EVOLUTIONS OF COMPUTER CONCEPTS IN A HYDRAULICS LABORATORY

KEY WORDS: computers; hydraulic laboratory; hydraulic problems; real-time control systems

ABSTRACT: Experiences of scientists and engineers in a hydraulic laboratory indicate that electronic computers have been accepted by the research staff, and that computers are being used to some extent in almost every one of the highly diversified problems encountered. Effective use is being made of three digital computers. A teletype time-sharing terminal is being used for small problems, in which fast turnaround time is essential, and for debugging subroutines and programs for larger computers. A medium-speed computer which provides three to four runs per day is being used for medium-size jobs and for final debugging of large programs. A high-speed computer with 24-hour turnaround time is being used for large programs. A subroutine documentation and index system by which the computers document subroutines and maintain an updated index has evolved. Examples from the system which has evolved are included. Also included is an example of the use of the teletype terminal to develop an acceleration-deceleration subroutine for a model of a well backfilling program. Future plans of the laboratory include the study for possible installation of a real-time, data collection, data evaluation, data recording and computer control system.

~~REFERENCE.~~ Enger, P., "Evolutions of Computer Concepts in a Hydraulics Laboratory," Hydraulics Division Conference, August 21-23 1968, MIT.

CONTENTS

EVOLUTIONS OF COMPUTER CONCEPTS IN A HYDRAULIC LABORATORY

By Phillip F. Enger*

Introduction

As computers have become more readily available, most engineers have accepted them as a powerful and easy-to-use tool. Although this acceptance among some engineers may have evolved rather slowly, as the computer has been accepted, engineers have realized the tremendous scope of work that can be turned over to the computer. They have also developed a desire to obtain a better understanding of the capabilities and limitations of the computer in order to use this relatively new tool more effectively. As understanding is developed, the need for different sizes and types of computers becomes apparent. In addition, as computers are used more extensively, the need for adequate documentation becomes imperative.

This paper discusses the evolution of computer usage in the Hydraulics Laboratory of the Bureau of Reclamation at Denver, Colorado. Included is a brief history of computer use in the laboratory, the present status of computers available to the laboratory and their use as well as information on studies of possible future plans.

*Hydraulic Research Engineer, Hydraulics Branch, Bureau of Reclamation, Denver ~~Federal Center,~~ Denver Colorado

The Laboratory

The Laboratory's present permanent staff consists of three engineering administrators, eleven engineers, three technicians, and one secretary.

Although the Laboratory staff is relatively small, presently under study are many and varied problems, including those associated with:

air demand at gates and valves
aquifers and drainage systems
canals
closed conduit systems
collection systems
dams
energy dissipators
fish facilities and spawning beds
flowmeters
gates and valves
inlets and outlets
instrumentation
laterals
open-channel flow
outlet works
pipeline surges
powerplants
pumping plants
repairs and modifications of hydraulic structures
reservoirs
sedimentation
seepage
siphons
spillways
stability of channels
stratified flow
surface protection
transitions
tunnels
turbines
water measurements
wave studies

With this large variety of problems and the relatively small staff, computer usage became imperative. Therefore, all the engineers have been provided with some training in computer programming. Many of the engineers developed rapidly into competent programmers, and at present, computer programs are written as an aid to nearly every study which is undertaken.

Past Training and Computers

The computers which were first made available for laboratory use were relatively small, and were programmable only in computer-oriented language. Although these computers clearly indicated the coming computer potential, only a few of the staff were willing to spend the time necessary to master the computer-oriented language. No formal in-house training in programming using the computer-oriented language was provided and the engineers interested obtained their training from manuals or from local schools. In 1963 the Bureau's Division of Data Processing at Denver made available to the staff a large second-generation computer. The large computer could be programmed in FØRTRAN II, and most engineers immediately became interested in using the computer. As a result, a short 6-hour course in FØRTRAN II was organized and taught.

The goal of the course was to provide the basic knowledge necessary to transform a small problem into a practical computer program which would provide results that could be easily interpreted by the engineer. Soon after the short course was taught, a FØRTRAN IV compiler became available, and a computer manufacturer's representative conducted classes in FØRTRAN IV. The FØRTRAN IV training consisted of 30 hours of instruction over a 2-week interval. Homework, tests, and grades were given. This training provided a firm background for the engineers in the Hydraulics Laboratory.

Since 1964, either a second or a third generation computer, and sometimes both, has been available to the staff. These computers with FØRTRAN IV compilers are capable of handling large problems at high speed. However, until about November 1967, turnaround time (the time interval between submitting a program and obtaining results) usually varied from about 12 to 24 hours. For small problems this represents a relatively long turnaround time.

Present Computers and Use

Records maintained by the laboratory indicate that the most common engineering problem encountered is of the type which requires 1/2- to 2-man-days to complete by manual methods. For these small problems, results are desired immediately. It is impractical to give these small everyday jobs the highest priority and thus provide rapid turnaround time on the large computers. Therefore, with the aid of the Division of Data Processing, a teletype time-sharing terminal was obtained. The teletype terminal communicates with a computer in another city by means of telephone lines. It may be programmed in a FØRTRAN language, or EXTENDED BASIC. EXTENDED BASIC is a mathematical-oriented language, and little formal training in its use is usually required of an engineer familiar with FØRTRAN.

Using the time-sharing terminal, small one-shot problems may be written, compiled, and executed in an acceptable format within minutes. Programs for the time-sharing terminal are written by engineers, and the terminal is operated on an open-shop basis. Technicians use the terminal to execute prewritten programs, or in some cases, to execute short programs of their own. The time-sharing terminal is also frequently used for debugging routines for the larger computers.

A medium-speed second generation computer with 32K core storage and a FØRTRAN IV compiler is avialable for problems which are too large to be readily placed on the time-sharing terminal. Limited access to the medium speed computer is available during the day. The limited access provides about 2 to 3 hours per day of quick turnaround time for the execution of engineering problems. During the quick turnaround time a job not exceeding 10 minutes may be taken to the computer for compiling and execution. The engineer is called when the job has been completed, and, if necessary, he may revise the program or data and resubmit the program for another run. When execution time becomes excessively long on the medium-speed computer, the program may be sent to a high-speed computer at a nearby university. As the program must be transported to the university, compiled, and executed and returned to the laboratory, only about one run per day is possible. However, if the program is of high enough priority, the engineer may accompany the program to the university where several runs per day are possible.

Using the three computers provides service for all types of programs - from small one-shot throw-away programs to large mathematical models.

The three computers function well as complementary units. The time-sharing terminal is frequently used to debug small routines for larger programs. The small routines are then revised for compatibility with the larger programs. The complete program is then executed on one of the larger computers. This process often considerably decreases the effort and time involved in debugging large programs. The routine for the time-sharing terminal may be written in FØRTRAN in which case very little change in the routine is necessary. However, the routine may also be written in EXTENDED BASIC. If EXTENDED BASIC is used, the program may be readily rewritten as a subroutine using FØRTRAN IV. The entire program may then be compiled and executed using the medium-speed computer or the high-speed computer.

An example of use of the computer for this type of problem is shown in Appendix I. The problem involves backfilling wells with sand-gravel mixtures. As the wells may be of different depths and the water surfaces at different elevations, it was necessary to write an acceleration-deceleration subroutine for the various size particles near the water surface. If the water surface in the well stands near the elevation that particles are being added, the particles will enter the water and accelerate to their terminal velocity. However, if the water surface stands several feet below the elevation at which particles are being introduced, the particles will accelerate to a relatively high velocity before entering the water and will then decelerate to their terminal velocity. In other words, the particles will travel different distances after entering the water in a given time interval depending on conditions under which the well is tested. In Appendix I the routine is presented written first in EXTENDED BASIC and then converted to FØRTRAN IV. It may be noticed that these two programs are not identical. However, an examination of the two programs will readily reveal their similarities.

Subroutine Documentation System

As computer use progressed, it was realized that many engineers were writing programs to solve similar routine problems. These small programs were somewhat specialized and dealt primarily with operations in the Hydraulic Laboratory. Often the programs were written for only a special range of conditions which were of interest to the engineer, and as a result the programs were not generalized.

Sufficient personnel were not available to develop a completely generalized and documented catalog of programs for the Laboratory; however, a limited subroutine documentation system was started. The purpose of the system was to maintain small programs and/or subroutines with sufficient documentation to allow them to be located readily and used in future work. The objectives of the documentation system were to:

1. Provide an index which would permit routines to be located and updated readily.

2. Furnish descriptive material in a standard format. The descriptive material would include the program purpose; information on how to call the program; input and output data required; and special information, such as tapes used and error reports for diagnosing.

3. Provide an estimate of the approximate necessary storage.

4. Furnish a program listing including clarification comments when necessary.

5. Provide mathematical ideas used in developments and examples of output data if necessary for clarification.

6. Provide name of programmer and date program was written.

To implement the system two short computer programs were written and a three-level system was established. The first level of the system consists of an index which lists the various routines available. The second level consists of the descriptive documentation, and the third level includes a program listing and necessary clarifying information.

An example of a page from the index with some typical headings is shown in Appendix II. Routines are entered under subject headings, examples of which would be air models, closed channel hydraulics, and mathematical subroutines. Information listed under the headings include the routine name, a brief description, and the pages on which additional information regarding the routine may be located in the next two levels. The index may be updated whenever a sufficient number of programs are added to or subtracted from the system.

The second level consists of the descriptive documentation of the program. An example of this level is shown in Appendix II. Standard headings are provided in the program which was written for this level and they may be used as desired. Standard messages under some headings are provided and may be used when they have meaning. Examples of standard messages which are useful for most programs are: Peripheral device indicators used for input and output of data; language used (usually FØRTRAN IV); and status of the program (developmental or production). Other information provided by the descriptive documentation includes: The programmer's name and date the program was written, purpose, calling sequence, input required, output data from the program, storage required, and how the program can be changed for minor variations. This information, furnished by the programmer, is then placed in standard format and listed by the computer. Sufficient descriptive documentation is encouraged. If it is desired to add subsequent information to the documentation, a new computer run may be made to readily add the updated information. Information in the descriptive documentation section is entered and page numbers are added as it is received in chronological order. The information is readily located by use of the index.

The third level of the system includes the subroutine listing, and any necessary clarifying information. Clarifying information may include comments about the program, mathematical information used in writing the program, graphical data to explain the program, error information, examples of output or input data, or other information the programmer believes should be included. An example of the third level with some clarifying information is shown in Appendix II. Routines are also entered and page numbers added as they are received in chronological order. Subroutine listings may be readily located by use of the index.

Use of Subroutine Documentation System

When the engineer desires to write a new program, he may look through the index section under subjects related to the problem. If he finds programs he thinks would be useful to him, he proceeds to the descriptive documentation and/or program sections. He may at this point accept or reject the subroutine. Acceptance may be by actual use of the subroutine with his new program, or in use of the subroutine as a base for developing a new routine which more nearly meets the needs of the problem.

When programming, engineers often recognize a portion of the problem as one which may be encountered in the future. When this is recognized, they write that portion of the problem as a subroutine and enter it into the system. The complete program may also be entered into the system if the engineer thinks that others may have a use for it.

Future Plans

A study is now underway in the Laboratory to determine the feasibility of using a computer complex to provide for the collection, evaluation, recording, and formating of data from models, as well as furnishing specified output signals for controlling Laboratory equipment.

Data collection would be controllable by programmed instructions entered through a control console and/or a remote terminal. The computer complex would be capable of accepting electrical input signals in both digital and analog form. Analog input signals would be of both low-level (millivolts) and high-level (volt) types. The input signals would originate both from voltage output devices and contacts, and from manually operated input devices located at the model.

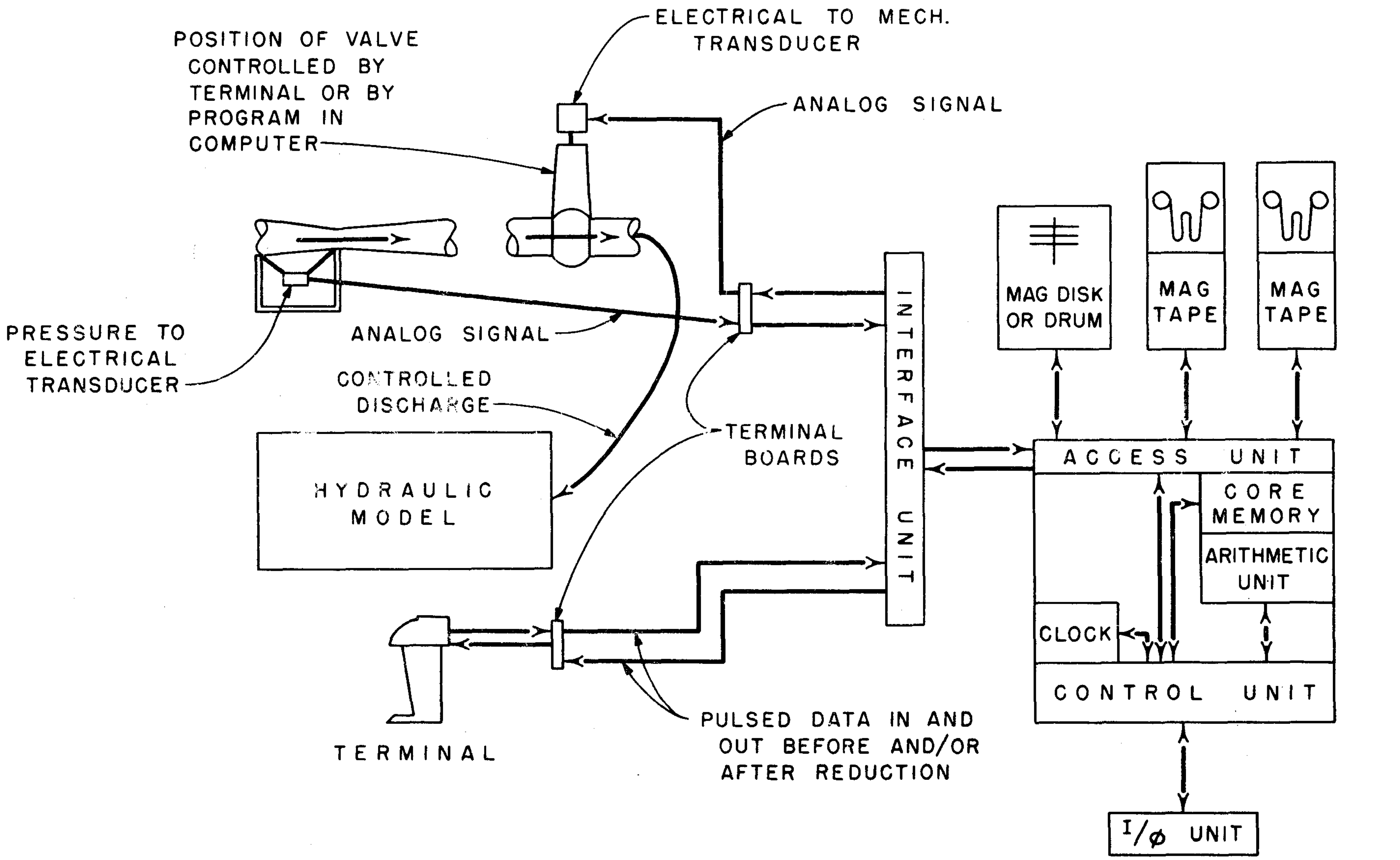
Input data would be processed and formated by a previously compiled program, and the compiled program and/or a remote terminal would originate specified output signals for controlling various equipment in the laboratory. Three types of output signals would be provided. These would be: (1) pulsed signals for typing at a remote terminal and programmable pulsed data for equipment operation; (2) on-off lines for controlling relays; and (3) programmable d-c analog signals. The necessary digital-to-analog, analog-to-digital, and other conversions would be included in the system.

A simplified concept of how a system of this type may be used to provide a varying programmable discharge through a hydraulic model is shown in Figure 1.

Input/Output terminal boards with the desired number of input-output plug-type receptacles would be placed at convenient locations throughout the laboratory. If desired, multiplexers (switching networks that perform the function of sequentially switching various signal sources into one common channel) may be placed external to the terminal board. Although this would increase the potential of the system, it would be necessary for the computer program to take into account the varying signal sources.

Both read/write magnetic tape drives, with formats compatible with other computers, and a large auxiliary memory unit of drum or disk type would be included in the system.

Software would include a control-oriented and problem-oriented language similar to FØRTRAN. The system could be used to compile and execute small programs (similar to those now being executed on the time-sharing system) when the data acquisition and control functions were not being used.



BASIC CONCEPT OF COMPUTER USE IN MODEL STUDY

FIGURE 1

APPENDIX I

DRAG

```
10 DIM Z(10), Y(10,5), V(10), F(10)
20 LET A = 1.94/2.39E-5
30 FØR J = 1 TØ 10
40 READ Z(J)
50 LET D = Z(J)/25.4
60 IF D >0.054 THEN 100
70 IF D >0.0054 THEN 120
80 LET V(J) = 43895 * D↑2
90 GØ TØ 130
100 LET V(J) = 55.943 * SQR(D)
110 GØ TØ 130
120 LET V(J) = 239 * D
130 LET V(J) = V(J)/30.48
140 PRINT "SIZE=";Z(J);"MM", "  ", "VEL=";V(J);"FT/SEC"
150 NEXT J
160 READ F1,D6
170 LET V1 = SQR(64.4*F1)
180 PRINT "INITIAL VELØCITY=";V1; "FT/SEC"
190 FØR J = 1 TØ 10
200 LET F(J) =0
210 LET D = Z(J) /25.4
220 LET D1 = D/12
230 LET A1 = 0.785398 * D1↑2
240 LET V2 = 0.523598 * D1↑3
250 LET W = V2 * 102.96
260 LET W1 = V2*165.36
270 LET P = V1
280 FØR K = 1 TØ 5
290 LET Y (J,K) = 0
300 LET T3= 0
310 FØR M = 1 TØ 100
320 LET T = ABS((P-V(J)) * 100/V(J))
330 IF T < 5 THEN 590
340 LET R1 = A * P * D1
350 IF R1>157 THEN 440
360 LET E = -1
370 LET G = LØG(27.3)
380 IF R1 < 1.5 THEN 410
390 LET E = -0.5
400 LET G = LØG(22.1)
410 LET C = E * LØG(R1) +G
420 LET C = EXP(C)
430 GØ TØ 450
440 LET C = 1.7505
450 LET H = 0.97 * C * P↑2 * A1
460 LET F2 = W-H
470 LET X = F2 * 32.2/W1
480 LET T4 = ABS(0.5 * (P-V(J))/X)
490 IF ( T3+T4) > D6 THEN 510
500 GØ TØ 520
```

DRAG CONTINUED

```
510 LET T4 = D6-T3
520 LET V6 = X * T4
530 LET S1 = P * T4 + 0.5*X*T4↑2
540 LET P = P + V6
550 LET Y(J,K) = Y(J,K) + S1
560 LET T = T3 + T4
570 IF T3 > = D6 THEN 630
580 NEXTM
590 LET T7 = D6 -T3
600 LET D7 = V(J) * T7
610 LET Y(J,K) = Y(J,K) + D7
620 LET F(J) = F(J) + Y(J,K)
630 PRINT "J=";J, "K=";K, "Y=";Y(J,K), "F=";F(J)
640 NEXT K
650 NEXT J
660 DATA 8.3,7,5.8,4.76,4.1,3.5,2.9,2.4,1.9,1.2
670 DATA 1,1
9999 END
```

```
      SUBROUTINE ACCDEC
      DIMENSION SIZE(20), SVEL(20), DS(20,5), TDFV(20), COUNTZ(20)
      COMMON SIZE, DS, SVEL, V, DT, TDFV, COUNT, COUNTZ, EFFSIZ
      A = 1.94 /2.39E-5
      DO 10 J = 1, 20
      TDFV(J) =0.0
      D = SIZE(J) /25.4
      D1 = D /12.0
      A1 = 0.785398 * D1**2
      V2 = 0.523598 * D1**3
      W = V2 * 102.96
      W1= V2 * 165.36
      PVEL = V
      DO 11 K = 1, 5
      DS(J,K) = 0.0
      TSUM = 0.0
      DO 12 M = 1, 100
      TEST = ABS((PVEL-SVEL(J))*100.0/ SVEL(J))
      IF (TEST .LT. 5.0) GO TO 15
      R1 = A * PVEL * D1
      IF (R1 .GT. 157.0) GO TO 16
      E = -1.0
      G = ALOG(27.3)
      IF (R1 .LT. 1.5) GO TO 17
      E = -0.5
      G = ALOG(22.1)
   17 C = E * ALOG(R1) + G
      C = EXP(C)
      GO TO 18
   16 C = 1.7505
   18 H = 0.97 * C * PVEL**2 * A1
      F = W - H
      X = F * 32.2 / W1
      T = ABS(0.5* (PVEL -SVEL(J))/ X)
      IF((TSUM +T) .GT. DT) T = DT - TSUM
      V6 = X * T
      S1 = PVEL * T  + 0.5 * X * T**2
      PVEL = PVEL + V6
      DS(J,K) = DS(J,K) + S1
      TSUM = TSUM + T
      IF (TSUM .GE. DT) GO TO 11
   12 CONTINUE
   15 TDIFF = DT - TSUM
      DISTIN = SVEL(J) * TDIFF
      DS(J,K) = DS(J,K) + DISTIN
      TDFV(J) = TDFV(J) + DS(J,K)
   11 CONTINUE
   10 CONTINUE
      RETURN
      END
```

APPENDIX II

AIR MODELS

CLOSED CHANNEL HYDRAULICS

DATA HANDLING ROUTINES

DOCUMENTATION PROGRAMS

DESCRIPTIVE DOCUMENTATION LEVEL

SUBROUTINE AIRWGT

PURPOSE

TO DETERMINE THE WEIGHT OF A CUBIC FOOT OF AIR AT CERTAIN TEMPERATURE, PRESSURE AND RELATIVE HUMIDITY CONDITIONS.

CALL BY -

CALL AIRWGT (T, HG, PRH, W)

NECESSARY INPUT VARIABLES

T, HG, PRH

ARRAY LIMITS

NONE

DESCRIPTIONS

T = TEMPERATURE IN DEGREES FAHRENHEIT
HG = PRESSURE IN INCHES OF HG (28.5 INCHES IS USED AS BASE)
PRH = PERCENT RELATIVE HUMIDITY

VARIABLE TYPES

* ALL VARIABLES ARE STANDARD REAL AND INTEGRS.

OUTPUT VARIABLES

W

ARRAYS

NONE

DESCRIPTIONS

W = WEIGHT OF ONE CUBIC FT OF AIR IN LBS

VARIABLE TYPES

* ALL VARIABLES ARE STANDARD REAL AND INTEGRS.

STORAGE REQUIRED

STORAGE REQUIRED IS ABOUT 144 WORDS (48 BITS)

TAPES USED

TAPE 3 MAY BE USED FOR WRITE IF TEMPERATURE IS OUT OF LIMITS

WRITTEN BY	DATE
PHILLIP F. ENGER	2/16/67

SUBROUTINE AIRWG!

ADDITIONAL INFORMATION

PROGRAM MAY EASILY BE CHANGED TO USE A PRESSURE READING IN PLACE OF INCHES OF HG.

* END DOCUMENTATION

SUBROUTINE LEVEL

15.0A 02/17/67

```
      SUBROUTINE ATRWGT (T, HG, PRH, W )
      IF((T .LT. 32.0) .OR. (T .GT. 105.0)) WRITE (3, 1) T
    1 FORMAT (1H1,5X,  62H YOUR TEMPERATURE IS BEYOND THE RANGE OF THIS
     1PROGRAM, IT IS =,F8.2 )
      IF (T .GT. 55.0) GO TO 2
      W =(-0.0001513)*T + 0.08172
      GO TO 3
    2 IF (T .GT. 85.0) GO TO 4
      W = (-0.000134)*T + 0.08077
      GO TO 3
    4 W = (-0.000122)*T + 0.07975
    3 TM = HG - 28.5
      DELW =(-0.0000049166)*T + 0.002851
      ADD = TM * DELW
      W = W + ADD
      TM = PRH / 10.0
      IF (T .GT. 70.0) GO TO 5
      DLNW = (0.034394)*T -11.9746
      DELW = EXP(DLNW)
      SUB = TM * DELW
      W = W - SUB
      GO TO 6
    5 DLNW = (0.029966)*T - 11.6646
      DELW = EXP(DLNW)
      SUB = TM * DELW
      W = W - SUB
    6 RETURN
      END
```

The system by which the weight of air is presently being calculated is essentially:

$$W = f_1(T) + (HG - 28.5)\, f_2(T) - \left(\frac{PRH}{10}\right) f_3(T)$$

W = weight of air in lbs

$f_1(T)$ = if $T \leq 55°$ F

$f_1(T) = -0.0001513\ T + 0.08172\ (32 + 55)$

if $55°$ F $< T < 85°$ F

$f_1(T) = -0.00134\ T + 0.08077\ (55 + 85)$

if $85°$ F $< T < 105°$ F

$f_1(T) = -0.000122\ T + 0.07975$

HG = inches of mercury

$f_2(T) = 4.9166 \times 10^{-6}\ T + 0.002851$

PRH = percent relative humidity

$f_3(T)$ = if $T \leq 70°$ F

$f_3(T) = 0.034394\ T - 11.9746$

if $70°$ F $< T$

$f_3(T) = 0.029966 - 11.6646$