compression

According to the type of compression given in NCB(4),

NCB(4) = 0

the subprogram computes the repetition number of the consecutive data being equal in the input data series, and stores them in IBC;
depending on the value of data to be stored, the subprogram uses one or two bytes in IBC to store;
the repetition number is always stored in one byte in IBC, and in case this number is greater than 25, a new subfield will be started;

NCB(4) = 1

the subprogram computes the differences between the consecutive data in the input data series and performs the compression for these differences as above;
in this type of compression the first element of the input data series will be stored in IBC(1).

Used internal codes in compacted block:

127	preceding data are stored on two-byte-subfield(s)
126	preceding data are stored on one-byte-subfield(s)
125 to 101	preceding data or internal code is to be repeted <code>-100 times
+100 to -100	stored data (byte left) NOTE: -25600 .LE. data .LE. +25600 !
-101 to -125	filler value is to be repeted ABS<code>-100 times
-126	code of initial value
-127	code of missing data

1.8 DCPB(NCB,IBC,IBD)

```
Purpose of the subprogram
     Reestablishing compacted data block (BYTE)
Parameters
     NCB     Control block of reestablishing
     NCB(1) Length of IBD array in words                       - INP -
        (2) Current number of reestablished data               - OUT -
        (3) Length of compacted block in words                 - INP -
     IBC     Array of compacted data                           - INP -
     IBD     Array of reestablished data                       - OUT -
COMMON field(s)          NONE
Called subprogram(s) NONE
```

1.9 CMPB(NCB,IBC,IBD)

```
Purpose of the subprogram
     Compression of data series (BYTE)
Parameters
     NCB     Control block of the compression
             Has the same meaning as in DCPB
             except for NCB(2) --->                            - INP -
                        NCB(3) --->                            - OUT -
     IBC     Array of compacted data                           - OUT -
     IBD     Array of data series to be compacted              - INP -
COMMON field(s)          NONE
Called subprogram(s) NONE
Method used for compression
     The subprogram computes the repetition number of the
     consecutive data being equal in the input data series
     and stores them in IBC;
     in case the repetition number is greater than 25
     a new subfield will be started.
     Used internal codes in compacted block:
     127             not used
     126             if the length of the compacted block
                     is an odd number, the last byte
                     of the last word will contain this code
     125 to 101      preceding data is to be repeted
                     <code>-100 times
     +100 to -128    stored data
```

1.10 SPLINE(N,RC)

Purpose of the subprogram

Computation of the parameters of a SPLINE-function

Parameters

N	Number of points in RC $1 < N < 34$	- INP -
RC	Two-dimensional array containing INP: RC(1,I) ---> independent variable at the Ith point RC(2,I) ---> dependent variable at the Ith point OUT: RC(3,I) ---> curvature of the function at the Ith point	- INP,OUT -

COMMON field(s)   NONE

Called subprogram(s) NONE

1.11 REAL FUNCTION FSPLN(X,N,RC)

Purpose of the subprogram

Computation of the value of a SPLINE function

Parameters

X	Independent variable of the function	- INP -
N	Same as in Section SPLINE	- INF -
RC	Two-dimansional array containing the SPLINE parameters See description in Section SPLINE	- INP -

COMMON field(s)   NONE

Called subprogram(s) NONE

Mathematical method used in computation

If a function $Y = f(X)$ is given by N+1 points:
$X(0),\ X(1),\ X(2),\ \dots\ X(N)$
$Y(0),\ Y(1),\ Y(2),\ \dots\ Y(N)$
the subprogram approximates on the interval
$X(I-1), X(I) \quad (I = 1,\ 2,\ 3,\ \dots\ N)$
with a polynomial of degree 3, satisfying the following conditions:
$f(X(I)) = S(X(I))$, and the polynomials of the neighbouring intervals have equal 1st and 2nd derivatives.
The polynomial has the following form:

$$S(X) = M(I-1) \frac{(X(I)-X)^3 - L(I)^2 (X(I)-X)}{6\,L(I)} + Y(I-1) \frac{X(I)-X}{L(I)} +$$

$$+ M(I) \frac{(X-X(I-1))^3 - L(I)^2 (X-X(I-1))}{6\,L(I)} + Y(I) \frac{X-X(I-1)}{L(I)}$$

where: $L(I) = X(I) - X(I-1)$,
$M(I) = S''(X(I))$
$M(I-1) = S''(X(I-1))$

By solving the following simultaneous equations for $M(I)$
$(I = 1,\ 2,\ 3,\ \dots\ N-1)$

$$\frac{L(I)}{L(I)+L(I+1)} M(I-1) + 2\,M(I) + \frac{L(I+1)}{L(I)+L(I+1)} M(I+1) =$$

$$= 6\, \frac{\dfrac{Y(I+1)-Y(I)}{L(I+1)} - \dfrac{Y(I)-Y(I-1)}{L(I)}}{L(I)+L(I+1)}$$

and taking into consideration the following conditions:

$M(0) = 0.0$ and $M(N) = 0.0$,

the polynomial of degree 3 'S' can be defined.

1.12 FOPEN(LUNF,FNM,LRL)

Purpose of the subprogram

Open disc file asking for file name on the display and screening of open file name

Parameters

LUNF Logical unit number of the file to open - INP -

FNM Array containing: - INP,OUT -
INP: question for the file name (max.16 char.)
OUT: file name entered on the display

LRL Logical record length in bytes - INP -

COMMON field(s) NONE

Called subprogram(s)

OPEN (FORLIB)

1.13 QCD(Q,B)

Purpose of the subprogram

Rounding of discharge value to first three digits and conversion of rounded discharge digits into characters

Parameters

Q Discharge value in m3/s to convert - INP -
Qmax = 99999.0 m3/s
Qmin = 0.0 m3/s

B BYTE array containing converted discharge- OUT -

COMMON field(s) NONE

Called subprogram(s) NONE

Method used for rounding

Q (m3/s)	B(1) to B(7)
0.000	0.000
0.001	0.001
0.023	0.023
0.456	0.456
7.890	7.89
89.087	89.1
123.789	124.
4567.891	4568
12468.357	12500

1.14 JWRITE(IW,I,B)

Purpose of the subprogram

Split of I-value into digit(s) and
conversion of the split digit(s) into character(s)

Parameters

IW Length of subfield in B output array - INP -

I Value to be converted - INP -

B BYTE array containing converted I-value - OUT -
if I less than the subfield given in IW
the leading blanks will be converted into zero

1.15 INTEGER FUNCTION MAXDY(IDAT)

```
Purpose of the subprogram
     Computation of the number of days for a given month
Parameters
     IDAT   Array containing the date                             - INP -
            of the requested month
            IDAT(1) ---> day
                (2) ---> month
                (3) ---> year
COMMON field(s)       NONE
Called subprogram(s)  NONE
```

1.16 INTEGER FUNCTION MAXDYF(NYEAR)

```
Purpose of the subprogram
     Computation of the number of days of February
     for a given year
Parameters
     NYEAR  Requested year                                        - INP -
COMMON field(s)       NONE
Called subprogram(s)  NONE
```

1.17 INTEGER FUNCTION NDTCHK(IDAT)

```
Purpose of the subprogram
     Checking of a given date
     If the date is admissible ---> NDTCHK = 0
     If the date is wrong      ---> NDTCHK = 1
     Range of years used in the subprogram:
     1700 to 2100
Parameters
     IDAT   Date to be checked                                    - INP -
            IDAT(1) ---> day
                (2) ---> month
                (3) ---> year
COMMON field(s)       NONE
Called subprogram(s)
     MAXDY
```

1.18 INTEGER FUNCTION NTSD(IDAT1,IDAT2)

```
Purpose of the subprogram
     Comparison of two dates
     IDAT1 greater than IDAT2 ---> NTSD = 1
     IDAT1   equal to   IDAT2 ---> NTSD = 0
     IDAT1    less than IDAT2 ---> NTSD =-1
Parameters
     IDAT1  Date to be compared                                   - INP -
     IDAT2  Date to be compared                                   - INP -
            IDAT(1) ---> day
                (2) ---> month
                (3) ---> year
     COMMON field(s)       NONE
     Called subprogram(s)  NONE
```

1.19 INTEGER FUNCTION NDDF(IDAT1,IDAT2)

```
Purpose of the subprogram
     Computation of the difference between two dates in days
     NDDF = IDAT1 - IDAT2
Parameters
     IDAT1  Date to be decreased                                  - INP -
     IDAT2  Date to be subtracted                                 - INP -
            IDAT(1) ---> day
                (2) ---> month
                (3) ---> year
COMMON field(s)        NONE
Called subprogram(s)
     MAXDY
```

1.20 NDSHF(N,IDAT1,IDAT2)

```
Purpose of the subprogram
     Shifting of date by a given number of days
Parameters
     N      Number of days for shifting                           - INP -
     IDAT1  Date to be shifted                                    - INP -
     IDAT2  Shifted date                                          - OUT -
            IDAT(1) ---> day
                (2) ---> month
                (3) ---> year
COMMON field(s)        NONE
Called subprogram(s)
     MAXDY
```

IDAT1 => 1-4-1929

N=0

IDAT2 => 1-4-1929

N=25

IDAT2 => 26-4-1929

N=-5

IDAT2 => 27-3-1979

1.21 NREAD(IR,N,NP)

```
Purpose of the subprogram
     Search of numerical subfield(s) in the input string an
     conversion into INTEGER decimal value
     The subprogram recognises only the following characters:
     0 1 2 3 4 5 6 7 8 9 $
Parameters
     IR      Input string containing character(s)       - INP -
     N       INP: left byte  number of characters
                             to be taken into consideration
                             in IR array
                  right byte number of subfields
                             to search in IR
                             if <right byte> = 0  the subprog-
                             ram looks for 40 subfields
             OUT: number of subfields found in IR
                  <  0 --->  a $ (dollar) has been found after
                             ABS(N) subfield(s) in IR
                  =-41 --->  the same as above, but no sub-
                             field had been found
                  =-42 --->  arithmetic overflow in a subfield
                             ( NP(I) > 32767 )
                  =-43 --->  IR contains more subfields than
                             requested
     NP      Array containing the converted decimal     - OUT -
             value(s) of the numerical subfield(s)
COMMON field(s)      NONE
Called subprogram(s) NONE
```

1.22 RREAD(IR,N,R)

```
Purpose of the subprogram
     Search of numerical subfield(s) in the input string and
     conversion into REAL decimal value
     The subprogram recognises only the following characters:
     0 1 2 3 4 5 6 7 8 9 - + . $
Parameters
     IR      Input string containing character(s)       - INP -
     N       Same as in NREAD, except for checking of
             arithmetic overflow
     R       Array containing the converted decimal     - OUT -
             value(s) of the numerical subfield(s)
COMMON field(s)      NONE
Called subprogram(s) NONE
```

1.23 HREAD(IR,N,NB,TX)

Purpose of the subprogram

Search of Hollerith subfield(s) in the input string
The Hollerith subfield has to be between apostrophes
The subprogram recognises any valid character
If the $ (dollar) is on a Hollerith subfield,
the subprogram recognises it as the part of the text,
no error code will be given

Parameters

IR	Input string containing character(s)	- INP -
N	Same as in NREAD, except for: OUT: =-42 ---> Hollerith subfield is not closed by apostrophe . =-43 ---> not used by the subprogram	
NB	Array containing the length of the subfield(s) in bytes	- OUT -
TX	Array containing the content of Hollerith subfield(s) found in IR.	- OUT -

COMMON field(s) NONE
Called subprogram(s): NONE

1.24 NNREAD(K,IR,N,NP,I)

Purpose of the subprogram

Search of numerical subfield(s) in the input string IR
starting from Ith character up to the Kth character
Conversion of the content of subfield(s) into
INTEGER decimal value
The subprogram recognises only the following characters:
0 1 2 3 4 5 6 7 8 9 + - $

Parameters

K	Position No. of last character	- INP -
IR	Input string containing character(s)	- INP -
N	Same as in NREAD, except for INP: number of subfields to search in IR	
NP	Array containing the converted decimal value(s) of the numerical subfield(s)	- OUT -
I	INP.: position No. of the starting character OUT: position No. of the character following the last tested subfield	

COMMON field(s) NONE
Called subprogram(s) NONE

1.25 INTEGER FUNCTION NTCH(K,IR,N,NCH,I)

Purpose of the subprogram

Testing of input string for a given character set
If NCH(L) has been found ---> NTCH = L, otherwise N+1

Parameters

K	Same as in NNREAD	- INP -
IR	Same as in NNREAD	- INP -
N	Number of character(s) to be tested from the first not blank character	- INP -
NCH	Array containig the character set to be searched	- INF -
I	Same as in NNREAD	- INP,OUT -

COMMON field(s) NONE
Called subprogram(s) NONE

1.26 INTEGER FUNCTION MEAN(NCB,N,IB)

Purpose of the subprogram

Computation of the INTEGER mean value of a given INTEGER data series

If IB(I)=NCB(6) or NCB(7) the Ith data will be ignored

Parameters

NCB	See description in Section DCPI	- INP -
N	Length of the input data series	- INP -
IB	Array containing the input data series	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

1.27 REAL FUNCTION RMEAN(NCB,N,RB)

Purpose of the subprogram

Computation of the REAL mean value of a given REAL data series

If RB(I)=NCB(6) or NCB(7) the Ith data will be ignored

Parameters

NCB	See description in Section DCPI	- INP -
N	Length of the input data series	- INP -
RB	Array containing the input data series	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

1.28 ENCDI(I,LBUFF,FORM)

Purpose of the subprogram

Conversion of the INTEGER value given in I into character string

Parameters

I	Variable containing the value to be converted	- INP -
LBUFF	Array containing the converted character string	- OUT -
FORM	Array containing the FORMAT description	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

1.29 ENCDR(R,LBUFF,FORM)

Purpose of the subprogram

Conversion of the REAL value given in R into character string

Parameters

R	Variable containing the value to be converted	- INP -
LBUFF	Array contining the converted character string	- OUT -
FORM	Array containing the FORMAT description	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

1.30 ENCDT(N,TEXT,LBUFF)

Purpose of the subprogram

Transfer of bytes from an array to another

Parameters

N	Number of byte(s) to be transferred	- INP -
TEXT	Array containing the input string	- INP -
LBUFF	Array containing the output string	- OUT -

COMMON field(s) NONE

Called subprogram(s) NONE

1.31 EDTERR(N,TEXT)

Purpose of the subprogram

Transfer of text from a program to display preceded by '! ' and followed by the message: ' Press C to continue :' The subprogram recognises only the character 'C' for continue the execution

Parameters

N	Number of character(s) in output string	- INP -
TEXT	Array containing the output string	- INP -

COMMON field(s) NONE

Called subprogram(s) NONE

2./ SUPPLEMENTARY INFORMATION

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 19
                                 (or equivalent)

This package has been elaborated by VITUKI/BUDAPEST-Hungary (c)
                                  the 10th of September 1983

1./ Purpose of the program TRNERR

The program displays text explaining the error detected by a TRN subprogram. In case a main program using TRN subprogram(s) stops with a TRN error, the current content of the Control Block (ICB) and the array of data identifiers (KEYS) appear on the screen. The code of the detected error can always be found in ICB(5) (for more information see TRNLIB/TXT). For easier recognition, the error code is underlined.

After starting, the program prompts the User to enter the error code by screening: '? Error no '. Having entered an error code the appropriate error message is screened and the program stops.

2./ Available command NONE

3./ Used library NONE

4./ Instruction for linkage of the program

```
L80 TRNERR-N,TRNERR-E
```

5./ Used file NONE

6./ Supplementary information

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 1981
(or equivalent)

This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ Purpose of the program HLPCMD

This program displays text explaining the meaning of the available commands of a program. Due to the limited capacity of the memory, the programs using commands has not enough place to edit these texts by themself.

After starting, HLPCMD displays the available program names and prompts the User to enter the required one, by screening: '? HELP = '. Having entered a program name the appropriate list of command is screened and the program asks for new program name. The program stops if 'Q' is entered in place of the required program name.

2./ Available command   NONE

3./ Used library   NONE

4./ Instruction for linkage of the program

```
L80 HLPCMD-N,HLPCMD-E
```

5./ Used file   NONE

6./ Supplementary information

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 1981
(or equivalent)

This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ Purpose of the program HDBMNT

This program maintains Data Base file(s). During the evolution of the Data Base, files have to be initialized, copyed, dumped or the User can be interested about the state of the files. The program HDBMNT is to perform these tasks in dialogue mode. The possible User's answers are listed after the question in paranthesis, each of them divided by slashes. Default answer is always shown in brackets. In case of entering more than one numerical information in a line, an arbitrary number of spaces should be left between them (FREE FORMAT). The User's answers are controlled by the program, and in case of error an error message will be screened before repeating the question.

After starting, the program asks for the output device (display or line printer), then prompts the User by screening an * (asterisk) to enter command.

2./ Available commands

*I Initialization of a Data Base file.
The Data Base file has to always be initialized before using the TRN subprograms. These subprograms need especial information about the file during the I/O manipulation (for more detail see TRNLIB/TXT). Under the control of this command the User can enter these information wich will be written on the File Head Record (see below). When the program asks for the number of records to be reset to zero, the total record number of the file should be given, because, the TRN subprograms, even in case of writing, read the file first (see TRNLIB/TXT). The program enters supplementary information in the first record of the file, wich are used only in listing.

*L Listing of the Head Record of a Data Base file.
The command resulte the writing current information of the Head Record. The list is very useful for seeing the current state of the Data Base file, especially the Size of used part for the extension, or the Size of dummy area (see TRNLIB/TXT) for the CLEAN operation.

*D Dump of a Data Base file sector by sector.
Generally the Dump operation is not needed during the normal use of a Data Base file. However, the possibility of dump can be very useful at debugging time of the TRN system.

*C Copy of a Data Base file to another. (CLEAN operation)
This operation tranfers every data record of a Data Base file to another, sweeping the dummy areas of the input file. Before a Copy operation the output file has to be initialized, using *I command, and the limits of the data identifiers have to be the same as on the input file. The Size of the output file should be at least the same than the input file.

*F Quick edition of the content of a Data Base file. (FLASH)
This operation performs a quick edition of a Data Base file containing station-year records. The program prepaers edition tables marking with * (asterisk) the presence of the statio year records on the Data Base. The limits of years are rounded by the program in the way to make possible to edit maximum 50 years at a time.

*Q Exit the program.
This command results exiting the program and the control returns to TRSDOS.

3./ Used libraries

MNTLIB For structuring the program, different subprograms have been written for the different program parts. These subprograms can not be used in any other program.

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the program

```
L80 HDBMNT-N,HDBMNT,MNTLIB,MSCLIB-S,TRNLIB-S-E
```

5./ Used file in execution of the program

The program can use any Data Base file with the commands: *L, *D or *C. Under the control of *I command the given file name should not be in TRSDOS File Directory, the program will create it. If the program finds the file in the directory, the previous content of this file will be lost, the program reinitializes it after entering the number of records to be reset to zero.

Using *F command the file has to have four subfields in its data identifier with the following lengths and meaning, in order:

- station code	from 1 to 6 digits
- any key-subfield	3 digits (not used by the program)
- year	4 digits
- type of data	2 digits

Description of the Head Record of a Data Base file.

The first record of a Data Base file is for especial task, information are stored in about the file. The TRN subprograms consult, in every case this record, named: Head Record, before executing any operation on the Data Base.

On the words from 1 to 69 the necessary information are stored for the TRN subprograms, and this part of the record is the Head Label.

a.) The content of the Head Label is the following:

```
     LBL( 1)   Size of the file in sectors
        ( 2)   Position of the last valid word in the last sector
        ( 3)   Serial number of the sector containing the current
               free position (word) on the file
        ( 4)   Position of the current free word on the sector
        ( 5)   Size total of dummy area(s) on the file (sectors)
        ( 6)   Size total of dummy area(s) on the file (words)
        ( 7)   File condition word
               if LBL(7) .NE. 0 --> the file structure is damaged
        ( 8)   Total length of data identifier to be used on the
               file in digits ( max.: 15 )
        ( 9)   Total number of element(s) of the data identifier
               used in the address-system of the file
  from (10)    Lower limit of data identifiers used on the file
  from (25)    Upper limit of data identifiers used on the file
  from (40)    Validity of data identifier element(s)
               LBL(39+I) = 1 | Ith element of the data identifier
                             | participate on the address-system
                         = 0 | Ith element of the data identifier
                             | is a dummy-key
  from (55)    Subfield descriptor in the data identifier
               LBL(54+I) = 0 | subfield
                         = 1 | end of subfield on the Ith position
```

The second part of the Head Record contains some supplementary information about the Data Base file, as follow:

b.) Supplementary information about the Data Base file:

```
   IREC(101) Date of initialization: DAY
       (102)                         MONTH
       (103)                         YEAR
   IREC(121) Owner's name (in 16 characters)
```

6./ Supplementary information

```
Computer to be used for run:    TRS-80 MODEL II ( or equivalent )
Operating system to be used:    TRSDOS version 2.0a January 1, 1981
                                ( or equivalent )
This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
                                 the 10th of September 1983
```

1./ Purpose og the program STEDIT

This program maintains the Data Base file containing the basic characteristics of the stations. The program performs its task in dialogue mode. The possible User's answers are always listed after the question in paranthesis, each of them divided by slashes. Default answer is shown in brackets. In case of entering more than one information in a line, an arbitrary number of spaces should be left between them (FREE FORMAT). The User's answers are controlled by the program, and in case of error, an error message will be screened before repeating the question.

The new information enter first in a memory buffer, and by using another command (see *W) they can be transferred to the Station File.

After starting, the program asks for the name of the Station File, then prompts the User by an * (asterisk) to enter command.

2./ Available commands

*I Insertion of the basic characteristics for a new station.
Under the control of this command the stations (i.e. the basic characteristics) already being on the Data Base are protected against accidental overwriting.

*L Listing of basic characteristics of a station.
Entering this command, an edition of the basic characteristics will be produced for the requested station. Depending on the used subcommand, this list can be transferred either to the display (P subcommand), or to the line printer (L subcommand).

*D Deletion of a station from the Station File.
This command results the deletion of the requested station from the Station File.

*M Modification of basic characteristics of a station.
Under the control of this command the current content of the record of basic characteristics can be modified for the requested station. The program displays the Menu containing all valid two-letter-subcommands and prompts the User by screening ** (asterisks) to enter the requested one.
After entering the subcommand, the current value of the appropriate basic characteristic will be screened and the program asks for the modified value. The only possible exit from this command is to type $ (dollar) in the place of the subcommand.

*W Writing new information onto the Station File.
The command results the writing of basic characteristic(s) transiting in memory buffer onto the Station File. The maximum capacity of the transiting zone is enough to store the basic characteristics of one station.

*E Exit program with writing.
The command results the executing of a Write command and the control returns to TRSDOS.

*@ Exit program without writing.
The control returns to TRSDOS without writing the last content of the transiting zone onto the Station File.

Typing $ (dollar)-key results, in every case, to escape the current command and the line containing the $ will be ignored.

3./ Used libraries

STELIB For structuring the program, different subprogråms have been written for different program parts. These subprograms can not be used in any other program.

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the program

```
L80 STEDIT-N,STEDIT,STELIB,MSCLIB-S,TRNLIB-S-E
```

5./ Used file in execution of the program

HDB/STN Station File

Type of the file:	direct access
Physical record length:	256 bytes
I/O manipulation:	with TRN-system
Form of record identifier:	XXXX
Size of key-subfield (digits):	1-6
Name of key-subfield:	station code

The program creates the following record for each station:

. Record of basic characteristics

```
- fixed length record containing
  mixed (REAL,INTEGER,HOLLERITH) values
  logical record length: 64 words
  words  1 to 10: name of the station                 (HOLL)
        11 to 20: name of the river                   (HOLL)
        21 to 25: previous identification of the station
                                                      (HOLL)
        26 to 30: name of operating institute         (HOLL)
              31: year of establishment of the station
                                                       (INT)
              32: coordinate latitude in  degrees      (INT)
        33 to 34:                         minutes     (REAL)
              35:            longitude in degrees      (INT)
        36 to 37:                         minutes     (REAL)
        38 to 39: elevation of the station            (REAL)
                  in meters above M.S.L.
        40 to 41: size of the catchment area          (REAL)
                  in square km2
        42 to 43: distance from the mouth             (REAL)
                  in kilometers
        44 to 64: NOT USED
```

6./ Supplementary information

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 1981 (or equivalent)

This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ Purpose of the program RCEDIT

This program enters discharge rating curves from the key board to the Data Base. Data input is performed in dialogue mode. The possible User's answers are always listed after the question in paranthesis, each of them divided by slashes. Default answer is shown in brackets. In case of entering more than one data in a line, an arbitrary number of spaces should be left between them (FREE FORMAT). The User's answers are controlled by the program and in case of error, an error message will be screened before repaeting the question.

The rating curves have to be entered in form of water level (in CM) and discharge (in M3/S) data couples. The steady rating curve has to be elaborated before entering.

The program can accept 99 different rating curves for each station, and every rating curve can have one or several validity period. The total number of validity periods is 990 for a station. The program accepts only one rating curve for a given period, no interpolation between two rating curves is possible.

A rating curve has to be given at least by 2, and at most by 33 data couples. Both the water level and the discharge have to be entered in increasing order. After entering the whole rating curve the program computes parameters of a SPLINE-function and stores them in the Data Base. The rating curve will be valid only between the lower and higher water levels given in input, it can not be extrapolated (for more information see MSCLIB/TXT Section SPLINE).

The rating curve and its computed parameters enter first in a memory buffer, and using another command (see *W) they can be transferred onto the Data Base.

After starting, the program asks for the used files, then prompts the User by screening an * (asterisk) to enter command.

2./ Available commands

*I Insertion of a new rating curve.
Under the control of this command the rating curves previously stored on the Date Base are protected against accidental overwriting.

*L Listing of a rating curve or the rating curve directory.
The list will be edited on the line printer.
Tabulated rating curves can be produced either for the entered water level and discharge couples, or by the given increment of water level.

*P Listing of a rating curve or the rating curve directory.
The list will be edited on the screen.
This command has the same effect as the *L command.

*D Deletion of an existing validity period from the directory.
Under the control of this command the rating curve directory is screened automatically, line by line (i.e.validity period), and after each the program asks the User for Deletion. If no Deletion is required the screened line can be skipped by ENTER key. When all validity period is deleted for a rating curve, the use of *W command results the deletion of the rating curve itself from the Data Base.

*M Modification of the rating curve directory.
Under the control of this command two subcommands can be used:
M-subcommand
Modification of existing validity period in the rating curve directory.
The rating curve directory is screened automatically, line by line (i.e. validity period), and after each the program asks the User for Modification. The User has to enter the whole modified period.
N-subcommand
Allows the User to enter a new validity period for an existing rating curve.

*W Writing new information onto the Data Base.
The command results the writing of rating curve and validity period transiting in memory buffer onto the Data Base. The maximum capacity of the transiting zone is enough to store one rating curve and one validity period.

*E Exit program with writing.
The command results the executing a Write command and the control returns to TRSDOS.

*@ Exit program without writing.
The control returns to TRSDOS without writing the last content of the transiting zone onto the Data Base.

Typing $ (dollar)-key results, in every case, to escape the current command and the line containing the $ will be ignored.

3./ Used libraries

RCELIB For structuring the program, different subprograms have been written for different program parts. These subprograms can not be used in any other program.

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the program

```
L80 RCEDIT-N,RCEDIT,RCELIB,MSCLIB-S,TRNLIB-S-E
```

5./ Used file in execution of the program

HDB/STN Station file

See description in STEDIT/TXT

HDB/RTC File of discharge rating curves

Type of the file:	direct access		
Physical record length:	256 bytes		
I/O manipulation:	with TRN-system		
Form of record identification:	XXXX	XX	XX
Size of key-subfields (digits):	1-6	2	2
Name of key-subfields:	station code	RC No.	Seq.No. of RC-directory parts

The program creates three types of record:

a.) Number of lines (i.e.validity periods) in RC-directory (<RC No.> = 00 and <Seq.No.> = 00)

- the length of this record = 1 word.

b.) Record of rating curves and its parameters (<RC No.> .GT. 00 and <Seq.No.> = 00)

- variable length record containing REAL values if a record contains N couples of RC-point with their computed parameter, the Ith group will contain
  RC(3*I-2): water level in CM
  RC(3*I-1): discharge in M3/S
  RC(3*I): curvature of RC in the Ith point

```
        c.) Record of RC-directory par
                          ( <RC No.> = .GT. 00 )

            - 10 lines of the RC-direc
              9 * 10 INTEGER words
              if <Sec.No.> = 01 the re
              first ten lines of the R
              the content of a line is
              word 1= starting date of    IDAY
                   2=                     MONTH
                   3=                     IYEAR
                   4= final    date of    IDAY
                   5=                     MONTH
                   6=                     IYEAR
                   7= number of RC
                   8= lower  limit of     IHMIN
                   9= higher limit of     IHMAX

HDB/WRK  work file used for transiting

        Type of the file:
        Physical record length:
        I/O manipulation:

        NOTE: this file has to be rese   first use
              The number of records ne
```

The easy use of the rating curves in a program is ensured by the QFN function. The following sample program shows the method of using this function:

SAMPLE PROGRAM:

```
  1.        BYTE BQ, JBQ
  2.        DIMENSION ICB(10), KEYS(15), IB(564),
  3.       /              IBI(31,12), BQ(31,12), JBI(96), JBQ(96),
  4.       /              JCB(10), IDT(3), Q(31,12), QD(96)
  5.        EQUIVALENCE (IBI(1,1),IB(1)), (BQ(1,1),IB(374)),
  6.       /                (NDAY,IDT(1)), (MONTH,IDT(2)), (NYEAR,IDT(3))
                .
                .
                .
  7.        CALL TRNRDR(ICB,KEYS,IB)
  8.        IF(ICB(5).EQ.0) GO TO 1
  9.        CALL TRNERR(ICB,KEYS)
 10.        STOP
 11.      1 CALL DCP(ICB,KEYS,IB,0,0,N,JBI,JBQ,1)
                .
                .
                .
 12.        DO 20 MONTH=1,12
 13.        DO 10 NDAY=1,31
 14.        IF(BQ(NDAY,MONTH).LT.0) GO TO 5
 15.        Q(NDAY,MONTH)=QFN(JCB,KEYS,IDT,IBI(NDAY,MONTH))
                .
                .
                .
 16.        GO TO 10
 17.      5 CALL DCP(ICB,KEYS,IB,MONTH,NDAY,N,JBI,JBQ,1)
 17.        DO 8 I=1,N
 18.        QD(I)=-900.
 19.        IF(JBI(I).EQ.-999) GO TO 8
 20.        IF(JBI(I).EQ.-998) GO TO 8
 21.        QD(I)=QFN(JCB,KEYS,IDT,JBI(I))
 22.      8 CONTINUE
                .
                .
                .
 23.     10 CONTINUE
 24.     20 CONTINUE
                .
                .
```

Lines 7. to 14. provide the water levels from the Data Base (for more information see STAGES/TXT).

The NYEAR variable and JCB array are set in the program part before the Line 15.

Line 15. converts the daily water level to discharge on the date given in IDT array. JCB array contains information concerning the rating curve file. KEYS array will be unchanged.

Lines 17. to 22. convert all water level of the day to discharge. QD array will contain -900.0 in case of no observation at a time step.

6./ Supplementary information

```
Computer to be used for run:    TRS-80 MODEL II ( or equivalent )
Operating system to be used:    TRSDOS version 2.0a January 1, 198
                                ( or equivalent )
This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
                                  the 10th of September 1983
```

1./ Purpose of the program STAGES

This program enters water levels from the key board to the Data Base. Data input is performed in dialogue mode. The possible User's answers are always listed after the question in paranthesis, each of them divided by slashes Default answer is shown in brackets.
In case of entering more than one data in one line, an arbitrary number of spaces should be left between them (FREE FORMAT). The User's answers are controlled by the program and in case of error, an error message will be screened before repeating the question.

The program can work in two modes of operation:

- in INSERT mode new data can be entered and
- in MODIFY mode existing data can be modified or deleted.

Water levels have to be entered in centimeters (CM). The program checks first the input data for lower and upper limits, then the difference between the input data and the immediatly preceding one for the limits relative to the differences.

Data can be entered and stored in four different time steps, depending on the 'TIME STEP CODE' chosen by the User. The possible TIME STEP CODEs are the followings:

0	for DAILY
1	60'
2	30'
3	15'

In case the frequency of observations changes in time the most appropriate time step has to be chosen and data can be entered at the time of observation (with a precision of 15 minutes). The other time steps are to skip by ENTER key.

The program can accept QUALITY CODE for each data. This is not produced by the program itself, it is the User's duty. The program recognises the following QUALITY CODEs:

blank	for Correct data
D	Doubtful data
E	Estimated data
U	Unusable data

The input data enter first in a memory buffer and in a working file (HDB/WRK), and using another command (see *W) they can be transferred onto the Data Base.

After starting, the program asks for the used files, then prompts the User by screening an * (asterisk) to enter command.

2./ Available commands

*I Insertion of new water levels.
This mode of operation protects the data already being on the Data Base against accidental overwriting.

*M Modification or Deletion of water levels.
Under the control of this mode of operation existing data can be corrected or deleted. The program makes automatic time step, data by data, and if no correction is required the screened data can be skipped by ENTER key. The 'D' subcommand deletes not only the screened data but each of the current day. In case the time step is to be changed, the whole day must be deleted first with 'D' subcommand, then the new time step and data can be inserted using *I command.

*W Write new information onto the Data Base.
The command results the writing of data transiting in memory buffer and on the work file onto the Data Base. The maximum capacity of the transiting zones is enough to store all data of one year (within 15 minutes).

*E Exit program with writing.
The command results the executing a Write command and the control returns to TRSDOS.

*@ Exit program without writing.
The control returns to TRSDOS without writing the last content of the transiting zones onto the Data Base.

Typing $ (dollar)-key results, in every case, to escape the current command and the line containing the $ will be ignored.

3./ Used libraries

STGLIB For structuring the program, different subprograms have been written for the different program parts. These subprograms can not be used in any other program.

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the program

```
L80 STAGES-N,STAGES,STGLIB,MSCLIB-S,TRNLIB-S-E
```

5./ Used files in execution of the program

HDB/STN Station file

See description in STEDIT/TXT

HDB/SGW Current data file of water levels

Type of the file:	direct access			
Physical record length:	256 bytes			
I/O manipulation:	with TRN-system			
Form of record identification:	XXXX	XXX	XXXX.	XX
Size of key-subfields (digits):	1-6	3	4	2
Name of key-subfields:	station code	stack No.	year	type

The code of data type is set to 10 in the program.
(See BLOCK DATA: NDTYP)

The program creates two types of record:

a.) Record of daily data of a year (<stack No.> = 000)

- daily water levels in CM — 1 to 372. words
  31 * 12 INTEGER words
  initial value: -998
  code for missing data: -999
  if more than one data exists for a day,
  the appropriate word contains:
  <stack No.> * 10 + <time step code>,
  where <stack No.> is the number of the stack
  containing data for this day.
- quality codes for daily data — 374 to 559. words
  31 * 12 BYTES
  initial value: blank
  code for missing data: blank
  if more than one data exists for a day,
  the appropriate byte contains:
  -INT(<starting address> / 24) - 1
  where <starting address> is the starting
  position of the data for this day in the stack
- time of daily observation — 560. word
  byte left: hour
  byte right: minute
- control words — 561 to 564. words
  for CMP and DCP subprograms

b.) Record of compacted water levels and quality codes
/ STACK / (000 .LT. <stack No.> .LE. 366)

- length of compacted water level block in words — 1. word
- length of compacted quality code block in words — 2. word
- compacted water levels and quality codes — 3 to 240.words

HDB/WRK work file used for transiting the input data

```
          Type of the file:                 direct access
          Physical record length:           256 bytes
          I/O manipulation:                 FORTRAN I/O

          NOTE: this file must be reset to zero before its first use
                The number of records needed = 300
```

Data are compacted on the Data Base due to the limited capacity of the available data storages. The operation of compression and its invers are done by the help of special subprograms (for more information see MSCLIB/TXT). The following sample program shows the method of using these subprograms:

SAMPLE PROGRAM:

```
  1.         BYTE BQ, JBQ
  2.         DIMENSION ICB(10), KEYS(15), IB(564),
  3.        /              IBI(31,12), BQ(31,12), JBI(96), JBQ(96)
  4.         EQUIVALENCE (IBI(1,1),IB(1)), (BQ(1,1),IB(374))
               .
               .
               .
  5.         CALL TRNRDR(ICB,KEYS,IB)
  6.         IF(ICB(5).EQ.0) GO TO 1
  7.         CALL TRNERR(ICB,KEYS)
  8.         STOP
  9.       1 CALL DCP(ICB,KEYS,IB,0,0,N,JBI,JBQ,1)
               .
               .
               .
 10.         DO 20 MONTH=1,12
 11.         DO 10 NDAY=1,31
 12.         IF(BQ(NDAY,MONTH).LT.0) GO TO 5
               .
               .
               .
 13.         GO TO 10
 14.       5 CALL DCP(ICB,KEYS,IB,MONTH,NDAY,N,JBI,JBQ,1)
               .
               .
               .
 15.      10 CONTINUE
 16.      20 CONTINUE
               .
               .
               .
```

Line 5. reads the record of daily data of the year given in KEYS array. ICB and KEYS arrays are set in the program part between Line 4. to Line 5.

Line 6. transfers the control to the label 1 in case of correct reading. Otherwise the program stops via Line 7. to Line 8.

Line 9. restores daily data from IB array (compacted input) and outputs them in IB. Since the fourth and fifth parameters are 0 the arrays ICB, KEYS, JBI and JB@ are not used. N will contain the time of observation valid for the whole year. The last parameter indicates the type of compression of IB array in input.

Line 12. transfers the control to the label 5 in case there is more than one data for the day NDAY of the month MONTH. Otherwise IBI(NDAY,MONTH) contains the water level and B@(NDAY,MONTH) the quality code of the given day

Line 14. restores all data of the day NDAY of the month MONTH from the appropriate STACK to JBI and JB@ arrays. N will contain the number of data in JBI and JB@ (N = 24 or 48 or 96).

6./ Supplementary information

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 1981
(or equivalent)
This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ Purpose of the program PRECIP

This program enters precipitations from the key board to the Data Base. Data input is performed in dialogue mode. The possible User's answers are always listed after the question in paranthesis, each of them divided by slashes. Default answer is shown in brackets.
In case of entering more than one data in one line, an arbitrary number of spaces should be left between them (FREE FORMAT). The User's answers are controlled by the program and in case of error, an error message will be screened before repeating the question.

The program can work in two modes of operation:

- in INSERT mode new data can be entered and
- in MODIFY mode existing data can be modified or deleted.

Precipitations have to be entered in tenths of millimeters. The program checks the input data for upper limit given by the User.

Data can be entered and stored in four different time steps, depending on the 'TIME STEP CODE' chosen by the User. The possible TIME STEP CODEs are the followings:

0	for	DAILY
1		60'
2		30'
3		15'

In case the frequency of observations changes in time the most appropriate time step has to be chosen and data can be entered at the time of observation (with a precision of 15 minutes). The other time steps are to skip by ENTER key.

The program can accept QUALITY CODE for each data. This is not produced by the program itself, it is the User's duty. The program recognises the following QUALITY CODEs:

blank	for	Correct data
D		Doubtful data
E		Estimated data
U		Unusable data

The input data enter first in a memory buffer and in a working file (HDB/WRK), and using another command (see *W) they can be transferred onto the Data Base.

After starting, the program asks for the used files, then prompts the User by screening an * (asterisk) to enter command.

2./ Available commands

*I Insertion of new precipitations.
This mode of operation protects the data already being on the Data Base against accidental overwriting.

*M Modification or Deletion of precipitations.
Under the control of this mode of operation existing data can be corrected or deleted. The program makes automatic time step, data by data, and if no correction is required the screened data can be skipped by ENTER key. The 'D' subcommand deletes not only the screened data but each of the current day. In case the time step is to be changed, the whole day must be deleted first with 'D' subcommand, then the new time step and data can be inserted using *I command.

*W Write new information onto the Data Base.
The command results the writing of data transiting in memory buffer and on the work file onto the Data Base. The maximum capacity of the transiting zones is enough to store all data of one year (within 15 minutes).

*E Exit program with writing.
The command results the executing a Write command and the control returns to TRSDOS.

*Q Exit program without writing.
The control returns to TRSDOS without writing the last content of the transiting zones onto the Data Base.

Typing $ (dollar)-key results, in every case, to escape the current command and the line containing the $ will be ignored.

3./ Used libraries

PRCLIB For structuring the program, different subprograms have been written for the different program parts. These subprograms can not be used in any other program.

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the program

```
L80 PRECIP-N,PRECIP,PRCLIB,MSCLIB-S,TRNLIB-S-E
```

5./ Used files in execution of the program

HDB/STN Station file

See description in STEDIT/TXT

HDB/PCW Current data file of precipitations

```
Type of the file:                  direct access
Physical record length:            256 bytes
I/O manipulation:                  with TRN-system
Form of record identification:     XXXX   XXX   XXXX   XX
Size of key-subfields (digits):     1-6    3     4      2
Name of key-subfields:             station
                                   code   stack
                                           No.   year
                                                        type
```

The code of data type is set to 20 in the program.
(See BLOCK DATA: NDTYP)

The program creates two types of record:

a.) Record of daily data of a year (<stack No.> = 000)

```
- daily precipitations in tenths of MM 1 to 372. words
  31 * 12 INTEGER words
  initial value: -998
  code for missing data: -999
  code for rain in trace: -100
  if more than one data exists for a day,
  the appropriate word contains:
  <stack No.> * 10 + <time step code>,
  where <stack No.> is the number of the stack
  containing data for this day
- quality codes for daily data        374 to 559. words
  31 * 12 BYTES
  initial value: blank
  code for missing data: blank
  if more than one data exists for a day,
  the appropriate byte contains:
  -INT( <starting address> / 24 ) - 1
  where <starting address> is the starting
  position of the data for this day in the stack
- time of daily observation                  560. word
  first byte:  hour
  second byte: minute
- control words                       561 to 564. words
  for CMP and DCP subprograms
```

b.) Record of compacted precipitations and quality codes
/ S T A C K / (000 .LT. <stack No.> .LE. 366)

```
- length of compacted                          1. word
  precipitation block in words
- length of compacted                          2. word
  quality code block in words
- compacted precipitations and        3 to 240.words
  quality codes
```

HDB/WRK work file used for transiting the input data

```
          Type of the file:                   direct access
          Physical record length:             256 bytes
          I/O manipulation:                   FORTRAN I/O

          NOTE: this file must be reset to zero before its first use
                The number of records needed = 300
```

Data are compacted on the Data Base due to the limited capacity of the available data storages. The operation of compression and its invers are made by the help of special subprograms (for more information see MSCLIB/TXT). The following sample program shows the method of using these subprograms:

SAMPLE PROGRAM:

```
  1.        BYTE BQ, JBQ
  2.        DIMENSION ICB(1Ø), KEYS(15), IB(564),
  3.       /          IBI(31,12), BQ(31,12), JBI(96), JBQ(96)
  4.        EQUIVALENCE (IBI(1,1),IB(1)), (BQ(1,1),IB(374))
              .
              .
              .
  5.        CALL TRNRDR(ICB,KEYS,IB)
  6.        IF(ICB(5).EQ.Ø) GO TO 1
  7.        CALL TRNERR(ICB,KEYS)
  8.        STOP
  9.      1 CALL DCP(ICB.KEYS,IB,Ø,Ø,N,JBI,JBQ,Ø)
              .
              .
              .
 1Ø.        DO 2Ø MONTH=1,12
 11.        DO 1Ø NDAY=1,31
 12.        IF(BQ(NDAY,MONTH).LT.Ø) GO TO 5
              .
              .
              .
 13.        GO TO 1Ø
 14.      5 CALL DCP(ICB,KEYS,IB,MONTH,NDAY,N,JBI,JBQ,Ø)
              .
              .
              .
 15.     1Ø CONTINUE
 16.     2Ø CONTINUE
              .
              .
              .
```

Line 5. reads the record of daily data of the year given in KEYS array. ICB and KEYS arrays are set in the program part between Line 4. to Line 5.

Line 6. transfers the control to the label 1 in case of correct reading. Otherwise the program stops via Line 7. to Line 8.

Line 9. restores daily data from IB array (compacted input) and outputs them in IB. Since the fourth and fifth parameters are 0 the arrays ICB, KEYS, JBI and JBQ are not used. N will contain the time of observation valid for the whole year. The last parameter indicates the type of compression of IB array in input.

Line 12. transfers the control to the label 5 in case there is more than one data for the day NDAY of the month MONTH. Otherwise IBI(NDAY,MONTH) contains the precipitation and BQ(NDAY,MONTH) the quality code of the given day.

Line 14. restores all data of the day NDAY of the month MONTH from the appropriate STACK to JBI and JBQ arrays. N will contain the number of data in JBI and JBQ (N = 24 or 48 or 96).

6./ Supplementary information

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 1981
(or equivalent)

This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ Purpose of the program HDBCPY

This program transfers data records from the Current data file to the Archive file, or, if necessary (data correction needed on the Archive file), from Archive to Current file. In addition, HDBCPY can execute transfer between Current to Current and Archive to Archive file too. The program performs its task in dialogue mode. The possible User's answers are listed after the question in paranthesis, each of them divided by slashes. Default answer is always shown in brackets. In case of entering more than one numerical information in a line, an arbitrary number of spaces should be left between them (FREE FORMAT). The User's answers are controlled by the program, and in case of error, an error message will be screened before repeating the question.

The Current data file has to contain station-year record(s) of the current year. The record(s) are not in compacted form on this file. When the data processing of the current year is terminated (all data of the year are input on the file), the record(s) have to be transferred onto the Archive data file. On this file the station-year record(s) are in compacted form. A sample program is shown in STAGES/TXT for read data either from the Current or the Archive file.

After starting, the program asks for the transfer conditions and executes the transfer. Once the transfer is terminated, the program asks for new transfer phase. If no new transfer phase is required, the program stops.

2./ Available command NONE

3./ Used libraries

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the program

```
L80 HDBCPY-N,HDBCPY,MSCLIB-S,TRNLIB-S-E
```

5./ Used files in execution of the program

The program works with two files (source and destination), wich have to have four subfields in their data identifier. These subfields are the followings, in order:

- station code	from 1 to 6 digits
- stack number	3 digits
- year	4 digits
- type of data	2 digits

The code of data type is fixed for each type of files, as follow:

10	is used for	Current Water Level File	(HDB/SGW)
11		Archive Water Level File	(HDB/STG)
20		Current Precipitation File	(HDB/PCW)
21		Archive Precipitation File	(HDB/PRC)

6./ Supplementary information

Computer to be used for run: TRS-80 MODEL II (or equivalent)
Operating system to be used: TRSDOS version 2.0a January 1, 198
(or equivalent)

This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
the 10th of September 1983

1./ Purpose of the LIST programs

These programs edit yearly or daily tables of water level, discharge or precipitation data. The edition table can be output either on the display or on the line printer according to the User's request. The listing programs can recognise the two types of data file (Current and Archive - for more information see HDBCPY/TXT)

The listing program package contains the following programs:

LSTSTG edites yearly or daily tables of water levels according to the User's request (D or A subcommands).

LSTDCD edites yearly tables of discharges.

LSTDCA edites daily tables of discharges.

LSTPRC edites yearly or daily tables of precipitation data according to the User's request (D or A subcommand).

The programs LSTSTG, LSTDCD and LSTDCA use uniform signs, as follows:

blank for the inexistant days (for example: 31 February).

--- for the MISSING data (internal code: -999).

... for the day or time step, where no data has been entered yet (internal code: -998).

The program LSTPRC uses some more sign according to the special requirements of the data:

blank for the inexistant days and for NO RAIN on the day

0.0 for RAIN IN TRACE on the day (internal code: -100).

2./ Available command NONE

3./ Used libraries

LSTLIB For structuring the programs different subprograms have been written for the different parts of each program. These subprograms can not be used in any other program.

MSCLIB See description in MSCLIB/TXT

TRNLIB See description in TRNLIB/TXT

4./ Instruction for linkage of the programs

For the program LSTSTG:

```
L80 LSTSTG-N,LSTSTG,LSTLIB-S,MSCLIB-S,TRNLIB-S-E
```

For the program LSTDCD:

```
L80 LSTDCD-N,LSTDCD,LSTLIB-S,MSCLIB-S,TRNLIB-S-E
```

For the program LSTDCA:

```
L80 LSTDCA-N,LSTDCA,LSTLIB-S,MSCLIB-S,TRNLIB-S-E
```

For the program LSTPRC:

```
L80 LSTPRC-N,LSTPRC,LSTLIB-S,MSCLIB-S,TRNLIB-S-E
```

5./ Used files in execution of the programs

HDB/STN Station file

See description in STEDIT/TXT

This file is used by each listing program.

HDB/SGW Current data file containing water levels

See description in STAGES/TXT

This file can be used by the following programs:

LSTSTG, LSTDCD and LSTDCA

HDB/STG Archive data file containing water levels

For more information about Archive files see HDBCPY/TXT

This file can be used by the following programs:

LSTSTG, LSTDCD and LSTDCA

HDB/RTC Rating Curve file

See description in RCEDIT/TXT

This file is to be used by the following programs:

LSTDCD and LSTDCA

HDB/PCW Current data file containing precipitations

See description in PRECIP/TXT

This file can be used by the following program:

LSTPRC

HDB/PRC Archive data file containing precipitations

For more information about Archive files see HDBCPY/TXT

This file can be used by the following program:

LSTPRC

6./ Supplementary information

```
Computer to be used for run:    TRS-80 MODEL II ( or equivalent )
Operating system to be used:    TRSDOS version 2.0a January 1, 1981
                                ( or equivalent )
This program has been elaborated by VITUKI/BUDAPEST-Hungary (c)
                                 the 10th of September 1983
```

APPENDIX- B

DESCRIPTION OF THE APPLICATION PROGRAM AND SAMPLE RESULT

PROGARM USERS

1./ Purpose of the program

Analysis program (U) can process some desired data for a particular station (Obs. well) and for a particular year. The program can meet the following needs of the users:

1. Listing of basic characteristics of a station (Obs. well).
2. Listing of groundwater level data (raw data).
3. Listing of groundwater level data (missing data estimated).
4. Computation of some statistical parameters.
5. Determination of significant harmonics of data.

The program first asks the user to enter desired station number and year of analysis. It then retrieves basic characteristic of the station from HDB/SIN and water level data from HDB/SGW. If missing data is present, the program sends the raw data in subroutine MISS to estimate missing data. When estimation is completed, the program then save the refined data in HBD/DAI file and show a screen displaying the Usres Main Menu, whence following the USER'S GUIDE user can have his needs.

2./ Available commands in Main Menu

1 Listing of station characteristics on screen and/or Printer.
2 Listing of raw data on screen and/or printer.
3 Listing of refined data on screen and/or printer.
4 Listing of desired statistical parameters.
5 Listing of significant harmonics

3./ Used Libraries

USRLIB - for processing desired processing of the data.
MSCLIB - for miscelleneous processings.
TRNLIB - for reading in data from Data Base files.

4./ Instruction for linkage of the program

```
L80 U-N,USER,USRLIB,MSCLIB-S,TRNLIB-S-E
```

5./ Used files in the execution of the program

HDB/SIN : Data Base file containing station characteristics
HDB/SGW : Data Base file containing water level data of the station.
HDB/DAT : Data file where refined data is saved.

6./ Parameters

FL1 Array containing station file name
FL2 Array containing water level file name
FL3 Array containing Data file name where refined data is saved
KST Array containing digits of station number
NYEAR Year of analysis
NST Station number
NREC Record number (of HDB/DAT) where data is saved

7./ COMMON field(s)

/RAWDAT/ RX Array containing raw data
/DATAIN/ See description in section "DATAIN"

8./ Called subprogram(s)

DATAIN
MISS1
LSTSIN
LIST
STAT
HARM
PERMIT

2.0/ DESCRIPTION OF SUBPROGRAMS OF URSLIB LIBRARY

2.1/ DATAIN (NYEAR, KST, RX)

Called subprogram(s)

Reading in station data (basic charateristics of station) and water leval data from Data Base files FL1, FL2 (HDB/STN , HDB/SGW)

Parameters

NYEAR Year of analysis
KST Array containing digits of station number
RX Array containing raw data
ICB Control block for Water level file

JCB Control block for Station file
KEYS Array containing key elements
IBUF Array containing water level data (including quality code data)
IB Array containingwater level data excluding quality code of data)
NDAYS Array containing nos. of days in month of a given year

COMMAND fields

/DATACM/ X Array containing water level data

IB Array containing 64 words of station data (basic characteristics of the station)

words	
1 to 10 :	Obs. Well No.
11 to 20 :	Location of the well
21 to 25 :	Old no, if any
26 to 30 :	Type of Well
31 :	Availabilty of records
32 :	Coordinate latitude (deg) INT
33 TO 34 :	(min),REAL
35 :	Coordinate longitude (deg),INT
36 to 37 :	(min),REAL
38 to 39 :	Elevation of MP above MSL
40 to 41 :	Height of MP from GL
42 to 43 :	Depth of Obs. well below GL
44 to 64 :	Not used

JST	Array containing digits of station number
DAY	Array containing date (DAY) of each data
MONTH	Array containing date(Month) of each data
YEAR	Year of analysis
NTOTAL	Total number of data in the year (52 to 53)
NMS	Nos. of missing data

2.2/ LSTSTN

Purpose of the program

Listing of basic station characteristics on screen/printer

Parameters

LO	Logical unit number for writing
COLAMI	Fraction of the coordinate latitude of the station (minute)
COLOMI	Fraction on the coordinate latitude of the station (minute)
ELMPSL	Elevation of MP above MSL
ELMPGL	Height of MP from ground surface
DEPTH	Depth of Obs. well below ground surface

COMMON field(s)

/DATACM/ See description in section "DATAIN"

Called subprogram(s)

PERMIT

2.3/ LIST (INDEX)

Purpose of the subprogram

Listing water level data (raw data and refined data) on screen/printer.

Parameters

INDEX An integer whose assigned value determines whether

raw data or refined data is to be listed.
INDEX = 0 for listing of raw data
INDEX = 1 for listing of refined data

LO Logical unit number for writing

COMMON field(s)

/RAWDAT/ RX Array containing the raw data
/DATACM/ P Array containing the refined data
See description in section "DATAIN"

Called subprogram(s)

PERMIT

2.4/ MISS1

Purpose of the program

Estimation of missing data either by Spline-fit interpolation
or by weighted-average method.

Parameters

NB Nos of data taken before and after missing data
to pass Spline-function.

NESC See description in section "NMISS"

XX Array containing independent variables through which
the Spline-function is passed.

YY Array containing dependent variable through which
the Spline funtion is passed.

COMMON field(s)

/DATACM/ See description in section "DATAIN"

Called subprogram(s)

NMISS
MISS2
SPLIN

2.5/ SPLIN (X,Y,M,XINT,YINT)

Purpose of the subprogram

Passing a Spline funtion through a set of points (X,Y)
to estimate an intermediate point (XINT, YINT)

Parameters

X Array containing the independent variable
Y Array containing the dependent variable
M Nos of points
XINT An intermediate abscissa in X array
YINT An intermediate ordinate in Y array
C Array containing coefficients of spline function.

COMMON field(s) NONE

Called subprogram(s)

SPLICO

2.6/ SPLICO (X,Y,M.C)

Purpose of the subprogram
Computation of coefficients (C - array) of spline function.

Parameters
X,Y,M,C — See description in section "SPLIN"
D,P,E,A,B,Z — Auxillary array required to compute C array

COMMON field(s) NONE

Called subprogarm(s) NONE

NMISS (NESC)

Purpose of the subprogram
Determines whether missing data can be estimated by Spline-fit interpolation.

Parameters
NESC — An integer whose value determined by the subprogram indicate whether missing data can be estimated by Spline-fit interpolation.
MS — Array containing positions in the X array where data is missing
NT — Total number of missing data

COMMON field(s)
/DATACM/ See description in section "DATAIN"

Called subprogram(s) NONE

2.8/ MISS2

Purpose of the program
Estimation of missing data by weighted-average method

Parameters
ST — One dimensional array containing station numbers whose missing data can be estimated by this subprogram.
RST — Two-dimensional array containing neibouring stations [related each of the ST station for estimating missing data]
SIDE — Two-dimensional array containing distences between ST station and related RST station and also distences between consecutive RST stations.
D — array containing area elements formed by SIDES
FACTOR — Array containing weighted factor of RST stations.
XH — Temporary one-dimensional array containing data of a RST station.
XS — Two-dimensional array containing all data of RST

stations for a particular ST station.

NSIDE Total number of RST station for a particular ST station.

NST Station number

COMMON field(s)

/DATACM/ See description in section "DATAIN"

Called subprogram(s)

DATAIN
PERMIT
SPLSTN
AREA

2.9/ SPLSTN (NST , IST)

Purpose of the program

Splitting a station number into three digits contained in an array

Parameters

NST Station number say 24

IST Array containing digits of station number [0,2,4 for station 24]

COMMON field(s) NONE

Called subprogram(s) NONE

2.10/ AREA (A,B,C,D,K)

Purpose:of ths subprogram

Computation of areas of a traingle from three sides and to split this area into two parts which the line drawing from the vertex (ST station) perpendicular to the opposite side (joining two consecutive RST station) divide this area . The divided areas in contained in D array.

Parameters

A,B,C Three sides of a traingle.

D Area containing elements of areas by this subprogram. It is obvious that total number of elements in D array is 6 x 2 = 12 .

K Position in the RST station for which elements of D array is computed.

COMMON field(s) NONE

Called subprograms NONE

2.11/ PERMIT

Purpose of the subprogram

Temporary of cessation of execution of the program.

The subprogram writes on the screen the message ' Press ENTER to continue ' and at the same time asks to enter data for an integer NCONT. pressing ENTER assigns a value zero to NCONT and the execution of the program resumes.

COMMON field(s)   NONE

Called subprogram(s)   NONE

2.12/ STAT

Purpose of the subprogram

Computation of some basic statistical parameters.

Parameters

NCHOIC	Integer number indicating choice of the parameter
XMAX	Maximum value in the X array.
XMIN	Mininmum value in the X array.
RANGE	Range of fluctuation (XMAX - XMIN)
XMINGL	Depth of lowest level from ground surface
SD	Standard deviation of the data
VAR	Variance of the data

COMMON field(s)

/DATACM/   See description in section   "DATAIN"

Called subprogram(s)

PERMIT

2.13/ ADDRES (NST, NYEAR, NREC)

Purpose of the subprogram

Compute record number (HDB/DAT file) in which refined data is saved

Parameters

NST	Station number
NYEAR	Year of analysis
NREC	Record number

COMMON field(s)   NONE

Called subprogram(s)   NONE

2.14/ HARM

Purpose of the subprogram

This program was originally intended for harmonic analysis of the data. Due for memory deficit the subprogarm has been isolated as a separate progarm and is named as HARM. The progarm HARM and its subroutines are described in the next article. Ihe purpose of this subroutine is just to inform the user what action to take for harmonic analysis of the data.

Parameters NONE

COMMON field(s) NONE

Called subprogarm(s) NONE

3.1/ HARM

Purpose of the program

Determination of significant harmonics present in the periodic component of the data.

Parameter(s)

X Array containing the groundwater level data.
AMP Array containing amplitudes of the harmonics
PHA Array containing the phases of the harmonics
A,B Auxillary array required to compute AMP and PHA array
H Periodic component computed taking a number of harmonics (1-26) taken as significant.
XXM Auxillary array used in computing H array
RES Array containing sum of residuals (stochastic component) for all possible number of significat harmonics (1,2,3,.....,26).
P,DP,DDP Auxillary arrays used to determine significant harmonics by graphical mathod.
SX Sum of X array
SXX Sum of squares of X array.
SD Standard deviation of X array
VAR Variance of X array

COMMON field(s) NONE

Called subprogarm(s)

ANALYS
ADDRES

3.2/ ANALYS (N,X)

Purpose of the subprogram

Analysis of the stochastic component computing standard deviation, variance, skewness coefficient, excess coefficient.

Parameters

X	array containing the stochastic component
N	Nos of data in the array
SX	Sum of the array
SXX	Sum of squares of the array
SD	Standard deviation of the array
XMEAN	Mean the array
CS	Skewness coefficient of the array
CE	Excess coefficient of the array

COMMON field(s) NONE

Called subprogram(s) NONE

3.3/ ADDRES (NST, NYEAR, NREC)

Purpose of the subprogram

Find the address of the record (record no.) in HDB/DAT where refined groundwater level data is saved. This subprogarm is identical to that described in article 2.13.

SAMPLE RESULTS

Basic characteristics of station

OBSERVATION WELL NO : MYMENSINGH-73 /(M-073)
Old No (if any) : MY-98
Location (Village) : PANGARA
Type of Well : PIEZOMETER
Depth of Well : 39.014 metre
Elevation of MP : 14.572 metre above MSL
Height of MP : 1.168 metre above GL
Latitude : 24 deg 32.5 min / Longitude : 90 deg 23.5 min
Availability of records (since) : 1978

SL NO.	DATE OF OBSERVATION	WATER LEVEL FROM MSL(m)	SL NO.	DATE OF OBSERVATION	'WATER LEVEL 'FROM MSL(m)
1.	7/ 1/1980	9.342	27.	7/ 7/1980	9.262
2.	14/ 1/1980	9.212	28.	14/ 7/1980	9.392
3.	21/ 1/1980	9.082	29.	21/ 7/1980	9.492
4.	28/ 1/1980	8.882	30.	28/ 7/1980	9.622
5.	4/ 2/1980	8.732	31.	4/ 8/1980	9.752
6.	11/ 2/1980	8.502	32.	11/ 8/1980	9.872
7.	18/ 2/1980	8.322	33.	18/ 8/1980	10.022
8.	25/ 2/1980	8.042	34.	25/ 8/1980	10.232
9.	3/ 3/1980	7.842	35.	1/ 9/1980	10.282
10.	10/ 3/1980	7.862	36.	8/ 9/1980	10.532
11.	17/ 3/1980	7.382	37.	15/ 9/1980	10.532
12.	24/ 3/1980	7.232	38.	22/ 9/1980	10.482
13.	31/ 3/1980	7.052	39.	29/ 9/1980	10.462
14.	7/ 4/1980	6.852	40.	6/10/1980	10.402
15.	14/ 4/1980	6.702	41.	13/10/1980	10.402
16.	21/ 4/1980	6.652	42.	20/10/1980	10.382
17.	28/ 4/1980	6.622	43.	27/10/1980	10.382
18.	5/ 5/1980	7.252	44.	3/11/1980	10.332
19.	12/ 5/1980	7.412	45.	10/11/1980	10.202
20.	19/ 5/1980	7.562	46.	17/11/1980	10.052
21.	26/ 5/1980	7.862	47.	24/11/1980	9.902
22.	2/ 6/1980	8.142	48.	1/12/1980	9.852
23.	9/ 6/1980	8.352	49.	8/12/1980	9.752
24.	16/ 6/1980	8.532	50.	15/12/1980	9.672
25.	23/ 6/1980	8.732	51.	22/12/1980	9.312
26.	30/ 6/1980	9.012	52.	29/12/1980	9.212

```
Highest Level =  10.532 metre above MSL
Lowest  Level =   6.622 metre above MSL
Depth of lowest level from GL =   6.782 metre
Range of fluctuation =   3.910 metre
Mean Level =   8.980 metre
Standard deviation =   1.206 metre
Variance =    1.455 sq. metre
```

```
Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M -73
Year : 1980



Number of significant harmonics =  1



Harmonics      Amplitude       Phase


    1           1.658         79.801
    2            .325        313.192
    3            .186         47.695
    4            .070        288.728
    5            .022        303.145
    6            .067         82.962
```

Basic characteristics of station

OBSERVATION WELL NO : JAMALPUR - 47 /(M-047)
Old No (if any) : MY-61
Location (Village) : SOUTHATKAPARA
Type of Well : DUG
Depth of Well : 17.831 metre
Elevation of MP : 9.098 metre above MSL
Height of MP : .457 metre above GL
Latitude : 25 deg 0.0 min / Longitude : 90 deg 50.8 min
Availability of records (since) : 1977

SL NO.	DATE OF OBSERVATION	WATER LEVEL FROM MSL(m)	SL NO.	DATE OF OBSERVATION	'WATER LEVEL 'FROM MSL(m)
1.	1/ 1/1979	5.878	27.	2/ 7/1979	7.728
2.	8/ 1/1979	5.768	28.	9/ 7/1979	8.008
3.	15/ 1/1979	5.648	29.	16/ 7/1979	8.128
4.	22/ 1/1979	5.568	30.	23/ 7/1979	7.678
5.	29/ 1/1979	5.538	31.	30/ 7/1979	8.998
6.	5/ 2/1979	5.488	32.	6/ 8/1979	8.388
7.	12/ 2/1979	5.388	33.	13/ 8/1979	7.958
8.	19/ 2/1979	5.338	34.	20/ 8/1979	8.008
9.	26/ 2/1979	5.208	35.	27/ 8/1979	8.158
10.	5/ 3/1979	5.088	36.	3/ 9/1979	8.258
11.	12/ 3/1979	4.988	37.	10/ 9/1979	8.388
12.	19/ 3/1979	4.958	38.	17/ 9/1979	8.238
13.	26/ 3/1979	4.958	39.	24/ 9/1979	7.828
14.	2/ 4/1979	4.908	40.	1/10/1979	7.678
15.	9/ 4/1979	4.878	41.	8/10/1979	7.958
16.	16/ 4/1979	4.858	42.	15/10/1979	7.628
17.	23/ 4/1979	4.878	43.	22/10/1979	7.648
18.	30/ 4/1979	4.878	44.	29/10/1979	7.068
19.	7/ 5/1979	4.958	45.	5/11/1979	6.708
20.	14/ 5/1979	5.898	46.	12/11/1979	6.688
21.	21/ 5/1979	5.918	47.	19/11/1979	6.478
22.	28/ 5/1979	5.878	48.	26/11/1979	6.428
23.	4/ 6/1979	5.748	49.	3/12/1979	6.788
24.	11/ 6/1979	6.478	50.	10/12/1979	6.608
25.	18/ 6/1979	7.018	51.	17/12/1979	6.528
26.	25/ 6/1979	7.398	52.	24/12/1979	6.428
			53.	31/12/1979	6.298

Basic characteristics of station

OBSERVATION WELL NO	: JAMALPUR - 47 /(M-047)
Old No (if any)	: MY-61
Location (Village)	: SOUTHATKAPARA
Type of Well	: DUG
Depth of Well	: 17.831 metre
Elevation of MP	: 9.098 metre above MSL
Height of MP	: .457 metre above GL

Latitude : 25 deg 0.0 min / Longitude : 90 deg 50.8 min
Availability of records (since) : 1977

SL NO.	DATE OF OBSERVATION	WATER LEVEL FROM MSL(m)	SL NO.	DATE OF OBSERVATION	'WATER LEVEL 'FROM MSL(m)
1.	1/ 1/1979	5.878	27.	2/ 7/1979	7.728
2.	8/ 1/1979	5.768	28.	9/ 7/1979	8.008
3.	15/ 1/1979	5.648	29.	16/ 7/1979	8.128
4.	22/ 1/1979	5.568	30.	23/ 7/1979	7.678
5.	29/ 1/1979	5.538	31.	30/ 7/1979	8.998
6.	5/ 2/1979	5.488	32.	6/ 8/1979	8.388
7.	12/ 2/1979	5.388	33.	13/ 8/1979	7.958
8.	19/ 2/1979	5.338	34.	20/ 8/1979	8.008
9.	26/ 2/1979	5.208	35.	27/ 8/1979	8.158
10.	5/ 3/1979	-----	36.	3/ 9/1979	8.258
11.	12/ 3/1979	-----	37.	10/ 9/1979	8.388
12.	19/ 3/1979	4.958	38.	17/ 9/1979	8.238
13.	26/ 3/1979	4.958	39.	24/ 9/1979	7.828
14.	2/ 4/1979	4.908	40.	1/10/1979	7.678
15.	9/ 4/1979	4.878	41.	8/10/1979	7.958
16.	16/ 4/1979	4.858	42.	15/10/1979	7.628
17.	23/ 4/1979	4.878	43.	22/10/1979	7.648
18.	30/ 4/1979	4.878	44.	29/10/1979	7.068
19.	7/ 5/1979	4.958	45.	5/11/1979	6.708
20.	14/ 5/1979	5.898	46.	12/11/1979	6.688
21.	21/ 5/1979	5.918	47.	19/11/1979	6.478
22.	28/ 5/1979	5.878	48.	26/11/1979	6.428
23.	4/ 6/1979	5.748	49.	3/12/1979	6.788
24.	11/ 6/1979	6.478	50.	10/12/1979	6.608
25.	18/ 6/1979	7.018	51.	17/12/1979	6.528
26.	25/ 6/1979	7.398	52.	24/12/1979	6.428
			53.	31/12/1979	6.298

Missing data estimated by spline-fit interpotation

Basic characteristics of station

OBSERVATION WELL NO : JAMALPUR - 47 /(M-047)
Old No (if any) : MY-61
Location (Village) : SOUTHATKAPARA
Type of Well : DUG
Depth of Well : 17.831 metre
Elevation of MP : 9.098 metre above MSL
Height of MP : .457 metre above GL
Latitude : 25 deg 0.0 min / Longitude : 90 deg 50.8 min
Availability of records (since) : 1977

SL NO.	DATE OF OBSERVATION	WATER LEVEL FROM MSL(m)	SL NO.	DATE OF OBSERVATION	WATER LEVEL FROM MSL(m)
1.	1/ 1/1979	5.878	27.	2/ 7/1979	7.728
2.	8/ 1/1979	5.768	28.	9/ 7/1979	8.008
3.	15/ 1/1979	5.648	29.	16/ 7/1979	8.128
4.	22/ 1/1979	5.568	30.	23/ 7/1979	7.678
5.	29/ 1/1979	5.538	31.	30/ 7/1979	8.998
6.	5/ 2/1979	5.488	32.	6/ 8/1979	8.388
7.	12/ 2/1979	5.388	33.	13/ 8/1979	7.958
8.	19/ 2/1979	5.338	34.	20/ 8/1979	8.008
9.	26/ 2/1979	5.208	35.	27/ 8/1979	8.158
10.	5/ 3/1979	5.125	36.	3/ 9/1979	8.258
11.	12/ 3/1979	5.041	37.	10/ 9/1979	8.398
12.	19/ 3/1979	4.958	38.	17/ 9/1979	8.238
13.	26/ 3/1979	4.958	39.	24/ 9/1979	7.828
14.	2/ 4/1979	4.908	40.	1/10/1979	7.678
15.	9/ 4/1979	4.878	41.	8/10/1979	7.958
16.	16/ 4/1979	4.858	42.	15/10/1979	7.628
17.	23/ 4/1979	4.878	43.	22/10/1979	7.648
18.	30/ 4/1979	4.878	44.	29/10/1979	7.068
19.	7/ 5/1979	4.958	45.	5/11/1979	6.708
20.	14/ 5/1979	5.898	46.	12/11/1979	6.688
21.	21/ 5/1979	5.918	47.	19/11/1979	6.478
22.	28/ 5/1979	5.878	48.	26/11/1979	6.428
23.	4/ 6/1979	5.748	49.	3/12/1979	6.788
24.	11/ 6/1979	6.478	50.	10/12/1979	6.608
25.	18/ 6/1979	7.018	51.	17/12/1979	6.528
26.	25/ 6/1979	7.398	52.	24/12/1979	6.428
			53.	31/12/1979	6.298

Missing data Estimated by weighted-average method

Basic characteristics of station

OBSERVATION WELL NO : JAMALPUR - 2 /(M-002)
Old No (if any) : MY-04
Location (Village) : BARAHALIA
Type of Well : DUG
Depth of Well : 9.271 metre
Elevation of MP : 10.160 metre above MSL
Height of MP : .914 metre above GL
Latitude : 24 deg 53.8 min / Longitude : 90 deg 53.7 min
Availability of records (since) : 1967

SL NO.	DATE OF OBSERVATION	WATER LEVEL FROM MSL(m)	SL. NO.	DATE OF OBSERVATION	'WATER LEVEL 'FROM MSL(m)
1.	1/ 1/1979	5.878	27.	2/ 7/1979	7.728
2.	8/ 1/1979	5.768	28.	9/ 7/1979	8.008
3.	15/ 1/1979	5.648	29.	16/ 7/1979	8.128
4.	22/ 1/1979	5.568	30.	23/ 7/1979	7.678
5.	29/ 1/1979	5.538	31.	30/ 7/1979	8.998
6.	5/ 2/1979	5.488	32.	6/ 8/1979	8.388
7.	12/ 2/1979	5.388	33.	13/ 8/1979	7.958
8.	19/ 2/1979	5.338	34.	20/ 8/1979	8.008
9.	26/ 2/1979	5.208	35.	27/ 8/1979	8.158
10.	5/ 3/1979	5.243	36.	3/ 9/1979	8.258
11.	12/ 3/1979	5.186	37.	10/ 9/1979	8.388
12.	19/ 3/1979	5.196	38.	17/ 9/1979	8.238
13.	26/ 3/1979	4.958	39.	24/ 9/1979	7.828
14.	2/ 4/1979	4.908	40.	1/10/1979	7.678
15.	9/ 4/1979	4.878	41.	8/10/1979	7.958
16.	16/ 4/1979	4.858	42.	15/10/1979	7.628
17.	23/ 4/1979	4.878	43.	22/10/1979	7.648
18.	30/ 4/1979	4.878	44.	29/10/1979	7.068
19.	7/ 5/1979	4.958	45.	5/11/1979	6.708
20.	14/ 5/1979	5.898	46.	12/11/1979	6.688
21.	21/ 5/1979	5.918	47.	19/11/1979	6.478
22.	28/ 5/1979	5.878	48.	26/11/1979	6.428
23.	4/ 6/1979	5.748	49.	3/12/1979	6.788
24.	11/ 6/1979	6.478	50.	10/12/1979	6.608
25.	18/ 6/1979	7.018	51.	17/12/1979	6.528
26.	25/ 6/1979	7.398	52.	24/12/1979	6.428
			53.	31/12/1979	6.298

```
Highest Level =   8.998 metre above MSL
Lowest  Level =   4.858 metre above MSL
Depth of lowest level from GL =   3.783 metre
Range of fluctuation =   4.140 metre
Mean Level =   6.532 metre
Standard deviation =   1.219 metre
Variance =   1.487 sq. metre
```

```
Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M -47
Year : 1979


 Number of significant harmonics =  1


Harmonics        Amplitude       Phase


    1            1.632          291.269
    2             .440          298.296
    3             .079          321.615
    4             .114            1.282
    5             .126           75.176
    6             .020          283.502
```

```
Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M - 2
Year : 1979



Number of significant harmonics =  1



Harmonics      Amplitude      Phase


    1           2.126        291.088
    2            .588        273.095
    3            .299        324.344
    4            .319        326.630
    5            .169        309.497
    6            .145        358.841



Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M - 2
Year : 1980



Number of significant harmonics =  1



Harmonics      Amplitude      Phase


    1           1.886        310.828
    2            .275        354.876
    3            .286         37.606
    4            .084         78.417
    5            .085         76.162
    6            .032        292.200
```

```
Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M - 2
Year : 1981



Number of significant harmonics =  1



Harmonics      Amplitude      Phase


    1           1.793        329.354
    2            .254         27.431
    3            .328        299.382
    4            .204        357.973
    5            .038         14.324
    6            .033         37.445
```

```
Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M -25
Year : 1979



Number of significant harmonics =  2



Harmonics      Amplitude      Phase


    1           1.681        284.990
    2            .345         85.324
    3            .109        331.464
    4            .123        303.983
    5            .115         46.342
    6            .060         68.640
```

```
Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M -25
Year : 1980



Number of significant harmonics =  1



Harmonics      Amplitude      Phase


    1           1.584        298.935
    2            .268         27.503
    3            .277         48.203
    4            .093        286.285
    5            .145        331.628
    6            .094        336.023



Harmonic analysis of groundwater level data
OBSERVATION WELL NO : M -25
Year : 1981



Number of significant harmonics =  2



Harmonics      Amplitude      Phase


    1           1.068        353.610
    2            .278         13.317
    3            .146        353.374
    4            .026          2.757
    5            .113         85.173
    6            .076        291.977
```

APPENDIX - C

USER'S GUIDE

APPENDIX - D

USER'S GUIDE

INTRODUCTION :

This Guide is intended for the users of present Groundwater level Data Processing System which can be implemented in a TRS-80 Model II or an enhanced Model such as TRS-80 Model 16B. An effectual use of this Manual need that users have a smattering of the operating system and are conversant with the terms : SYSTEM, TRSDOS, BOOTING A DISKETTE, LOADING THE OPERATING SYSTEM, EXECUTING OR RUNNING A PROGRAM etc. Beginners are advised to scan the first two chapters (INTRODUCTION, COMMAND & UTILITIES) of the OWNER'S MANUAL of the TRS-80 Microcomputer.

CLASSES OF USERS

Users of this DATA PROCESSING SYSTEM can be looked upon into three categories:

CLASS A : Those who will retrieve and analyse data already stored in Data Base files.

CLASS B : Those who will enter data onto Data Base files and then use use the data like class A users.

CLASS C : Those who will first create the Data Base files and then insert and retrieve data like class B users.

HOW TO USE THE GUIDE :

This guide is divided into four sections :

Section I decribes how to boot the program diskette and start up with the system as well as what precautions to adopt during running the programs. This section is for all classes of users.

Section IV illustrates use of data after retrieval from the Data Base Here the steps involved in running the application program developed in this study are described. So this section is a must for the beginner of a class A user. Class A and Class B users may also need to consult this section if they wish to retrieve and analyse data from Data Base.

Section II describes Data Base files and their creation and is for class C users only

Section III describes steps for entering data onto Data Base files and is for class B and C users.

SECTION I

1.1/ LOADING THE PROGRAM DISKETTE

Load the program diskette by following the instructions below :

1. Turn on the TRS-80 Model 16B Computer.
2. Turn all peripheriels (including Disk Expansion Unit (if any),
3. printers etc.) on.
3. Insert the program diskette in drive zero [the left disk drive built into the Computer] . Be sure that the level on the diskette faces left.
4. Close the door of drive zero firmly.
5. The screen should show INITIALIZING or LOADING. If the screen shows BOOT ERROR 1K press RESET button located below the system power light and the screen will now show INITIALIZING OR LOADING. [if still a BOOT ERROR occurs, then probably the program diskette is not a system diskette, insert the proper diskstte]. The Computer will take a few seconds to load the operating system and then will display on the screen
   Enter Date (MM/DD/YYYY)...........
   Now the user is to enter today's date exactly in the format (Month//Day/Year). The year must be a four digit year. As for example if today is the first day of february, 1985 , then type
   02/01/1985
   and press ENTER. The screen will now show
   Enter Time (HH.MM.SS)........
   Suppose it is 10 minutes 5 seconds after 9 a.m. Then type
   09.10.05
   and press ENTER. The screen will now show
   TRSDOS READY

....

Look at the CAPS key. The red light should be on. If the light is off press the key once.

1.2/ CAUTION

Never leave the program diskette when work is done. It could be harmed by power failure or mechanical failure in the drive(s).
Never remove a program/data diskette(s) from the drive(s) when the program is running.
Keep copies of Data file(s) reserved. Because if a power failure occurs during execution of the program, open Data file(s) will not be closed and a next attempt to open it will destroy it.

SECTION II

2.1/ GENERAL INFORMATION ABOUT PROGRAM AND DATA FILES

The Data Processing System composed of some files which contains the body of the programs. They are called program files and are the followings

HDBMNT
STEDIT
STAGES
U
H

While a number of files, called Data files are used during execution of the programs. They are :

HDB/STN : Station file
HDB/SGW : Water level file
HDB/WRK : Work file

HDB/STN contains all the basic characteristics of stations (Obs. well). Actual data (water level data) is stored in HDB/SGW. Before entering actual data of a station in HDB/SGW, basic characteristics of the station must be entetred first in HDB/SIN. HDB/WRK is a temporary file used during execution entering onto HDB/SGW. The input data first enters a memory buffer and in HDB/WRK and then onto HDB/SGW. STEDIT program is used to enter station data (basic characteristics of station) in HDB/STN and STAGES program is used to enter water level data in HDB/SGW. But before entering data in HDB/STN or HDB/SGW, they must be created and initialized by HDBMNT.

2.2/ CREATION OF DATA BASE FILES

2.2.1. Initializing HDB/STN

When TRSDOS READY is on the screen type HDBMNT and press ENTER. Follow the sequence of steps that are given below :

```
            THE SCREEN SHOWS                    TYPE AS GIVEN IN THIS COLUMN
                                                AND PRESS ENTER IN EACH STEP

1. HDBMNT-program ready
     *..............................                  I
2. INIT of TRN-files ready
   ? Wich file  FILENAME/EXT:d = ......               HDB/STN
3.   Opening file : HDB/STN
   ? No of sectors to be reset to zero =              20
4. ? LOWER limit of data identifier =                 001
5. ? UPPER limit of data identifier =                 999
6. ? Enter date(DD/MM/YYYY) =                   01/02/1985
7. ? Enter NAME of OWNER in 16 char =                 NIAZ
8.   File init done for HDB/STN
   *.......................                           Q
```

```
    B y e
    STOP    **at address 32E8**
    TRSDOS READY
```

REMARKS: Response given against steps 1, 2, 4, 5 and 8 are not changable for this Data Processing System. Nos. of sectors (Step 3) depends on the number of stations to be stored in HDB/STN file and is equal to half the number of stations plus 2.

2.2.2. Initializing HDB/SGW

When TRSDOS READY is on the screen, type HDBMNT and press ENTER. Follow the sequence of steps given below :

```
                                          TYPE AS GIVEN IN THIS COLUMN
          THE SCREEN SHOWS                  AND PRESS ENTER IN EACH STEP

1. HDBMNT-program ready
      *............................                  I
2. INIT of TRN-files ready
   ? Wich file  FILENAME/EXT:d =                     HDB/SGW
3.   Opening file : HDB/SGW
      ?  No of sectors to be reset to zero =         200
4. ? LOWER limit of data identifier =                001 000 1950 10
5. ? UPPER limit of data identifier =                999 366 2000 10
6. ? Enter date(DD/MM/YYYY) =                        01/02/1985
7. ? Enter NAME of OWNER in 16 char. =               NIAZ
8.   File INIT done for HDB/SGW
      *...................................           Q
   B y e
   STOP    **at address 32E8**
   TRSDOS READY
```

REMARKS : Response to be given against steps 1,2 and 8 are not changable for this Data Processing System. Nos. of records depends on data sets to be entered in HDB/SGW and is equal to 4 records per Year per Station. Limits of data identifier (steps 4, 5) consists of four numbers (Station limits, Not used, Year limits, Code of data type limits). Only the Year limits can be changed depending on the time of data analysis. The four numbers of data identifier limit is entered in free format. Leave arbitary number of space(s) (1 in the above example) beteween the numbers.

2.3/ Initializing HDB/WRK

HDB/WRK be initialized exactly as the same way as of HDB/SGW, except that in this case only first three numbers is to be entered for limits of data identifier. More clearly , type 001 000 1950 for lower limit and type 999 366 2000 for upper limits of data identifier.

2.4/ Initializing HDB/DAT

HDB/DAT is not in fact a Data Base file and as such it need not be created by HDBMNT. It is automatically created by Users program.

S E C T I O N III

3.1/ ENTERING STATION DATA IN HDB/SIN

When TRSDOS READY is on the screen type STEDIT and press ENTER. Enter data following the sequence of steps below :

```
                                                     TYPE AS GIVEN IN THIS COLUMN
           THE SCREEN SHOWS                          AND PRESS ENTER IN EACH STEP

 1. STEDIT-program ready
    ? Station file    FILENAME/EXT:d =                       HDB/SIN
 2.  Opening file : HDB/SIN
    *...............................                         I
 3. ? Station code =                                 002
 4. ? Observation well no =                          MYMENSINGH-02
 5. ? Location ( village ) =                         BARAHATTA
 6. ? Old station code =                                     MY-04
 7. ? Type of well =                                 DUG
 8. ? Availability of records ( since ) =                    1976
 9. ? Coordinate latitude (degrees) =                24
10. ?                (minutes)                       53.8
11. ? Coordinate longitude (degrees) =                       90
12. ?                (minutes) =                     53.7
13. ? Elevation of MP above MSL (m) =                10.160
14. ? Height of MP above ground surface =                    0.914
15. ? Depth of well =                                        9.271
16. *...............................                         E
    B y e
    STOP    **at address 3419**
      TRSDOS READY
```

REMARKS : All the data except those entered in step 1, 2, 16 are changable and depend on the station to be entered in Data Base.

3.2/ ENTERING WATER LEVEL DATA IN HDB/SGW

Make sure that basic characteristics of the station is entered in HDB/SIN. When TRSDOS READY is on the screen , type STAGES and press ENTER and follow the procedure below :

```
                                                TYPE AS GIVEN IN THIS COLUMN
          THE SCREEN SHOWS                      AND PRESS ENTER IN EACH STEP

1. S T A G E S  PROGRAM READY
   ? Station file :   FILENAME/EXT:d =                 HDB/SIN
2.   Opening file : HDB/SIN
   ? Waterlevel file: FILENAME/EXT:d =                 HDB/SGW
3.   Opening file : HDB/SGW
   ? Work file    :   FILENAME/EXT:d =                 HDB/WRK
4.   Opening file : HDB/WRK
   *                                            1
5. ? LIMITS in CM for checking stages
     min, max, -diff min, diff max =            Ø 999 -999 999
6. ? Starting date(DD/MM/YYYY) =                       Ø1/Ø1/1979
7. ? Time step code (1/7) =                     7
8. ? Station code =                             ØØ2
```

9. Now enter water level data and quality code, if any as per directions shown on the screen. When insetion of data is completed screen will show *, type E and press ENTER; Computer will save the data in disk (file HDB/SGW) and return to TRSDOS.

REMARKS : Responses given in steps 1, 2, 3 and 4 in the above example is not changable. Data entered in step 9 must be in integer and in CM.

S E C T I O N IV

4.1/ HOW TO RUN THE APPLICATION PROGRAM DEVELOPED IN THIS STUDY

When TRSDOS READY is on the screen type U and press ENTER ; The screen will show

```
WELCOME TO U S E R S-program

THIS PROGRAM CAN --------------------------------------------
                     (i) List station charateristics and data
                    (ii) Estimate missing data
                   (iii) Give statistical parameters of data
                    (iv) Give harmonics of data
                    --------------------------------------------
FOR ANY YEAR AND FOR ANY STATION ESTABLISHED IN DATA BASE
```

Press ENTER to Continue

Press ENTER and the screen will show

Station :

Type station (Obs. well No.) number in three numeric digits and press ENTER and the screen will show

Year :

Type year of analysis in four digits and press ENTER ;
Computer will now retrieve station data from HDB/SIN file and water level data from HDB/SGW file. If missing data is present in the data Computer will proceed to estimate it first by Spline-fit interpolation. If Spline-fit is aborted to estimate the missing data screen will show

Missing data estimation by Spline-fit is aborted
Do you wish to continue estimation by weighted average method (Y/N) :

Type Y and press ENTER . If missing data estimation by weighted-average method for the station under analysis is established in the program then Computer will proceed to retrieve data for the neibouring station and the estimate the missing data by weighed-average method. The screen will then show
[If missing data estimation is not possible, screen will show
! OUT OF RANGE
Missing data estimation is still aborted
Press ENTER to Continue

Press ENTER and the screen will show]

```
                    ENTER USERS SELECTION
                    =====================

            1. Listing of station characteristics
            2. Listing of raw data
            3. Listing of refined data ( missing data estimated )
            4. Harmonic analysis of data
            5. Exit from program
```

This is the Main Menu

Suppose you need to list the station characteristics then type 1 and press ENTER , the screen will show
Printing desired on Vedio Screen/ Printer (V/P):
Type P if printing is desired on Printer, otherwise type V and press ENTER . Computer will print you basic characteristics characteristicse of the station you are currently analysing and return to the screen

Press ENTER to continue.

Press ENTER and the computer will again return to the Main Menu

Now suppose you wish for some statistical parameter of the data In that case type 4 and press ENTER and the screen show a subMenu

```
                    DESIRED STATISTICS----------------

               1.  Highest level above MSL
               2.  Lowest  level above MSL
               3.  Depth of lowest level from GL
               4.  Range of fluctuation
               5.  Mean level
               6.  Standard deviation
               7.  Variance
               8.  All of above
               9.  Exit from subroutine
```

Suppose you need all items (1-7), type 8 and press ENTER . The screen will show

```
Printing desired on  PRINTER, CONSOL (P/C) :
```

Type P or C and the Computer will print you all desired parameters and return to the subMenu again. Type 9 to branch to the Main Menu again.

In this way you can have all your needs for the current station and for the year you have entered at the start of the program.

Now if you wish to terminate analysis for for the current station type 6 from the Main Menu. The screen show

```
Analysis for the station [    ] is completed
    Do you want a new phase (Y/N) :
```

Now if you wish to analysis a new station type Y and the Computer will ask for station and year of analysis. Typing N will close all the data files and return to TRSDOS.

4.27 HOW TO RETRIEVE DATA FROM DATA BASE FOR USER'S PROGRAM

Any user requiring data for his program should write a subroutine like DATAIN (NYEAR, KST, RX) presented in appendix - D. Input to this subroutine is NYEAR (Year of data) and KST (array containing 3 digits of station number). Output from this subroutine are RX (array containing raw data), IB (array containing station data), DAY & MONTH (arrays containing dates of data collection) The arrays IB , DAY, MONTH are contained in COMMON BLOCK. In most case users are likely to need only the raw data (array RX)

In such cases DATAIN subroutine can be directly used. If other data are required a COMMON BLOCK must be present in the user's program.

An example is given below to illustrate data retrieval from Data Base. The problem is to write a program to compute differences in groundwater levels between two station 002 and 024 in 1980 retrieving data from from Data Base.

```
C       PROGRAM ILLUSTRATING RETRIEVAL OF DATA FROM DATA BASE
        DIMENSION X(53), Y(53), DIFF(53) , RX(53), FL1(4), FL2(4)
        COMMON /DATACM/ Z(53), IB(64), KST(3), DAY(53), MONTH(53),
        NTOTAL, NMS

        DATA FL1/'HDB/','SIN ',2*'    '/, FL2/'HDB/','SGW ',2*'    '/
        CALL OPEN (6, FL1, 0)
        CALL OPEN (7, FL2, 0)
        CALL DATAIN (1980,0,0,2,Y)
        CALL DATAIN (1980,0,2,4,Y)
        DO 111 I = 1 TO NTOTAL
 111    DIFF(I) = Y(I) - X(I)
        STOP
        END
```

This program must be linked to the DBMS following instruction given in appendix - D.

4.3/ ENDING DATA PROCESSING

When work with the Computer is done

1. Remove the diskette(s) from the drive(s)
2. Turn all peripherials [including floopy disk expansion unit (if any), Printers etc.] OFF
3. Turn the Computer OFF

- Good Bye -

APPENDIX - D

LISTING OF THE APPLICATION PROGRAM

```
      PROGRAM USERS
      INTEGER DAY,YEAR
      BYTE CNTRL,NO,ANS
      DIMENSION  FL1(4),FL2(4), FL3(2), KST(3)
      COMMON /DATACM/ X(53),IB(64),JST(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL, NMS
      COMMON /RAWDAT/ RX(53)
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +        (ELMPSL,IB(38)), (ELMPGL,IB(40)), (DEPTH,IB(42))
      DATA CNTRL/30/,NO/'N'/
      DATA FL1/'HDB/','STN ',2*'    '/,FL2/'HDB/','SGW ',2*'    '/
      DATA FL3/'HDB/','DAT '/
      WRITE(3,1) CNTRL
 1    FORMAT(1X,A1,' WELCOME TO U S E R S-program'////)
      WRITE(3,2)
 2    FORMAT(' THIS PROGRAM CAN   -----------------------------------'
     +        /'                         (i) List  Station characteristcs & data'
     + /'                      (ii) Esimate missing data                '
     + /'                     (iii) Give statististic of data           '
     + /'                        (v) Give harmonics of data             '
     +        /'                         -----------------------------'
     + /' FOR ANY YEAR AND FOR ANY STATION ESTABLISHED IN DATA BASE'
     + //)
      CALL PERMIT
      CALL OPEN(6,FL1,Ø)
      CALL OPEN(7,FL2,Ø)
      CALL OPEN(8,FL3,Ø)
100   WRITE(3,3)
3     FORMAT(' Station : ')
      READ(3,4) KST
4     FORMAT(3I1)
      WRITE(3,5)
5     FORMAT(' Year : ')
      READ(3,6) NYEAR
6     FORMAT(I4)
      CALL DATAIN( NYEAR, KST, RX )
      NMS=Ø
      DO 1Ø1 I = 1 , 53
      IF (RX(1) .EQ. -997.Ø) NMS=NMS+1
1Ø1   X(I) = RX(1)
      IF (NMS .EQ. Ø ) GO TO 991
      CALL MISS1
      NMS=Ø
      DO 113 I = 1 , NTOTAL
      IF( X(1) .EQ. -997.Ø) NMS = NMS + 1
113   CONTINUE
991   NST = 1ØØ*KST(1) + 1Ø*KST(2) + KST(3)
      CALL ADDRES(NST,NYEAR,NREC)
      IF (NREC .EQ. Ø) GO TO 998
      WRITE(8,REC=NREC) NST,NYEAR,NTOTAL,NMS,X
998   WRITE(3,111) CNTRL
111   FORMAT(1X,A1,/////2ØX,'ENTER USERS SELECTION'/2ØX,21('='),/,
     +         /1ØX,'1. Listing of station characteristics '
     + /1ØX,'2. Listing of raw data '
     :         /1ØX,'3. Listing of refined data (missing data estimated)'
     +  /1ØX,'4. Statistical analysis of data'
     +  /1ØX,'5. Harmonic analysis of data'
     +  /1ØX,'6. Exit from program'////9X,' ')
        READ(3,222) INDEX
222     FORMAT(I1)
        IF( INDEX .GE. 1 .AND. INDEX .LE. 6 ) GO TO 999
        GO TO 998

999     GO TO (1111,2222,3333,4444,5555,6666), INDEX

1111    CALL LSTSTN
        GO TO 998
2222    CALL LIST(Ø)
        GO TO 998
3333    CALL LIST(1)
        GO TO 998
4444    CALL STAT
        GO TO 998
5555    CALL HARM
        GO TO 998
6666    WRITE(3,6667) KST
6667    FORMAT(////1X,'Analysis for station ',3I1,' is completed'
     +          /1X,'Do you want a new phase (Y/N) : ')
        READ(3,6668) ANS
6668    FORMAT(A1)
        IF ( ANS .EQ. NO ) STOP USERS
        GO TO 1ØØ
        END
```

```
      SUBROUTINE DATAIN (NYEAR, KST,RX)
      INTEGER DAY, YEAR
      BYTE ANS, NO,CNTRL
      DIMENSION ICB(1Ø),JCB(1Ø),KEYS(15),IBUF(564),RX(53),
     +        KST(3),IST(3),IH(31,12),NDAYS(12),FNM1(4),FNM2(4)
      COMMON /DATACM/ X(53),IB(64),JST(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL,NMS
      EQUIVALENCE (IST(1),KEYS(1)),(IH(1,1),IBUF(1))
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +        (ELMPSL,IB(38)), (ELMPGL,IB(4Ø)), (DEPTH,IB(42))
      DATA KEYS/15*Ø/,ICB/7,Ø,12,7*Ø/,JCB/6,Ø,3,Ø,Ø,Ø,Ø,Ø,Ø,Ø/,
     +        NDAYS/31,28,31,30,31,30,31,31,30,31,30,31/,CNTRL/30/
      WRITE(3,333) CNTRL
333   FORMAT(1X,A1)
      ICB(4) = -1
      CALL TRNLMN(ICB,KEYS)
      IF(ICB(5) .EQ. Ø) GO TO 2
      CALL TRNERR(ICB,KEYS)
      RETURN
2     DO 222 I = 1 , 3
      JST(I) = KST(I)
222   IST(I) = KST(I)
      CALL TRNR(JCB,KEYS,IB)
      IF(JCB(5) .EQ. Ø) GO TO 5
      IF(JCB(5) .EQ. 4) GO TO 3
      CALL TRNERR(JCB,KEYS)
      RETURN
3     WRITE(3,4)
4     FORMAT(' ! Unknown station')
      RETURN
5     YEAR = NYEAR
      CALL STOKEY(7,4,YEAR,KEYS)
      CALL TRNR(ICB,KEYS,IBUF)
      IF(ICB(5) .EQ. Ø) GO TO 6
      CALL TRNERR(ICB,KEYS)
      RETURN
6     CONTINUE
      IF(MOD(YEAR,4) .EQ. Ø) NDAYS(2) = 29
      N = Ø
      DO 7 M = 1 , 12
      DO 7 J = 1 , 31
      IF( J .GT. NDAYS(M)) GO TO 7
      IF( IH(J,M) .EQ. -999) GO TO 7
      IF( IH(J,M) .EQ. -998) GO TO 7
      N = N + 1
      IF ( IH(J,M) .EQ. -997 ) GO TO 67
      RX(N) = FLOAT(IH(J,M))/1ØØ.Ø
      RX(N) = ELMPSL - RX(N)
      GO TO 68
67    RX(N) =  -997.Ø
68    CONTINUE
      DAY(N) = J
      MONTH(N) = M
7     CONTINUE
      NTOTAL = N
      RETURN
      END
      BLOCK DATA
      INTEGER DAY,YEAR
      COMMON /CNTR/ LUN,IPS,IPW,LZ,IS(128),LSPS,LSLUN,NWFG
      COMMON /CTRN/LUNL,LUNH,NCX,LTF,LTB,IXZ,ITB(435),KEY(15),ND1(15),
     +  NDS(15)
      COMMON /DATACM/ X(53),IB(64),JST(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL, NMS
      DATA LUNL,LUNH/6,10/,NCX/-1/,LTF/145/,LTB/435/,IXZ/290/
      DATA ITB/85*0,2,14*0,1,44*0,85*0,2,14*0,1,44*0,85*0,2,14*0,
     +   1,44*0/
      DATA LSPS,LSLUN,NWFG/3*-1/,IB/30*'  ',34*0/
      END
              SUBROUTINE LSTSTN
      INTEGER YEAR, DAY
      BYTE CNTRL, VIDEO, PRINTR, ANS
      COMMON /DATACM/ X(53),IB(64), JST(3), DAY(53), MONTH(53), YEAR
     +       , NTOTAL, NMS
      EQUIVALENCE  ( COLAMI,IB(33) ),( COLOMI,IB(36) ),
     +        ( ELMPSL,IB(38) ),(ELMPGL,IB(40) ),( DEPTH,IB(42) )
      DATA CNTRL/30/,VIDEO/'V'/, PRINTR/'P'/
      LO=3
      WRITE(3,1)
1     FORMAT(' Printing desired on Video-screen/Printer (V/P) : ')
      READ(3,2) ANS
2     FORMAT(A1)
      IF ( ANS .EQ. PRINTR ) LO = 2
      IF ( LO .EQ. 3 ) WRITE(3,3) CNTRL
3     FORMAT(1X,A1)
      WRITE(LO,4)
4     FORMAT(///1X,'Basic characteristics of station'/1X,32('-')//)
      WRITE(LO,5) (IB(I),I=1,10), JST
5     FORMAT(1X,'OBSERVATION WELL NO : ',10A2,'/(','M-',3I1,')',)
      WRITE(LO,6) (IB(I),I=21,25)
6     FORMAT(1X,'Old No ( if any )   : ',5A2)
      WRITE(LO,7) (IB(I),I=11,20)
7     FORMAT(1X,'Location (Village ) : ',10A2)
      WRITE(LO,8) (IB(I),I=26,30)
8     FORMAT(1X,'Type of Well        : ',5A2)
      WRITE(LO,9) DEPTH
9     FORMAT(1X,'Depth of Well       : ',F9.3,' metre')
      WRITE(LO,10) ELMPSL
10    FORMAT(1X,'Elevation of MP     : ',F9.3,' metre above MSL')
      WRITE(LO,11) ELMPGL
11    FORMAT(1X,'Height of MP        : ',F9.3,' metre above GL')
      WRITE(LO,12) IB(32),COLAMI,IB(35),COLOMI
12    FORMAT(1X,'Latitude : ',I4,' deg ',F4.1,' min / Longitude : '
     +        ,I4,' deg ',F4.1,' min ')
      WRITE(LO,13) IB(31)
13    FORMAT(1X,'Availability of records ( since ) : ',I4)
      CALL PERMIT
      RETURN
      END
              SUBROUTINE LIST(INDEX)
      INTEGER DAY,YEAR
      BYTE CNTRL, ANS, YES, NO, LPRINT, CONSOL
      DIMENSION  X(53)
      COMMON /DATACM/ P(53),IB(64),KSI(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL, NMS
      COMMON /RAWDAT/ RX(53)
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +        (ELMPSL,IB(38)), (ELMPGL,IB(4Ø)), (DEPTH,IB(42))
      DATA CNTRL/3Ø/,YES/'Y'/,NO/'N'/,LPRINT/'L'/,CONSOL/'P'/
      DO 99 I = 1, NTOTAL
      IF (INDEX .EQ. Ø) X(I) = RX(I)
      IF (INDEX .EQ. 1) X(I) = P(I)
99    CONTINUE
      WRITE(3,1) CNTRL
1     FORMAT(1X,A1,'Listing desired on (PRINTER, CONSOL) (L/P) : ')
      READ(3,2) ANS
2     FORMAT(A1)
      LO = 3
      IF( ANS .EQ. LPRINT) LO = 2
      IF( LO .EQ. 3) WRITE(3,3) CNTRL
3     FORMAT(1X,A1)
      WRITE(LO,4)
4     FORMAT(1X,
     +        'SL    DATE OF        WATER LEVEL    SL    DATE OF      '
     +  'WATER LEVEL',/1X,
     +        'NO.   OBSERVATION  FROM MSL(m)     NO.   OBSERVATION '
     +  'FROM MSL(m)'/)
      DO 7 I = 1 , 26
      J= I + 26
      IF( LO .EQ. 3 .AND. I .EQ. 14 ) CALL PERMIT
      IF( X(I) .EQ. -997.Ø .AND. X(J) .EQ. -997.Ø) WRITE(LO,8)
     +        I,DAY(I),MONTH(I),YEAR,J,DAY(J),MONTH(J),YEAR
      IF(X(I) .EQ. -997.Ø .AND. X(J) .NE. -997.Ø)  WRITE(LO,9)
     +        I,DAY(I),MONTH(I),YEAR,J,DAY(J),MONTH(J),YEAR,X(J)
      IF(X(I) .NE. -997.Ø .AND. X(J) .EQ. -997.Ø) WRITE(LO,1Ø)
     +        I,DAY(I),MONTH(I),YEAR,X(I),J,DAY(J),MONTH(J),YEAR
      IF(X(I) .NE. -997.Ø .AND. X(J) .NE. -997.Ø) WRITE(LO,11)
     +        I,DAY(I),MONTH(I),YEAR,X(I),J,DAY(J),MONTH(J),YEAR,X(J)
7     CONTINUE
8     FORMAT(1X,I2,'.   ',I2,'/',I2,'/',I4, 4X,' -----',8X,I2,'.   ',
     +           2(I2,'/'),I4, 4X,' -----')
9     FORMAT(1X,I2,'.   ',2(I2,'/'),I4, 4X,' -----', 8X,I2,
     +        '.   ',   2(I2,'/'),I4,4X,F6.3)
1Ø    FORMAT(1X,I2,'.   ',2(I2,'/'),I4, 4X,F6.3, 8X,I2,'.   ',
     +           2(I2,'/'),I4, 4X, ' -----')
11    FORMAT(1X,I2,'.   ',2(I2,'/'),I4, 4X,F6.3, 8X, I2,'.   ',
     +           2(I2,'/'),I4,4X,F6.3,I4, 4X, F6.3)
      IF(X(53) .EQ. -997.Ø) GO TO 91
      IF( NTOTAL .EQ. 53) WRITE(LO,12) DAY(53),MONTH(53),YEAR,X(53)
12    FORMAT(35X,'53.   ',2(I2,'/'),I4,4X,F6.3)
      CALL PERMIT
      RETURN
91    WRITE(LO,13) DAY(53),MONTH(53),YEAR
13    FORMAT(53X,'53.   ',2(I2,'/'),I4,4X,' -----')
      CALL PERMIT
      RETURN
      END
                SUBROUTINE MISS1
      INTEGER DAY, YEAR
      BYTE ANS, YES, NO
      DIMENSION XX(10), YY(10)
      COMMON /DATACM/ X(53),IB(64),JST(3),DAY(53),MONTH(53),YEAR,
     +          NTOTAL, NMS
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +          (ELMPSL,IB(38)), (ELMPGL,IB(40)), (DEPTH,IB(42))
      DATA NB/3/,FR/ 1.0/,LO/3/, YES/'Y'/, NO/'N'/
      NB1=NB+1
      NBD=2*NB
      CALL NMISS(NESC)
      IF( NESC .EQ. 111) GO TO 1
      WRITE(3,2)
2     FORMAT(/1X,'Missing data estimation by Spline-fit is aborted'
     +          /1X,'Do you wish to continue estimating by Surface-fitting ?
     +          ,'(Y/N) : ')
      READ(3,33) ANS
33    FORMAT(A1)
      IF (ANS .EQ. NO) RETURN
      CALL MISS2
      RETURN
1     IS = NB1
      IF = NTOTAL - NB
      DO 3 I = IS , IF
      IF( X(I) .NE. -997.0) GO TO 3
      IF( X(I+1) .EQ. -997.0 ) GO TO 4
      DO 11 J = 1 , NB
      XX(J) = FLOAT(J)*FR
      K1 = I + J - NB - 1
11    YY(J) = X(K1)
      XINT = XX(NB) + FR
      DO 12 J = NB1 , NBD
      XX(J) = FLOAT(J+1)*FR
      K2 = I+J-NB
12    YY(J) = X(K2)
C     WRITE(3,99) NBD , XX , YY
99    FORMAT(/1X,I3/(1X,5F10.3/5F10.3/5F10.3/5F10.3))
      CALL SPLIN(XX,YY,NBD,XINT,YINT)
      X(I) = YINT
      GO TO 3
4     DO 13 J = 1 , NB
      XX(J) = FLOAT(J)*FR
      K4 = I+J-NB-1
13    YY(J) = X(K4)
      XINT = XX(NB) + FR
      DO 14 J = NB1 , NBD
      XX(J) = FLOAT(J+2)*FR
      K3 = I+J-NB+1
      YY(J) = X(K3)
14    CONTINUE
      CALL SPLIN(XX,YY,NBD,XINT,YINT)
      X(I) = YINT
```

```
3     CONTINUE
      RETURN
      END
      SUBROUTINE SPLIN(X,Y,M,XINT,YINT)
      DIMENSION X(1Ø),Y(1Ø),C(4,1Ø)
      IF(XINT - X(1)) 7 , 1 , 2
1     YINT = Y(1)
      RETURN
2     K = 1
3     IF(XINT-X(K+1)) 6 , 4 , 5
4     YINT = Y(K+1)
      RETURN
5     K = K + 1
      IF(M-K) 7 , 7 , 3
C 6   WRITE(3,97) M , XINT , YINT
97    FORMAT(1X,I3,2X,F1Ø.3,2X,E1Ø.3)
6     CALL SPLICO(X,Y,M,C)
      YINT = (X(K+1)-XINT)*(C(1,K)*(X(K+1)-XINT)**2 + C(3,K))
      YINT = YINT + (XINT-X(K))*(C(2,K)*(XINT-X(K))**2 + C(4,K))
C     WRITE(3,92) X(K),X(K+1),(C(I,K),I=1,4)
92    FORMAT(1X,6E12.5)
C     WRITE(3,93) M , XINT , YINT
93    FORMAT(/1X,I4,2X,2E12.5)
      RETURN
7     WRITE(3,8)
8     FORMAT(' OUT OF RANGE FOR INTERPOLATION')
      END
      SUBROUTINE SPLICO(X,Y,M,C)
      DIMENSION X(1Ø),Y(1Ø),D(1Ø),P(1Ø),E(1Ø),C(4,1Ø),A(1Ø,3),
     +          B(1Ø),Z(1Ø)
C     WRITE(3,95) M , X , Y
95    FORMAT(1X,'M,X,Y',I3,/(1X,5F1Ø.3))
      MM = M - 1
      DO 2 K = 1 , MM
      D(K) = X(K+1) - X(K)
      P(K) = D(K)/6.Ø
2     E(K) = (Y(K+1)-Y(K))/D(K)
      DO 3 K = 2, MM
3     B(K) = E(K) - E(K-1)
      A(1,2) = -1.Ø - D(1)/D(2)
      A(1,3) = D(1)/D(2)
      A(2,3) = P(2) - P(1)*A(1,3)
      A(2,2) = 2.Ø*(P(1) + P(2)) - P(1)*A(1,2)
      A(2,3) = A(2,3)/A(2,2)
      B(2) = B(2)/A(2,2)
      DO 4 K = 3 , MM
      A(K,2) = 2.Ø*(P(K-1) + P(K)) - P(K-1)*A(K-1,3)
      B(K) = B(K) - P(K-1)*B(K-1)
      A(K,3) = P(K)/A(K,2)
4     B(K) = B(K)/A(K,2)
      Q = D(M-2)/D(M-1)
      A(M,1) = 1.Ø + Q + A(M-2,3)
      A(M,2) = -Q - A(M,1)*A(M-1,3)
      B(M) = B(M-2)-A(M,1)*B(M-1)
      Z(M) = B(M)/A(M,2)
      MN = M - 2
      DO 6 I = 1 , MN
      K = M - 1
6     Z(K) = B(K) - A(K,3)*Z(K+1)
      Z(1) = -A(1,2)*Z(2)-A(1,3)*Z(3)
      DO 7 K = 1 , MM
      Q = 1.Ø/(6.Ø*D(K))
      C(1,K) = Z(K)*Q
      C(2,K) = Z(K+1)*Q
      C(3,K) = Y(K)/D(K)-Z(K)*P(K)
7     C(4,K) = Y(K+1)/D(K)-Z(K+1)*P(K)
      RETURN
      END
      SUBROUTINE NMISS( NESC )
      INTEGER DAY, YEAR
      DIMENSION MS(53)
      COMMON /DATACM/ X(53),IB(64),JST(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL, NMS
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +        (ELMLSL,IB(38)), (ELMPGL,IB(4Ø)), (DEPTH,IB(42))
      DATA NB/3/,NB1/4/
      NT = Ø
      DO 1 I = 1 , NTOTAL
      IF(X(I) .NE. -997.Ø) GO TO 1
      NT = NT + 1
      MS(NT) = I
1     CONTINUE
      NESC = 999
      IF ( NT .EQ. 1 .AND. MS(1) .GE. NB1 ) GO TO 9
      IF( NT .EQ. 2 .AND. MS(1) .GE. NB1 .AND. (MS(2)-MS(1))
     +        .GE. NB1 .AND. MS(2) .LT. (NTOTAL-NB1)) GO TO 9
      IF ( NT .EQ. 2 .AND. MS(1) .GE. NB1 .AND. (MS(2)-MS(1))
     +        .EQ. 1 .AND. MS(2) .LT. (NTOTAL-NB1) ) GO TO 9
      DO 2 I = 3 , NT
      IF(MS(I) .LE. NB) GO TO 7
      IF( (MS(I)-MS(I-1)) .EQ. (MS(I-1)-MS(I-2))) GO TO 6
      IF( (MS(I)-MS(I-1)) .GE. 4) NESC = 111
2     CONTINUE
      GO TO 9
6     NESC = 222
      RETURN
7     NESC = 222
      RETURN
9     NESC = 111
      RETURN
      END
            SUBROUTINE MISS2
      INTEGER STN, RSTN,ST, YEAR, DAY
      DIMENSION D(12), ST(1Ø), RSTN(6,1Ø), SIDE(12,1Ø),FACTOR(6)
     +        , XS(6,53), XH(53),JST(3)
      COMMON /DATACM/ X(53),IB(64),KST(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL , NMS
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +        (ELMPSL,IB(38)), (ELMPGL,IB(4Ø)), (DEPTH,IB(42))
      DATA ST/24,47,8*Ø/, RSTN/42,25,72,3*Ø,43,42,72,2,2*Ø,48*Ø/
      DATA SIDE/5.25,7.35,9.7,3*Ø.Ø,12.Ø,11.45,12.6,3*Ø.Ø,
     +          7.65,9.55,3.7,4.35,2*Ø.Ø,11.6,12.6,3.7,6.9,2*Ø.Ø,96*Ø.Ø/
C
C     SEARCH FOR THE STATION
C
      STN = 1ØØ*KST(1)+1Ø*KST(2)+KST(3)
      DO 11 NST = 1 , 1Ø
      IF( ST(NST) .EQ. STN ) GO TO 12
11    CONTINUE
      WRITE( 3 ,3) STN
3     FORMAT( / 1X, '! OUT OF RANGE')
77    WRITE(3,1Ø)
1Ø    FORMAT(1X,'Misiing data estimation is still aborted')
      CALL PERMIT
      RETURN
12    DO 22 NSIDE = 1 , 6
      IF( SIDE(NSIDE,NST) .EQ. Ø.Ø) GO TO 13
22    CONTINUE
      GO TO 14
13    NSIDE = NSIDE - 1
14    CONTINUE
      DO 33 K = 1 , NSIDE
      A = SIDE(K,NST)
      B = SIDE(K+1,NST)
      IF ( K .EQ. NSIDE ) B = SIDE(1,NST)
      C = SIDE(K+6,NST)
      CALL AREA( A , B , C , D , K)
33    CONTINUE
      FACTOR(1) = D(1) + D(2*NSIDE)
      DO 44 K = 2 , NSIDE
      FACTOR(K) = D(2*K-2) + D(2*K-1)
44    CONTINUE
      SUMFR=Ø
      DO 3Ø3 I = 1 , NSIDE
3Ø3   SUMFR = SUMFR + FACTOR(I)
      DO 1Ø1 I = 1, NSIDE
      NSTN = RSTN(I,NST)
      CALL SPLSTN(NSTN,JST)
      NYEAR= YEAR
      CALL DATAIN(NYEAR,JST,XH)
      DO 1Ø3 J = 1 , 53
1Ø3   XS(I,J) = XH(J)
1Ø1   CONTINUE
      DO 2Ø2 I = 1 , NTOTAL
      IF ( X(I) .NE. -997.Ø ) GO TO 2Ø2
      XSUM=Ø
      DO 2Ø3 K = 1 , NSIDE
      IF (XS(K,I) .EQ. -997.Ø) GO TO 77
2Ø3   XSUM = XSUM + FACTOR(K)*XS(K,I)
      X(I) = XSUM/SUMFR
2Ø2   CONTINUE
      CALL PERMIT
      RETURN
      END
      SUBROUTINE AREA(A,B,C,D,K)
      DIMENSION D(12), FACTOR(12)
      COSQ = (A**2 +C**2 -B**2)/(2.0*A*C)
      SINQ = SQRT(1.0-COSQ**2)
      D(2*K-1) = A*COSQ*A*SINQ/2.0
      D(2*K)   = A*SINQ*(C-A*COSQ)/2.0
      RETURN
      END
              SUBROUTINE PERMIT
      BYTE ANS, YES, NO
      DATA YES/'Y'/ , NO/'N'/
      WRITE(3,1)
1     FORMAT(//1X,' Press ENTER to Continue '//)
      READ(3,2) NCONT
2     FORMAT(I1)
      RETURN
      END
      SUBROUTINE SPLSTN( NSTN , IST )
      DIMENSION IST(3)
C
C     000 < NSTN < 999
C
      DO 1 I = 1 , 9
      I3 = I*100
      IF( I3 .GT. NSTN) GO TO 2
1     CONTINUE
2     IST(1) = I - 1
      NSTN = NSTN - IST(1)*100
      DO 3 I = 1 , 9
      I2 = I*10
      IF( I2 .GT. NSTN ) GO TO 4
3     CONTINUE
4     IST(2) = I - 1
      IST(3) = NSTN - IST(2)*10
      RETURN
      END
              SUBROUTINE STAT
      INTEGER DAY,YEAR
      BYTE ANS , PRINTR, CONSOL, CNTRL
      COMMON /DATACM/ X(53),IB(64),KST(3),DAY(53),MONTH(53),YEAR,
     +        NTOTAL, NMS
      EQUIVALENCE (COLAMI,IB(33)), (COLOMI,IB(36)),
     +        (ELMPSL,IB(38)), (ELMPGL,IB(40)), (DEPTH,IB(42))
      DATA PRINTR/'P'/, CONSOL/'C'/, CNTRL/30/
      INDEX = 0
10    WRITE(3,1) CNTRL
1     FORMAT(1X,A1,20X,'DESIRED STATISTICS------------------------------'
     +        //27X,'1.  Highest Level above MSL'
     +        /27X, '2.  Lowest  Level above MSL'
     +        /27X, '3.  Depth of lowest Level from GL'
     +        /27X, '4.  Range of fluctuation'
     +        /27X, '5.  Mean level'
     + /27X, '6.  Standard Deviation'
     +        /27X, '7.  Variance'
     + /27X, '8.  All of above'
     +        /27X, '9.  Exit from subroutine'/)
```

```
      READ(3,2) NCHOIC
 2    FORMAT(I1)
      IF( NCHOIC .GE. 1 .AND. NCHOIC .LE. 9) GO TO 3
      GO TO 10
 3    INDEX = INDEX + 1
      IF ( INDEX .GT. 1 ) GO TO 99
      XMIN = X(1)
      XMAX = X(1)
      DO 7 I = 2 , NTOTAL
      IF(X(I) .GT. XMAX) XMAX=X(I)
      IF(X(I) .LT. XMIN) XMIN=X(I)
 7    CONTINUE
      RANGE = XMAX - XMIN
      XMINGL = ELMPSL-XMIN-ELMPGL
      SX= 0.0
      SXX=0.0
      DO 8 I = 1, NTOTAL
      SX = SX + X(I)
      SXX = SXX + X(I)**2
 8    CONTINUE
      SSX = SX*SX
      VAR = (SXX-SSX/FLOAT(NTOTAL))/FLOAT(NTOTAL)
      SD = SQRT(VAR)
      XMEAN = SX/FLOAT(NTOTAL)
 99   IF( NCHOIC .EQ. 9 ) GO TO 98
      WRITE(3,4)
 4    FORMAT(/1X,' Printing desired on PRINTER, CONSOL (P/C) : ')
      READ(3,5) ANS
 5    FORMAT(A1)
      LO = 3
      IF ( ANS .EQ. PRINTR ) LO = 2
 98   GO TO (11,22,30,33,44,55,66,77,88), NCHOIC
 11   WRITE(LO,12) XMAX
 12   FORMAT(/1X, 'Highest Level = ',F7.3,' metre above MSL ')
      CALL PERMIT
      GO TO 10
 22   WRITE(LO,23) XMIN
 23   FORMAT(/1X, 'Lowest Level  = ',F7.3,' metre above MSL')
      CALL PERMIT
      GO TO 10
 30   WRITE(LO,31) XMINGL
 31   FORMAT(/1X, 'Depth of lowest leveL from GL = ',F7.3,'-metre ')
      CALL PERMIT
      GO TO 10
 33   WRITE(3,34) RANGE
 34   FORMAT(/1X, 'Range of fluctuation = ',F7.3,' metre')
      CALL PERMIT
      GO TO 10
 44   WRITE(LO,45) XMEAN
 45   FORMAT(/1X, 'Mean Level = ',F7.3, ' metre')
      CALL PERMIT
      GO TO 10
 55   WRITE(LO,56) SD
 56   FORMAT(/1X, 'Standard deviation = ',F7.3, ' metre')
      CALL PERMIT
      GO TO 1Ø
66    WRITE(LO,67) VAR
67    FORMAT(/1X, 'Variane = ',F8.3,' sq. metre')
      CALL PERMIT
      GO TO 1Ø
77    WRITE(LO,78) XMAX, XMIN, XMINGL,RANGE , XMEAN, SD, VAR
78    FORMAT(/1X, 'Highest Level = ',F7.3, ' metre above MSL'/
     +                 1X, 'Lowest  Level = ',F7.3, ' metre above MSL'/
     +                 1X, 'Depth of lowest level from GL = ',F7.3,' metre'/
     +                 1X, 'Range of fluctuation = ',F7.3, ' metre'/
     +                 1X, 'Mean Level = ',F7.3, ' metre'/
     +                 1X, 'Standard deviation = ',F7.3,' metre'/
     +            1X, 'Variance = ',F8.3, ' sq. metre'/)
      CALL PERMIT
      GO TO 1Ø
88    RETURN
      END
ØØ          SUBROUTINE HARM
      WRITE(3,1)
1     FORMAT(//1X,'Well, If you are interested in harmonic analysis'
     +        ,' then exit this system'/1X,'From system type H ; press ENTER'
     +        ,' and proceed'///)
      CALL PERMIT
      RETURN
      END
            SUBROUTINE ADDRES(NST,NYEAR,NREC)
      INTEGER*1  STN(10)
      DATA STN/2,24,25,42,43,47,65,72,73,74/
      NREC=Ø
      IF ( NYEAR .LT. 1979 .OR. NYEAR .GT. 1983 ) GO TO 7
      DO 1 I = 1 , 1Ø
      IF ( NST .EQ. STN(I)) GO TO 4
1     CONTINUE
4     NSTNUM=I
      IF ( NSTNUM .EQ. 1Ø .AND. NST .NE. STN(1Ø) ) GO TO 7
      NREC = (NSTNUM-1)*5 + NYEAR - 1978
7     RETURN
      END
      PROGRAM HARMON
        INTEGER YEAR, STN, ADDR
      BYTE ANS, CNTRL, YES, NO
        DIMENSION X(53),AMP(53),PHA(53),AMM(53),AMP2(53) ,
     +  DIFF(53),A(53),B(53),H(26,53),XXM(53),RES(26),XR(53),
     +  P(26), DP(26), DDP(26),IST(3), FL(2),R1(53),R2(53)
      DATA FL/'HDB/','DAT '/, PI/3.1415927/ ,PMIN/.0167/,CNTRL/30/
      WRITE(3,1) CNTRL
1     FORMAT(1X,A1)
      WRITE(3,4)
4     FORMAT(1X,'Station : ')
      READ(3,2) IST
2     FORMAT(3I1)
      WRITE(3,5)
5     FORMAT(//1X,'Year : ')
      READ(3,6) NYEAR
6     FORMAT(I4)
      STN = 100*IST(1) + 10*IST(2) + IST(3)
      CALL ADDRES(STN, NYEAR, NREC )
      IF ( NREC .EQ. Ø ) GO TO 7
      CALL OPEN(8,FL,Ø)
      READ(8,REC=NREC) STN , YEAR, NTOTAL, NMS, X
      ENDFILE 8
      IF (NMS .GT. Ø ) GO TO 9
      GO TO 8
7     WRITE(3,37)
37    FORMAT(//1X,'Data not found'//)
      STOP
38    FORMAT(//1X,'Analysis is aborted due to presence of missing'
     +           ,' data')
8     CONTINUE
      N= NTOTAL
      W= FLOAT(N)
      C = 2.Ø
      SX= Ø.Ø
      SXX=Ø.Ø
      DO 25 I = 1 , N
      SX = SX+ X(I)
      SXX = SXX + X(I)**2
25    CONTINUE
      XM = SX/FLOAT(N)
      SSX = SX*SX
      VAR= (SXX-SSX/FLOAT(N))/FLOAT(N)
      SD  = SQRT(VAR)
        DO 35 K=1,26
        FK=K
        Y=Ø.Ø
        Z=Ø.Ø
        DO 26 J=1,N
        XXM(J)=(X(J)-XM)
        FJ=J
        Y=Y+XXM(J)*COS(2.ØØ*PI*FK*FJ/FLOAT(N))
26      Z=Z+XXM(J)*SIN(2.ØØ*PI*FK*FJ/FLOAT(N))
        A(K)=2.Ø/FLOAT(N)*Y
        B(K)=2.Ø/FLOAT(N)*Z
        PA=A(K)
        PB=B(K)
        AMP(K)=SQRT(PA*PA+PB*PB)
        AMP2(K) = AMP(K)**2/2.0
        AMM(K)=AMP(K)/AMP( 1)*100.0
        PHA(K)=ATAN(-PB/PA)*180.0/PI
        IF(PHA(K) .LT. 0.0)PHA(K)=PHA(K)+360.0
   35   CONTINUE
9     WRITE(2,81) SIN, NYEAR
81    FORMAT(//////1X,'Harmonic analysis of groundwater level data',
     +        /1X,'OBSERVATION WELL NO : M -',I2,
     +        /1X,'Year : ',I4///)
      IF (NMS .GT. 0 ) WRITE(2,38)
      IF (NMS .GT. 0 ) STOP
      WRITE(2,82)
82    FORMAT(1X,'Groundwater level data above MSL are as follows'///)
      WRITE(2,84) ( X(I),I=1,N )
84    FORMAT(1X,10F10.4)
      WRITE(2,83) XM , SD
83    FORMAT(1X,'Mean of all data = ',F8.3,' metres',
     +        /1X,'Standard deviation = ',F8.3,' sq. metres'///)
      DO 1111 INDEX= 1 , 13
        DO 1001 J=1,N
        SUM=0.0
        RJ=J
        DO 1002 K=1,INDEX
        RK=K
        SUM=SUM+AMP(K)*COS(2.0*PI*RK*RJ/FLOAT(N)+PHA(K)*PI/180.0)
1002  CONTINUE
        H(INDEX,J)=XM+SUM
1001  CONTINUE
      RES(INDEX) = 0.0
        DO 1004 J=1,N
      XR(J) = ( X(J) - H(INDEX,J))/SD
        DIFF(J)= (X(J)-H(INDEX,J))**2
      RES(INDEX) = RES(INDEX) + DIFF(J)
1004  CONTINUE
      WRITE(2,104) INDEX , RES(INDEX)
      CALL ANALYS(N,XR)
C       WRITE(2,103)(J,X(J),H(J),DIFF(J),XXM(J),J=1,N)
 103    FORMAT(4X,'WEEK='I2,4X,'ORIGINAL DATA='F6.3,1X,'   COMPUTED DATA
     + ='F8.3,3X,'DIFFERANCE='F8.3,4X,'ORI. DATA-MEAN)',  F8.3)
 15     CONTINUE
1111    CONTINUE
104   FORMAT(/9X,'Residuals for',I3,' harmonics =',F10.5/9X,38('=')/)
      DO 10 NH= 1 , 26
      SUM = 0.0
      DO 79 I = 1 , NH
      SUM = SUM + AMP2(I)
79    CONTINUE
      P(NH) = SUM/VAR
10    CONTINUE
      WRITE(2,3)
3     FORMAT(//9X,'Sl.No.',13X,'P(I)',13X,'DEL(P(I))',13X,'DDP(I)'/)
      DO 21 NH = 1 , 25
```

```
21      DP(NH) = P(NH+1) - P(NH)
        DO 22 NH= 1. 25
22      DDP(NH) = DP(NH+1) - DP(NH)
        WRITE(2.41) ( I . P(I) , DP(I). DDP(I). I= 1. 25)
41      FORMAT(10X,I3,10X,F10.5,10X,F10.5,10X,F10.5)
        NSIG = 1
        SMIN = RES(1)
        DO 117 I = 1 , 13
        IF ( RES(I) .GE. SMIN ) GO TO 117
        SMIN = RES(I)
        NSIG = I
117     CONTINUE
        WRITE(2, 118) NSIG
118     FORMAT(////1X,'Number of significant harmonics = ',I2//)
        WRITE(2,119)
119     FORMAT(//1X,'Harmonics     Amplitude     Phase '//)
        WRITE(2,227) ( I , AMP(I), PHA(I) , I = 1 , NSIG )
227     FORMAT(4X,I2,5X,F9.3,5X,F9.3)
        DO 228 I = 1 , N
        R1(I) = X(I) - H(NSIG,I)
        R2(I) = R1(I)**2
228     CONTINUE
        R1SUM=0
        R2SUM=0.0
        DO 337 I = 1 , N
        R1SUM=R1SUM + R1(I)
        R2SUM=R2SUM + R2(I)
337     CONTINUE
        WRITE(2,338)
338     FORMAT(//1X,'S1.      OBS.value      CAL.value '//)
        WRITE(2,339) (I,X(I),H(1,I),H(2,I),H(3,I),H(6,I),H(8,I),I=1,N)
339     FORMAT(1X,I2,5X,F9.4,5X,F9.4,5X,F9.4,5X,F9.4,5X,F9.4,5X,F9.4)
          STOP
          END
```

```
         PROGRAM HARMON
           INTEGER YEAR, STN, ADDR
         BYTE ANS, CNTRL, YES, NO
           DIMENSION X(53),AMP(53),PHA(53),AMM(53),AMP2(53) ,
        +  DIFF(53),A(53),B(53),H(26,53),XXM(53),RES(26),XR(53),
        +  P(26), DP(26), DDP(26),IST(3), FL(2),R1(53),R2(53)
         DATA FL/'HDB/','DAI '/, PI/3.1415927/ ,PMIN/.016/,CNTRL/30/
         WRITE(3,1) CNTRL
1        FORMAT(1X,A1)
         WRITE(3,4)
```

```
      PROGRAM HARMON
        INTEGER YEAR, STN, ADDR
      BYTE ANS, CNTRL, YES, NO
        DIMENSION X(53),AMP(53),PHA(53),AMM(53),AMP2(53) ,
     +  DIFF(53),A(53),B(53),H(26,53),XXM(53),RES(26),XR(53),
     +  P(26), DP(26), DDP(26),IST(3), FL(2),R1(53),R2(53)
      DATA FL/'HDB/','DAT '/, PI/3.1415927/ ,PMIN/.Ø167/,CNTRL/3Ø/
      WRITE(3,1) CNTRL
 1    FORMAT(1X,A1)
      WRITE(3,4)
 4    FORMAT(1X,'Station : ')
      READ(3,2) IST
 2    FORMAT(3I1)
      WRITE(3,5)
 5    FORMAT(//1X,'Year : ')
      READ(3,6) NYEAR
 6    FORMAT(I4)
      STN = 1ØØ*IST(1) + 1Ø*IST(2) + IST(3)
      CALL ADDRES(STN, NYEAR, NREC )
      IF ( NREC .EQ. Ø ) GO TO 7
      CALL OPEN(8,FL,Ø)
      READ(8,REC=NREC) STN , YEAR, NTOTAL, NMS, X
      ENDFILE 8
      IF (NMS .GT. Ø ) GO TO 9
      GO TO 8
 7    WRITE(3,37)
 37   FORMAT(//1X,'Data not found'//)
      STOP
 38   FORMAT(//1X,'Analysis is aborted due to presence of missing'
     +         ,' data')
 8    CONTINUE
      N= NTOTAL
      W= FLOAT(N)
      C = 2.Ø
      SX= Ø.Ø
      SXX=Ø.Ø
      DO 25 I = 1 , N
      SX = SX+ X(I)
      SXX = SXX + X(I)**2
 25   CONTINUE
      XM = SX/FLOAT(N)
      SSX = SX*SX
      VAR= (SXX-SSX/FLOAT(N))/FLOAT(N)
      SD  = SQRT(VAR)
        DO 35 K=1,26
        FK=K
        Y=Ø.Ø
        Z=Ø.Ø
        DO 26 J=1,N
        XXM(J)=(X(J)-XM)
        FJ=J
        Y=Y+XXM(J)*COS(2.ØØ*PI*FK*FJ/FLOAT(N))
 26     Z=Z+XXM(J)*SIN(2.ØØ*PI*FK*FJ/FLOAT(N))
        A(K)=2.Ø/FLOAT(N)*Y
        B(K)=2.Ø/FLOAT(N)*Z
        PA=A(K)
        PB=B(K)
        AMP(K)=SQRT(PA*PA+PB*PB)
        AMP2(K) = AMP(K)**2/2.0
        AMM(K)=AMP(K)/AMP( 1)*100.0
        PHA(K)=ATAN(-PB/PA)*180.0/PI
        IF(PHA(K) .LT. 0.0)PHA(K)=PHA(K)+360.0
   35   CONTINUE
9     WRITE(2,81) SIN, NYEAR
81    FORMAT(//////1X,'Harmonic analysis of groundwater level data',
     +         /1X,'OBSERVATION WELL NO : M  ',I2,
     :         /1X,'Year : ',I4///)
      IF (NMS .GT. 0 ) WRITE(2,38)
      IF (NMS .GT. 0 ) STOP
      WRITE(2,82)
82    FORMAT(1X,'Groundwater level data above MSL are as follows'///)
      WRITE(2,84) ( X(I),I=1,N )
84    FORMAT(1X,10F10.4)
      WRITE(2,83) XM , SD
83    FORMAT(1X,'Mean of all data = ',F8.3,' metres',
     +         /1X,'Standard deviation = ',F8.3,' sq. metres'///)
      DO 1111 INDEX= 1 , 13
        DO 1001 J=1,N
        SUM=0.0
        RJ=J
        DO 1002 K=1,INDEX
        RK=K
        SUM=SUM+AMP(K)*COS(2.0*PI*RK*RJ/FLOAT(N)+PHA(K)*PI/180.0)
1002  CONTINUE
        H(INDEX,J)=XM+SUM
1001  CONTINUE
      RES(INDEX) = 0.0
        DO 1004 J=1,N
      XR(J) = ( X(J) - H(INDEX,J))/SD
        DIFF(J)= (X(J)-H(INDEX,J))**2
      RES(INDEX) = RES(INDEX) + DIFF(J)
1004  CONTINUE
      WRITE(2,104) INDEX , RES(INDEX)
      CALL ANALYS(N,XR)
C       WRITE(2,103)(J,X(J),H(J),DIFF(J),XXM(J),J=1,N)
 103    FORMAT(4X,'WEEK='I2,4X,'ORIGINAL DATA='F6.3,1X,'   COMPUTED DATA
     + ='F8.3,3X,'DIFFERANCE='F8.3,4X,'ORI. DATA-MEAN)',   F8.3)
 15     CONTINUE
1111    CONTINUE
104   FORMAT(/9X,'Residuals for',I3,' harmonics =',F10.5/9X,38('=')/)
      DO 10 NH= 1 , 26
      SUM = 0.0
      DO 79 I = 1 , NH
      SUM = SUM + AMP2(I)
79    CONTINUE
      P(NH) = SUM/VAR
10    CONTINUE
      WRITE(2,3)
3     FORMAT(//9X,'Sl.No.',13X,'P(I)',13X,'DEL(P(I))',13X,'DDP(I)'/)
      DO 21 NH = 1 , 25
```

```
21     DP(NH) = P(NH+1) - P(NH)
       DO 22 NH= 1, 25
22     DDP(NH) = DP(NH+1) - DP(NH)
       WRITE(2,41) ( I , P(I) , DP(I), DDP(I), I= 1, 25)
41     FORMAT(10X,I3,10X,F10.5,10X,F10.5,10X,F10.5)
       NSIG = 1
       SMIN = RES(1)
       DO 117 I = 1 , 13
       IF ( RES(I) .GE. SMIN ) GO TO 117
       SMIN = RES(I)
       NSIG = I
117    CONTINUE
       WRITE(2, 118) NSIG
118    FORMAT(////1X,'Number of significant harmonics = ',I2//)
       WRITE(2,119)
119    FORMAT(//1X,'Harmonics     Amplitude     Phase '//)
       WRITE(2,227) ( I , AMP(I), PHA(I) , I = 1 , NSIG )
227    FORMAT(4X,I2,5X,F9.3,5X,F9.3)
       DO 228 I = 1 , N
       R1(I) = X(I) - H(NSIG,I)
       R2(I) = R1(I)**2
228    CONTINUE
       R1SUM=0
       R2SUM=0.0
       DO 337 I = 1 , N
       R1SUM=R1SUM + R1(I)
       R2SUM=R2SUM + R2(I)
337    CONTINUE
       WRITE(2,338)
338    FORMAT(//1X,'Sl.     OBS.value     CAL.value '//)
       WRITE(2,339) (I,X(I),H(1,I),H(2,I),H(3,I),H(6,I),H(8,I),I=1,N)
339    FORMAT(1X,I2,5X,F9.4,5X,F9.4,5X,F9.4,5X,F9.4,5X,F9.4,5X,F9.4)
         STOP
         END
                SUBROUTINE ANALYS(N,X)
         DIMENSION X(53)
       SX= 0.0
       SXX=0.0
       DO 25 I = 1 , N
       SX = SX+ X(I)
       SXX = SXX + X(I)**2
25     CONTINUE
       XM = SX/FLOAT(N)
       SSX = SX*SX
       VAR= (SXX-SSX/FLOAT(N))/FLOAT(N)
       SD  = SQRT(VAR)
       S3=0.0
       S4=0.0
       DO 111 I = 1 , N
       S3= S3 + (X(I)-XM)**3
       S4 = S4 + (X(I)-XM)**4
111    CONTINUE
C      CV=SD/XM
       CS=S3/FLOAT(N)/SD**3
       CE=S4/FLOAT(N)/VAR**2 -3.0
```

```
        WRITE(2,113) XM,SD,CS,CE
113   FORMAT( 9X,'Mean = ',E12.5,5X,' S.D. = ',F8.4,5X,' Cs = '
     +         ,F8.4,5X,' Ce = ',F8.4 )
      RETURN
      END
              SUBROUTINE ADDRES(NST,NYEAR,NREC)
      INTEGER STN(1Ø), YEAR(5)
      DATA STN/2,24,25,42,43,47,65,72,73,74/
      DATA YEAR/1979,198Ø,1981,1982,1983/
      NREC=Ø
      IF ( NYEAR .LT. 1979 .OR. NYEAR .GT. 1983 ) GO TO 7
      DO 1 I = 1 , 1Ø
      IF ( NST .EQ. STN(I)) GO TO 4
1     CONTINUE
4     NSTNUM=I
      IF ( NSTNUM .EQ. 1Ø .AND. NST .NE. STN(1Ø) ) GO TO 7
      DO 2 I = 1 , 5
      IF ( NYEAR .EQ. YEAR(I) ) GO TO 3
2     CONTINUE
3     NYRNUM = I
      NREC = (NSTNUM-1)*5 + NYRNUM
7     RETURN
      END
```