Technical Report TR-65-24   October, 1965

Subroutines for Digital Computation of Chemical Engineering Problems*

Part I

by

Chan M. Park

* This work was supported by National Aeronautics and Space Administration Grant NSG-398 to the Computer Science Center, University of Maryland, College Park, Maryland.

CONTENTS

I. INTRODUCTION

This report contains a collection of subroutine programs developed for chemical engineers and chemical engineering students to aid in the solution of their problems. Rather than developing all-purpose general routines with many options which often make them difficult to use, these subroutines were developed with emphasis on practical applicability in the chemical engineering field and organized to make them usable with a minimum of effort.

The subroutines described in this paper are basic and are frequently required in the reduction of experimental data to usable form as well as in the solution of computer-oriented chemical engineering problems. Both FORTRAN II and FORTRAN IV versions are provided so that a user can choose according to his control program. Control programs as well as numerical results are also included for each subroutine to illustrate its usage. Since there are only small differences between the programs in FORTRAN II and FORTRAN IV, only FORTRAN II program listings are given in the Appendices at the end of this report.

Some of the restrictions involved in the subroutines, such as dimension statements, may be easily removed by modifying the source deck. Both source and binary decks are available from the author.

Additional routines dealing with mathematical equations such as ordinary and partial differential equations, as well as some specific programs for solving complex chemical engineering problems such as kinetics, diffusion and unsteady-state transport phenomena, are presently being developed and will be the subject of Part II of this report.

A word of caution: The maximum number of elements in a two- or three-array variable specified in a dimension statement of a control program should be the same as that in a subroutine which is called by the control program. The user must therefore be careful when working with two- or three-dimensional variables.

II. DESCRIPTION OF ROUTINES

The routines are divided into two classes. The first class consists of non-mathematical routines which are useful in the treatment of chemical engineering problems. The second class consists of mathematical subroutines which are frequently employed in the solution of the problems. The theoretical and procedural aspects of the mathematical routines are described in standard works on numerical analysis such as those listed as references at the end of this report. Control programs have been used to illustrate practical applications of the subroutines in each case.

A. Non-Mathematical Routines

1) Unit Conversion Routine (Subroutine CONVRT)

Conversions of complex units from one scale to another are very often necessary in the solution of chemical engineering problems. For example, if a chemical engineer wants to use thermal conductivity data obtained by a physicist who often chooses Kcal/sec-cm-°C as its unit, he may want to convert the unit into Btu/hr-ft^2/in-°F. Although it is not difficult to convert one unit into another, usually it takes time and effort, and there is the possibility of introducing errors. This routine has been written to make it possible to perform conversions with greater speed and accuracy than by manual methods. The routine is especially effective if the entire problem is being solved by the computer, as will be the case in most instances. CONVRT has been developed in such a way that any additional units required can be easily added. Most of the practical units employed by a chemical engineer are already included in the routine.

The calling sequence of the routine is

```
CALL CONVRT (N, A, B, FACTOR)
```

where

N = a zero or an integer number

A = original unit

B = desired unit

FACTOR = conversion factor

There are two ways of introducing the units into the machine, one as Hollerith arguments of the subroutine and the other as input data to the machine. The arguments N, A, and B will have somewhat different features depending upon the above two cases, as shown in the following example.

Let us consider the conversion of KCAL/(SEC*CM*DEGC) into BTU/(HR*(FT**2/INCH)*DEGF).

a) If these units are introduced as Hollerith arguments, the calling sequence will have the form

CALL CONVRT (0, 19H^{b}KCAL/(SEC*CM*DEGC),

27H^{b}BTU/(HR*(FT**2/INCH)*DEGF), FACTOR)

Hence N = 0

A = 19H^{b}KCAL/(SEC*CM*DEGC)

B = 27H^{b}BTU/(HR*(FT**2/INCH)*DEGF)

(b means a blank space)

In this case, N should have the value zero because this is an index to tell the machine that the units are introduced as Hollerith arguments. The numbers (19 and 27) in front of the units are the numbers of characters (including blank spaces) following the letter H, which stands for Hollerith field.

Some additional examples:

i) conversion of lb/gal into Kg/m^3

CALL CONVRT (0, 7H LB/GAL, 10H KG/(M**3), FACTOR)

ii) conversion of centipoise into lb/ft-sec

CALL CONVRT (0, 3H CP, 12H LB/(FT*SEC), FACTOR)

b) If the units are introduced as input data to the machine, the calling sequence is

CALL CONVRT (N, UNIT1, UNIT2, FACTOR)

where UNIT 1 and UNIT2 correspond to A and B, respectively. In this case UNIT1 and UNIT2 are variables with one-dimensional subscripts, and the following statements should be included in the control program before the subroutine CONVRT is called. (The following example is in FORTRAN II.)

```
      DIMENSION UNIT1(24), UNIT2(24)
      N = 12 (or 24, see below)
      READ INPUT TAPE 5, 1,(UNIT1(I), I = 1,N)
      READ INPUT TAPE 5, 1, (UNIT2(I), I = 1,N)
    1 FORMAT(12A6)
            ......................
            ......................
      CALL CONVRT (N, UNIT1, UNIT2, FACTOR)
            ......................
            ......................
      END
*     DATA
      KCAL/(SEC*CM*DEGC)
      BTU/(HR*(FT**2/INCH)*DEGF)
```

The units, both original and desired, are punched on the separate cards as input data. If any one of the units has more than 72 characters (which means that one card is not enough for that unit), the value of N should be 24 and two cards should be given as data for each unit. If one of the units has less than 72 characters while the other has more than 72, then the former should have a blank card immediately following the data card to make two data cards. If a unit requires more than two cards, the dimension statement should be changed to increase the storage, and N may then be increased to read in more than two cards.

Notes:

a) Whenever feasible, conventional abbreviations such as BTU, KG, etc. were used in the program. However, users are urged to check the list of abbreviations provided in front of the subroutine CONVRT before using it.

b) The formation of a complex unit is very simple; it is similar to writing an arithmetic expression in FORTRAN. Some rules are:

(i) Parentheses may be used anywhere in the complex unit, so as to perform the computation of that part of the unit first, except in front of an exponentiation operator (**). This is not a serious restriction since in practice an exponential operation is performed on a simple unit (e.g. cm^2, ft^3, sec^2, etc.) rather than on a compound unit in the formation of a complex unit.

(ii) The multiplication symbol (*) is used between simple units (e.g. SEC*CM*DEGC for sec-cm-°C) to separate them. (Actually, the simple units are multiplied to form the complex unit, even though the (-) symbol is used in the conventional expression.)

(iii) Squares and cubes are expressed using the exponentiation operator (e.g. CM**2 for cm^2, and FT**3 for cubic foot).

(iv) The exponent should be an integer whose value is less than 8. (Only 2 and 3 are used in most cases.)

(v) No addition (+) or subtraction (-) symbols may be used.

c) It should be noted that this routine computes a conversion factor from one unit to another. For example, it will give 1.8 as the conversion factor from °C into °F. However, the routine will not convert a temperature of 100°C into Fahrenheit temperature unless the appropriate equation is provided after the routine CONVRT is called. In this case the appropriate equation is:

```
FTEMP = 100. * FACTOR + 32.
```

d) When an error is detected in the formation of a complex unit, an error message will be printed out, and the computation will stop.

e) There are ten subprograms associated with subroutine CONVRT. Their names and functions are:

SEPCOM ......Called by CONVRT. Separate a complex unit into simple units and operators and compute a characteristic number for the complex unit.

GETNO ....Called by SEPCOM. Search the table of simple units and get a characteristic number for a simple unit separated from the complex unit.

ERROR1 through ERROR7 ......Called by SEPCOM. Write error messages if there are any errors in the formation of a complex unit.

WRITE1 ......Called by GETNO. Write an error message when a given simple unit is not found in the table of simple units.

<u>Control Program</u> (<u>CCNVRT</u>)

Two control programs are given in Appendix 2 to illustrate the two cases described above. In the first case, the factor for a conversion from Btu/sec into ft-lb/sec is computed. The second control program finds the conversion factor for a conversion from centipoise into lb/(ft-hr).

2) <u>Log-log Plot Routine</u> (Subroutine <u>LNPLOT</u>)

A plot of experimental data or computed results on a log-log scale is often desired by a chemical engineer. A preliminary analysis usually does not require a very fine curve, yet it takes considerable time and effort if large amounts of data are being treated. This log-log plot routine was written to free the researcher from the drudgery of a manual plot.

The calling sequence is

CALL LINPLOT (N, X, Y, SYMBOL)

where

N = number of points
X = array of abscissa values
Y = array of ordinate values
SYMBOL = a character to be used in the plot

The points do not have to be in any order. (In other words, the X-coordinates of the points need not be in an ascending or a descending order.)

SYMBOL is a character which is defined before the routine is called. The most frequently used are "*", and "+", although any character such as "A", "1", etc. may be used. The definition of SYMBOL will have the form

SYMBOL = 1H*
SYMBOL = 1H+   etc.

in a FORTRAN II program. An alternative calling sequence in FORTRAN II is to define the symbol as an argument of the subroutine, e.g. CALL LNPLOT (N, X, Y, 1H*). In a FORTRAN IV program, SYMBOL will be defined with a DATA statement.

<u>Notes</u>:

a) Limitation: $N \leq 200$
b) Dimensioned variables: X(200), Y(200)

<u>Control Program</u> (<u>CLNPLT</u>)

The sample problem considered here is that of plotting a curve of dispersion group vs. Reynolds number in a straight pipe, as compiled by Levenspiel (Industrial and Engineering Chemistry vol. 50, p. 343, 1958).

The equation used to compute the dispersion group for a Reynolds number less than 20,000 was obtained by the author by applying the least-squares fit method to the curve presented by Levenspiel.

$$\ln\left(\frac{D}{u\, d_t}\right) = 0.63082532\,(\ln(N_{Re}))^2 - 12.5531745\,(\ln(N_{Re})) + 61.25958824 \qquad \text{for } N_{RE} \leq 20{,}000$$

The dispersion group for a Reynolds number greater than 20,000 was computed by use of the theoretical equation derived by Taylor.

$$\frac{D}{u\, d_t} = 3.57 \sqrt{0.0014 + 0.09\left(\frac{1}{N_{Re}}\right)^{0.27}}$$

for $N_{Re} > 20{,}000$

B. Mathematical Routines

1) Lagrangian Interpolation Routine (Subroutine LAGINT)

This routine may be used in the evaluation of Y-coordinate values for given values on X-axis by use of the Lagrangian interpolation method.

The calling sequence is

CALL LAGINT (N, X, Y, NV, XV, YV)

where

N = number of points in a given table

X, Y = coordinates of the given points

NV = number of points to be approximated by the interpolation

XV = X-coordinates of points for which the Y-coordinates are sought

YV = Y-coordinates obtained from interpolations

The X-coordinates in the given table should be in ascending order.

Notes:

a) Limitations: N ≦ 500 and NV ≦ 500
b) Dimensioned variables: X(500), Y(500), XV(500), YV(500)
c) If XV(I) is less than the smallest or greater than the largest value of X, a message will be printed out and no interpolation will be attempted for that particular XV.

Control Program (CLAGIN)

The control program is written to illustrate a computation of mass rates for given pressure drops in an orifice manometer.

Suppose that the following table is obtained from a calibration experiment of an orifice meter.

Pressure Drop (in.)	1.0	2.0	3.0	4.0	5.0	6.0	7.0
Mass Rate (lb/min)	1.32	1.87	2.29	2.64	2.96	3.23	3.49

It is desired to compute the mass rates for runs carried out with the pressure drop as follows:

Run No.	1	2	3	4
Pressure Drop (in.)	1.5	3.5	4.0	5.5

The control program listing in Appendix 2 shows how subroutine LAGINT is used.

2) Gauss Elimination Routine (Subroutine GAUSS)

This routine will solve linear simultaneous equations by a Gauss elimination method. It is also employed as a sub-program of other subroutines such as least-squares fit and matrix inversion routines.

The calling sequence is

CALL GAUSS (M, A, X)

where

M = number of unknowns

A = coefficients of unknowns

X = unknowns

Notes:

a) Limitation: $M \leqq 50$
b) Dimensioned variables: A(50,51), X(50)
c) If the matrix of coefficients is singular or not solvable because of zero coefficients generated in the process of elimination, a message is printed out and computation stops.
d) The original matrix of coefficients is not altered during the process.

Control Program (CGAUSS)

The following equations show component balances of acetone, water and air for an acetone recovery system.

acetone	$F(0.0295) = P(0.99) + W(0.05) + A(0)$
water	$120 = P(0.01) + W(0.95) + A(0)$
air	$F(0.9705) = P(0) + W(0) + A$
(over-all	$F + 120 = F + W + A)$

where

F = rate of feed to absorber, lb/hr

A = air leaving absorber, lb/hr

P = product, lb/hr

W = waste, lb/hr

the control program computes P, W, and A for a given F. The numerical computation was done with F = 50,000lb/hr.

3) Least-Squares Fit Routine (Subroutine LEASQU)

The object of this routine is to find an Nth degree polynomial which will best fit a set of data using the least-squares criterion. The polynomial is expressed in the form

$$Y = C_1 X^N + C_2 X^{N-1} + C_3 X^{N-2} + \ldots\ldots + C_N X + C_{N+1}$$

The routine will compute C_1, C_2, ..., C_{N+1} provided that a set of X and Y data is given.

The calling sequence is

```
CALL LEASQU (N, X, Y, MDEGRE, C)
```

where

N = number of points given
X, Y = coordinates of the given points
MDEGRE = highest degree desired for a fit (e.g. MDEGRE = 1 for a straight line)
C = coefficients of the polynomial

Notes:

a) Limitation: MDEGRE ≦ 50
b) Dimensioned variables: X(200), C(50) , Y(200)
c) Subroutine GAUSS is called by this routine. Therefore, a source or binary deck of GAUSS should be submitted together with this routine.

Control Program (CLEASQ)

In an experiment, a flow meter is being calibrated. The following table has been obtained.

Readings (GPH)	20	30	40	50	60	70	80	90	100	120
True Rate (GPH)	19.8	29.7	39.7	49.6	60.2	70.3	80.2	90.0	102.5	120.5

It is desired to obtain a straight line correlating the flow meter reading with the true rate. The equation for this line is to be found by the method of least squares.

4) Matrix Multiplication Routine (Subroutine MATMPY)

This routine computes the product of two square matrices.

The calling sequence is

CALL MATMPY (N, A, B, C)

where

N = number of elements in a row or column
A = matrix A
B = matrix B
C = matrix C, the product of A and B

Notes:

a) Limitation: N≦50
b) Dimensioned variables: A(50,51), B(50,51), C(50,51)

Control Program (CMTMPY)

In solution of an engineering problem, the following two matrices A and B are obtained.

$$A = \begin{vmatrix} 8.467 & 5.137 & 3.141 & 2.063 \\ 5.137 & 6.421 & 2.617 & 2.003 \\ 3.141 & 2.617 & 4.128 & 1.628 \\ 2.063 & 2.003 & 1.628 & 3.446 \end{vmatrix}$$

$$B = \begin{vmatrix} 0.25087 & -0.16214 & -0.08116 & -0.01760 \\ -0.16214 & 0.33074 & -0.05994 & -0.06686 \\ -0.08116 & -0.05994 & 0.37988 & -0.09604 \\ -0.01760 & -0.06686 & -0.09604 & 0.38497 \end{vmatrix}$$

It is desired to find out whether matrix B is the inverse of matrix A. An easy way of solving this problem is to take the product of A and B and to check if the resulting matrix is the identity matrix. This control program will find the product of A and B.

5) Matrix Inversion Routine (Subroutine MATINV)

This subroutine inverts a square matrix by the Gauss elimination method.

The calling sequence is

CALL MATINV (M, A, B)

where

M = number of elements in a row or column
A = original matrix
B = inverse matrix of A

Notes:

a) Limitation: $M \leq 50$
b) Dimensioned variables: A(50,51), B(50,51)
c) Subroutine GAUSS is a subprogram of this routine.

Control Program (CMTINV)

One way of solving a system of linear equations is by use of the inverse matrix of the coefficients.

i.e.: $A\vec{x} = \vec{b}$;

Multiplying by A^{-1}, $A^{-1}A\vec{x} = A^{-1}\vec{b}$.

Since $A^{-1}A = I$,

there results $\vec{x} = A^{-1}\vec{b}$

This method is very effective when $\vec{b}$ changes with $\vec{x}$ while A is kept constant, because $\vec{x}$ can be found very easily for different sets of $\vec{b}$ once A^{-1} is computed.

In the illustration of this control program, the following matrix A is inverted.

$$A = \begin{vmatrix} 8.467 & 5.137 & 3.141 & 2.063 \\ 5.137 & 6.421 & 2.617 & 2.003 \\ 3.141 & 2.617 & 4.128 & 1.628 \\ 2.063 & 2.003 & 1.628 & 3.446 \end{vmatrix}$$

6) Numerical Integration Routine by Simpson's Method (Subroutine SIMPSN)

This routine finds the area under a curve given in the form of numerical points by Simpson's integration method. In this program Simpson's method has been modified to make it possible to work with both even and odd numbers of increments.

The calling sequence is

CALL SIMPSN (N, Y, H, A)

where

N = number of points (= number of increments + 1)
Y = Y-coordinates of the points
H = size of an increment on the X-axis
A = area obtained from the integration

Notes:

a) Limitation: $N \leqq 200$
b) Dimensioned variable: Y(200)
c) The points should be equally spaced on the X-axis.

Control Program (CSMPSN)

In a dispersion experiment using the trace injection technique, Levenspiel and Smith obtained the following results (Chem. Engr. Sci. vol. 6, p. 227, 1957):

Time (sec)	Conc. (arbitrary unit)
0.0	0.0
2.0	11.0
4.0	53.0
6.0	64.0
8.0	58.0
10.0	48.0
12.0	39.0
14.0	29.0
16.0	22.0
18.0	16.0
20.0	11.0
22.0	9.0
24.0	7.0
26.0	5.0
28.0	4.0
30.0	2.0
32.0	2.0
34.0	2.0
36.0	1.0
38.0	1.0
40.0	1.0
42.0	1.0

In order to normalize the concentration data, it is desired to compute the area under the concentration curve. This control program also shows how the dimensionless variance can be calculated in the dispersion experiment.

7) Modified Runge-Kutta Method Routine (Subroutine RUNMOD)

Subroutine RUNMOD solves an ordinary differential equation (first order) by a Runge-Kutta method as modified by Gill. The ordinary differential equation has the form $y' = F(x, y)$ with an initial condition $y(x_o) = y_o$.

The calling sequence is

```
CALL RUNMOD (N, X, Y, H)
```

where

N = number of points
X = X-coordinates of points (equally spaced)
Y = Y-coordinates of points computed
H = size of an increment on the X-axis

The function $y' = F(x, y)$ should be given in the following form (Example in FORTRAN II):

```
FUNCTION FUNCT(X, Y)
FUNCT = α(X) + β(Y) + γ(X*Y) + δ*
RETURN
END
```

*α, β, γ, δ are constants, functions such as sine, cosine, log, etc. or combinations of them. For example, if $y' = \sin x + 2\cos 3y$ then

```
FUNCT = SINF(X) + 2.  * COSF(3. * Y)
```

Notes:

a) Limitation: $N \leq 500$
b) Dimensioned variables: X(500), Y(500)
c) Since X's are equally spaced, it is necessary to give only the value of X(1). The others will be computed by the routine.
d) The initial value Y(1) should be defined before the routine is called.

Control Program (CRUNMO)

A mixing tank problem will be considered as an example. In a large tank equipped with a stirrer, salt solution is introduced at a rate of 5 gal/min. The concentration of salt is 2lb/gal. Initially, the tank contained 100 gal of pure water. Assuming that the system is a perfectly mixing system and that the solution leaves the tank at the bottom at a rate of 1 gal/min, it is desired to find the concentration of salt in the tank as a function of time. In this case, the mathematical equation to be solved is

$$\frac{dc}{dt} = \frac{10}{100 + 4t} - \frac{5c}{100 + 4t}$$

with an initial condition c = 0 when t = 0.

APPENDIX 1. LISTING OF SUBROUTINE PROGRAMS

Disclaimer

ABBREVIATIONS USED FOR UNITS IN SUBROUTINE CONVRT

ACRE	ACRE
ANGSTROM	ANGST
ATMOSPHERE	ATM
BAR	BAR
BRITISH THERMAL UNIT	BTU
BUSHEL	BUSH
CALORIE (GRAM-)	CAL
CENTIMETER	CM
CENTIPOISE	CP
DAY	DAY
DEGREE CENTIGRADE	DEGC
DEGREE FAHRENHEIT	DEGF
DYNE	DYNE
ERG	ERG
FOOT	FT
FEET OF WATER (PRESSURE, AT 15 DEGREES C)........	FTWAT
GALLON	GAL
GRAM	GR
GRAIN	GRAIN
GRAVITY	GRAVIT
HORSEPOWER..	HP
HOUR	HR
INCH	INCH
INCHES OF MERCURY (PRESSURE, AT 0 DEGREES C)	INHG
INCHES OF WATER (PRESSURE, AT 15 DEGREES C)	IWATER
JOULE	JOULE
KELVIN (ABS. TEMP.)	KELVIN
KILOCALORIE	KCAL
KILOGRAM	KG
KILOMETER	KM
KILOWATT	KW
POUND	LB
LITER	LITER
METER	M
MEGABAR	MEGBAR
MICRON	MICRON
MILE	MILE
MINUTE	MIN
MILIMETER	MM
MM OF MERCURY (PRESSURE)	MMHG
MILIMICRON	MMICR
METERS OF WATER (PRESSURE, AT 15 DEGREES C)	MWATER
OUNCE (WEIGHT)	OUNCE
OUNCE (VOLUME)	OZ
POISE	POISE
QUART (DRY)	QUARTD
QUART (LIQUID)	QUARTL
RANKIN (ABS. TEMP.)	RANKIN
ROD	ROD
SECOND	SEC
TON (METRIC)	TONM
TON-REFRIGERATION	TONREF
TON (SHORT)	TONS
WATT	WATT
YARD	YD

PROGRAMS

```
*       FAP
        LBL     CONVRT,2
        ENTRY   CONVRT
CONVRT  SXA     X1,1
        SXA     X2,2
        SXA     X4,4
        CLA*    1,4
        TZE     DIRECT
        PDC     ,1
        CLA     2,4
        STA     RELOC1
        CLA     3,4
        STA     RELOC2
        CAL     ALLONE
        SLW     S1,1
        SLW     S2,1
        AXT     0,1
        AXT     0,2
TEST    CAL     S1,1
        LAS     ALLONE
        TRA     *+2
        TRA     FINISH+1
RELOC1  CAL     **,2
        SLW     S1,1
RELOC2  CAL     **,2
        SLW     S2,1
        TXI     *+1,1,-1
        TXI     TEST,2,1
FINISH  PZE     S1
        CLA     *-1
        STA     ARG2
        TRA     *+2
        PZE     S2
        CLA     *-1
        STA     ARG3
        TRA     COMMON
DIRECT  CLA     2,4
        STA     ARG2
        CLA     3,4
        STA     ARG3
COMMON  CLA     4,4
        STA     ANSWER
        TSX     SEPCOM,4
ARG2    PZE     **
        PZE     VALUE1
        TSX     SEPCOM,4
ARG3    PZE     **
        PZE     VALUE2
        CLA     VALUE2
        FDP     VALUE1
ANSWER  STQ     **
X1      AXT     **,1
X2      AXT     **,2
X4      AXT     **,4
        TRA     5,4
VALUE1  BSS     1
VALUE2  BSS     1
S1      BSS     100
S2      BSS     100
ALLONE  OCT     777777777777
```

```
EXTERN  SEPCOM
END
```

```
*       FAP
        LBL     SEPCOM,2
        ENTRY   SEPCOM
SEPCOM  SXA     SAVE1,1
        SXA     SAVE2,2
        SXA     SAVE4,4
        AXT     100,1
        STZ     S+100,1
        TIX     *-1,1,1
        CLA     1,4
        STA     BIGLP
        CLA     2,4
        STA     ANSWER
        AXT     -30,1
        AXT     0,2
        ZAC
BIGLP   LDQ     **
SMLP    LGL     6
        LAS     BLANK1
        TRA     *+2
        TRA     BRNCH2
        LAS     LEFTP
        TRA     CHECKM
        TRA     *+2
        TRA     CHECKM
        SLW     S,2
        TXI     BRNCH2,2,-1
CHECKM  LAS     MULTPL
        TRA     CHECKD
        TRA     *+2
        TRA     CHECKD
        TXI     *+1,2,-1
        SLW     S,2
        CAL     S-1,2
        TZE     *+2
        TRA     M
        CAL     S-2,2
        LAS     MULTPL
        CALL    ERROR1
        TRA     *+2
        CALL    ERROR1
        TXI     *+1,2,1
        CAL     EXPNT
        SLW     S-1,2
        TRA     BRNCH2
M       TXI     BRNCH1,2,-1
CHECKD  LAS     DIVIDE
        TRA     CHECKR
        TRA     *+2
        TRA     CHECKR
        TXI     *+1,2,-1
        SLW     S,2
        TXI     BRNCH1,2,-1
CHECKR  LAS     RIGHTP
        TRA     CHARCT
        TRA     *+2
        TRA     CHARCT
        TXI     *+1,2,-1
        SLW     S,2
        TRA     BRNCH3
```

```
CHARCT ALS    0,1
       ORS    S,2
       TXI    BRNCH2,1,6
BRNCH1 CAL    S-2,2
       LAS    RIGHTP
       TRA    *+2
       TRA    BRNCH2
       ANA    MASK1
       TZE    NUMBER
       PXA    ,1
       PAC    ,1
       CAL    BLANK
       ARS    30,1
       ORS    S-2,2
       TRA    RESET1
BRNCH3 CAL    S-1,2
       LAS    RIGHTP
       TRA    *+2
       TRA    BRNCH2
       ANA    MASK1
       TZE    NUMBR2
       PXA    ,1
       PAC    ,1
       CAL    BLANK
       ARS    30,1
       ORS    S-1,2
       TRA    RESET1
NUMBR2 CAL    S-1,2
       ARS    30
       SLW    S-1,2
       TRA    RESET1
NUMBER CAL    S-2,2
       ARS    30
       SLW    S-2,2
RESET1 AXT    -30,1
BRNCH2 XCL
       TZE    *+4
       XCL
       CLM
       TRA    SMLP
       CAL    BIGLP
       ADD    =01
       STA    BIGLP
       STA    TEST2
TEST2  CAL    **
       LAS    =07777777777777
       TRA    BIGLP-1
       TRA    *+2
       TRA    BIGLP-1
       PXA    ,2
       STO    IR2
       CAL    S,2
       LAS    RIGHTP
       TRA    *+2
       TRA    RESET2
       ANA    MASK1
       TZE    NUMBR3
       PXA    ,1
```

```
        PAC     ,1
        CAL     BLANK
        ARS     30,1
        ORS     S,2
        TRA     RESET2
NUMBR3  CAL     S,2
        ARS     30
        SLW     S,2
RESET2  AXT     0,1
GETNOS  CAL     S,2
        ANA     MASK1
        TZE     REPEAT-1
        CAL     S,2
        SLW     A
        TSX     GETNO,4
        PZE     A
        PZE     B
        CLA     B
        STO     S,2
        TXI     *+1,1,1
REPEAT  TXL     CALC,2,0
        TXI     GETNOS,2,1
CALC    AXT     0,1
BIGLP2  CAL     S,1
        LAS     EXPNT
        TRA     *+2
        TRA     FIRST
        PXA     ,1
        SUB     IR2
        TZE     SMLP2
        TXI     BIGLP2,1,-1
SMLP2   AXT     0,1
        CAL     S,1
        LAS     RIGHTP
        TRA     *+2
        TRA     SECOND
        PXA     ,1
        SUB     IR2
        TZE     THIRD
        TXI     SMLP2+1,1,-1
FIRST   CLA     S-1,1
        STO     TEMP1
LOOP1   CLA     S+1,1
        SUB     =01
        STO     S+1,1
        TZE     *+5
        LDQ     S-1,1
        FMP     TEMP1
        STO     S-1,1
        TRA     LOOP1
        CAL     BLANK
        SLW     S,1
        SLW     S+1,1
        AXT     0,1
        TRA     BIGLP2
SECOND  TXI     *+1,1,1
        CAL     S,1
        LAS     LEFTP
```

```
        TRA     *+2
        TRA     LOOP2
        TIX     SECOND,1,0
        CALL    ERROR2
LOOP2   SXA     X1,1
        SXA     XX1,1
        CLA     S+1,1
        STO     TEMP2
        TXI     *+1,1,-1
        CAL     S+1,1
        LAS     BLANK
        TRA     *+2
        TRA     *-4
        LAS     MULTPL
        TRA     *+2
        TRA     PROD
        LAS     DIVIDE
        TRA     *+2
        TRA     DIVISN
        CAL     BLANK
        SLW     S+1,1
XX1     AXT     **,1
        SLW     S,1
        TRA     SMLP2
PROD    CAL     BLANK
        SLW     S+1,1
        TXI     *+1,1,-1
        CAL     S+1,1
        LAS     BLANK
        TRA     *+2
        TRA     PROD+2
        LDQ     S+1,1
        FMP     TEMP2
        STO     TEMP3
        CAL     BLANK
        SLW     S+1,1
X1      AXT     **,1
        CLA     TEMP3
        STO     S+1,1
        TRA     LOOP2
DIVISN  CAL     BLANK
        SLW     S+1,1
        TXI     *+1,1,-1
        CAL     S+1,1
        LAS     BLANK
        TRA     *+2
        TRA     DIVISN+2
        CLA     TEMP2
        FDP     S+1,1
        STQ     TEMP3
        CAL     BLANK
        SLW     S+1,1
        TRA     X1
THIRD   AXT     0,1
        CAL     S,1
        LAS     BLANK
        TRA     FINAL
        TRA     *+2
```

```
        TRA     FINAL
        PXA     ,1
        SUB     IR2
        TZE     *+2
        TXI     THIRD+1,1,-1
        CALL    ERROR4
FINAL   CLA     S,1
        STO     RESULT
        PXA     ,1
        SUB     IR2
        TZE     END
LOOP3   TXI     *+1,1,-1
        CAL     S,1
        LAS     BLANK
        TRA     *+2
        TRA     *+5
        LAS     LEFTP
        TRA     COMPUT
        CALL    ERROR3
        TRA     COMPUT
        PXA     ,1
        SUB     IR2
        TZE     END
        TRA     LOOP3
COMPUT  LAS     MULTPL
        TRA     DCHECK
        TRA     *+2
        TRA     DCHECK
        SXA     XA1,1
LOOP4   TXI     *+1,1,-1
        CAL     S,1
        LAS     BLANK
        TRA     MPY
        TRA     *+2
        TRA     MPY
        PXA     ,1
        SUB     IR2
        TZE     *+2
        TRA     LOOP4
        CALL    ERROR5
MPY     LAS     LEFTP
        TRA     *+2
        CALL    ERROR3
        LDQ     RESULT
        FMP     S,1
        STO     RESULT
        CAL     BLANK
        SLW     S,1
XA1     AXT     **,1
        SLW     S,1
        TRA     LOOP3
DCHECK  LAS     DIVIDE
        CALL    ERROR6
        TRA     *+2
        CALL    ERROR6
        SXA     XXA1,1
LOOP5   TXI     *+1,1,-1
        CAL     S,1
```

```
         LAS     BLANK
         TRA     DIV
         TRA     *+2
         TRA     DIV
         PXA     ,1
         SUB     IR2
         TZE     *+2
         TRA     LOOP5
         CALL    ERROR7
DIV      LAS     LEFTP
         TRA     *+2
         CALL    ERROR3
         CLA     RESULT
         FDP     S,1
         STQ     RESULT
         CAL     BLANK
         SLW     S,1
XXA1     AXT     **,1
         SLW     S,1
         TRA     LOOP3
END      CLA     RESULT
ANSWER   STO     **
SAVE1    AXT     **,1
SAVE2    AXT     **,2
SAVE4    AXT     **,4
         TRA     3,4
BLANK1   OCT     60
BLANK    OCT     606060606060
LEFTP    OCT     74
MULTPL   OCT     54
EXPNT    OCT     5454
DIVIDE   OCT     61
RIGHTP   OCT     34
MASK1    OCT     700000000000
A        BSS     1
B        BSS     1
S        BSS     100
IR2      BSS     1
TEMP1    BSS     1
TEMP2    BSS     1
TEMP3    BSS     1
RESULT   BSS     1
         EXTERN  GETNO
         EXTERN  ERROR1,ERROR2,ERROR3,ERROR4,ERROR5,ERROR6,ERROR7
         END
```

```
*       FAP
        LBL     GETNO,2
        ENTRY   GETNO
GETNO   SXA     X2,2
        SXA     X4,4
        SXA     X1,1
        CLA     2,4
        STA     NUMBER
        ZAC
        LDQ*    1,4
        LGL     6
        LAS     =O31
        TRA     MIDDLE
        TRA     *+1
        SUB     =O21
        TRA     SETIND
MIDDLE  LAS     =O51
        TRA     LAST
        TRA     *+1
        SUB     =O30
        TRA     SETIND
LAST    SUB     =O40
SETIND  PAC     ,1
        CLA     COUNT,1
        SUB     COUNT-1,1
        PAX     ,2
        CLA     COUNT,1
        PAC     ,1
        CAL*    1,4
COMP    LAS     TABLE-1,1
        TRA     *+2
        TRA     FOUND
        TIX     *+2,2,1
        CALL    WRITE1
        TXI     COMP,1,1
FOUND   CLA     FACTOR-1,1
NUMBER  STO     **
X2      AXT     **,2
X4      AXT     **,4
X1      AXT     **,1
        TRA     3,4
STARTA  SYN     *
TABLE   BCI     1,ACRE
        BCI     1,ANGST
        BCI     1,ATM
AN      EQU     *-STARTA
STARTB  SYN     *
        BCI     1,BAR
        BCI     1,BTU
        BCI     1,BUSH
BN      EQU     *-STARTA
STARTC  SYN     *
        BCI     1,CAL
        BCI     1,CM
        BCI     1,CP
CN      EQU     *-STARTA
STARTD  SYN     *
        BCI     1,DAY
        BCI     1,DEGC
        BCI     1,DEGF
```

```
        BCI     1,DYNE
DN      EQU     *-STARTA
STARTE  SYN     *
        BCI     1,ERG
EN      EQU     *-STARTA
STARTF  SYN     *
        BCI     1,FT
        BCI     1,FTWAT
FN      EQU     *-STARTA
STARTG  SYN     *
        BCI     1,GAL
        BCI     1,GR
        BCI     1,GRAIN
        BCI     1,GRAVIT
GN      EQU     *-STARTA
STARTH  SYN     *
        BCI     1,HP
        BCI     1,HR
HN      EQU     *-STARTA
STARTI  SYN     *
        BCI     1,INCH
        BCI     1,INHG
        BCI     1,IWATER
IN      EQU     *-STARTA
STARTJ  SYN     *
        BCI     1,JOULE
JN      EQU     *-STARTA
STARTK  SYN     *
        BCI     1,KCAL
        BCI     1,KELVIN
        BCI     1,KG
        BCI     1,KM
        BCI     1,KW
KN      EQU     *-STARTA
STARTL  SYN     *
        BCI     1,LB
        BCI     1,LITER
LN      EQU     *-STARTA
STARTM  SYN     *
        BCI     1,M
        BCI     1,MEGBAR
        BCI     1,MICRON
        BCI     1,MILE
        BCI     1,MIN
        BCI     1,MM
        BCI     1,MMHG
        BCI     1,MMICR
        BCI     1,MWATER
MN      EQU     *-STARTA
STARTN  SYN     *
NN      EQU     *-STARTA
STARTO  SYN     *
        BCI     1,OUNCE
        BCI     1,OZ
ON      EQU     *-STARTA
STARTP  SYN     *
        BCI     1,POISE
PN      EQU     *-STARTA
```

```
STARTQ SYN    *
       BCI    1,QUARTD
       BCI    1,QUARTL
QN     EQU    *-STARTA
STARTR SYN    *
       BCI    1,RANKIN
       BCI    1,ROD
RN     EQU    *-STARTA
STARTS SYN    *
       BCI    1,SEC
SN     EQU    *-STARTA
STARTT SYN    *
       BCI    1,TONM
       BCI    1,TONREF
       BCI    1,TONS
TN     EQU    *-STARTA
STARTU SYN    *
UN     EQU    *-STARTA
STARTV SYN    *
VN     EQU    *-STARTA
STARTW SYN    *
       BCI    1,WATT
WN     EQU    *-STARTA
STARTX SYN    *
XN     EQU    *-STARTA
STARTY SYN    *
       BCI    1,YD
YN     EQU    *-STARTA
STARTZ SYN    *
ZN     EQU    *-STARTA
       PZE    0
COUNT  PZE    AN
       PZE    BN
       PZE    CN
       PZE    DN
       PZE    EN
       PZE    FN
       PZE    GN
       PZE    HN
       PZE    IN
       PZE    JN
       PZE    KN
       PZE    LN
       PZE    MN
       PZE    NN
       PZE    ON
       PZE    PN
       PZE    QN
       PZE    RN
       PZE    SN
       PZE    TN
       PZE    UN
       PZE    VN
       PZE    WN
       PZE    XN
       PZE    YN
       PZE    ZN
FACTOR DEC    2.471E-08
```

```
DEC     1.0E+08             ANGST
DEC     0.9678E-03          ATM
DEC     0.9807              BAR
DEC     9.296E-08           BTU
DEC     0.02838E-03         BUSH
DEC     2.344E-02           CAL
DEC     1.                  CM
DEC     0.27777778E-01      CP
DEC     0.041667            DAY
DEC     1.0                 DEGC
DEC     1.8                 DEGF
DEC     980.655             DYNE
DEC     9.80665E+02         ERG
DEC     0.032808            FT
DEC     32.84E-03           FTWAT
DEC     0.2642E-03          GAL
DEC     1.0                 GR
DEC     15.432              GRAIN
DEC     0.7868253E-10       GRAVIT
DEC     0.3653E-10          HP
DEC     1.0                 HR
DEC     0.3937              INCH
DEC     28.96E-03           INHG
DEC     394.05E-03          IWATER
DEC     9.80665E-05         JOULE
DEC     2.344E-05           KCAL
DEC     1.0                 KELVIN
DEC     0.001               KG
DEC     1.0E-05             KM
DEC     0.2724E-10          KW
DEC     0.002205            LB
DEC     0.001               LITER
DEC     0.01                M
DEC     0.9807E-03          MEGBAR
DEC     1.0E+04             MICRON
DEC     6.2E-06             MILE
DEC     60.                 MIN
DEC     10.                 MM
DEC     735.559E-03         MMHG
DEC     1.0E+07             MMICR
DEC     10.01E-03           MWATER
DEC     0.03527             OUNCE
DEC     33.81E-03           OZ
DEC     0.27777778E-03      POISE
DEC     0.9081E-03          QUARTD
DEC     1.057E-03           QUARTL
DEC     1.8                 RANKIN
DEC     0.0019884           ROD
DEC     3600.               SEC
DEC     0.1E-05             TONM
DEC     7.74667E-12         TONREF
DEC     0.1102E-05          TONS
DEC     0.2724E-07          WATT
DEC     0.010936            YD
END
```

```
      SUBROUTINE ERROR1
      WRITE OUTPUT TAPE 6, 1000,
 1000 FORMAT(1H0/1H05X,104H...ERROR1... TWO OPERATORS SUCH AS * AND / AR
     1E WRITTEN CONSECUTIVELY IN THE FORMATION OF A COMPLEX UNIT.  )
      CALL EXIT
      END
```

```
      SUBROUTINE ERROR2
      WRITE OUTPUT TAPE 6, 1000
 1000 FORMAT(1H0/1H05X,75H...ERROR2... TOO MANY RIGHT PARENTHESES IN THE
     1 FORMATION OF A COMPLEX UNIT.    )
      CALL EXIT
      END
```

```
      SUBROUTINE ERROR3
      WRITE OUTPUT TAPE 6, 1000
 1000 FORMAT(1H0/1H05X,74H...ERROR3... TOO MANY LEFT PARENTHESES IN THE
     1FORMATION OF A COMPLEX UNIT.    )
      CALL EXIT
      END
```

```
      SUBROUTINE ERROR4
      WRITE OUTPUT TAPE 6, 1000
 1000 FORMAT(1H0/1H05X,68H...ERROR4... THE UNIT IS NOT INTRODUCED PROPER
     1LY OR A PROGRAM ERROR.    )
      CALL EXIT
      END
```

```
      SUBROUTINE ERROR5
      WRITE OUTPUT TAPE 6, 1000
 1000 FORMAT(1H0/1H05X,74H...ERROR5... NO SIMPLE UNIT IS PRESENT AFTER A
     1 MULTIPLICATION OPERATOR(*).   )
      CALL EXIT
      END
```

```
      SUBROUTINE ERROR6
      WRITE OUTPUT TAPE 6, 1000
 1000 FORMAT(1H0/1H05X,64H...ERROR6... NO PROPER OPERATOR EXISTS BETWEEN
     1 TWO SIMPLE UNITS.    )
      CALL EXIT
      END
```

```
      SUBROUTINE ERROR7
      WRITE OUTPUT TAPE 6, 1000
 1000 FORMAT(1H0/1H05X,68H...ERROR7... THE UNIT IS NOT INTRODUCED PROPER
     1LY OR A PROGRAM ERROR.    )
      CALL EXIT
      END
```

```
      SUBROUTINE WRITE1
      WRITE OUTPUT TAPE 6, 1,
    1 FORMAT(1H0/1H05X,89H...ERROR... CHECK THE LIST OF ABBREVIATION OF
     1UNITS WHETHER YOU ARE USING THE RIGHT ONES.    )
      CALL EXIT
      END
```

```
C                         SUBROUTINE - LNPLOT
C       A LOG - LOG PLOT ROUTINE
C
        SUBROUTINE LNPLOT(N,X,Y,SYMBOL)
C
        DIMENSION X(200),Y(200),XL(200),YL(200),POINT(121,97),YNUMB(97),
       1NSX(10),NSY(10),XNUMB(10)
C
        DO 5 I=1,N
        XL(I)=LOGF(X(I))*0.434294
        YL(I)=LOGF(Y(I))*0.434294
      5 CONTINUE
C
        XMIN=XL(1)
        XMAX=XL(1)
        YMIN=YL(1)
        YMAX=YL(1)
C
        DO 50 I=2,N
        IF(XMIN-XL(I)) 20,20,15
     15 XMIN=XL(I)
     20 IF(XMAX-XL(I)) 25,30,30
     25 XMAX=XL(I)
     30 IF(YMIN-YL(I)) 40,40,35
     35 YMIN=YL(I)
     40 IF(YMAX-YL(I)) 45,50,50
     45 YMAX=YL(I)
     50 CONTINUE
C
        IF(INTF(XMIN)-XMIN)54,56,55
     54 XMIN=INTF(XMIN)
        GO TO 56
     55 XMIN=INTF(XMIN)-1.0
     56 IF(INTF(XMAX)-XMAX) 59,61,60
     59 XMAX=INTF(XMAX)+1.0
        GO TO 61
     60 XMAX=INTF(XMAX)
     61 IF(INTF(YMIN)-YMIN) 64,66,65
     64 YMIN=INTF(YMIN)
        GO TO 66
     65 YMIN=INTF(YMIN)-1.0
     66 IF(INTF(YMAX)-YMAX) 69,75,70
     69 YMAX=INTF(YMAX)+1.0
        GO TO 75
     70 YMAX=INTF(YMAX)
C
     75 XCYCLE=(XMAX-XMIN)+0.5
        XSPACE=120./INTF(XCYCLE)
        NXSPCE=XINTF(XSPACE+0.5)
C
        YCYCLE=(YMAX-YMIN)+0.5
        YSPACE=12.*XSPACE/20.
        NYSPCE=XINTF(YSPACE+0.5)
        NYH=XINTF(YSPACE*INTF(YCYCLE)+0.5)+1
C
        DO 80 I=1,121
        DO 80 J=1,NYH
B       POINT(I,J)=606060606060
        YNUMB(J)=0.0
     80 CONTINUE
```

```
C
      DO 85 I=1,8
      AI=FLOATF(I)+1.0
      YSCALE=LOGF(AI)*0.434294*YSPACE+0.5
      NSY(I)=XINTF(YSCALE)
      XSCALE=LOGF(AI)*0.434294*XSPACE+0.5
      NSX(I)=XINTF(XSCALE)
   85 CONTINUE
C
      DO 95 I=1,121
      DO 95 J=1,NYH,NYSPCE
      NPOWER=(J-1)/NYSPCE
      POINT(I,J)=1H-
      YNUMB(J)=10.**(YMIN)*(10.**NPOWER)
      IF(J-NYH) 86,95,95
   86 DO 90 K=1,8
      AK=K
      JK=J+NSY(K)
   90 YNUMB(JK)=(AK+1.0)*YNUMB(J)
   95 CONTINUE
C
      DO 105 J=1,NYH
      DO 105 I=1,121,NXSPCE
      POINT(I,J)=1HI
  105 CONTINUE
C
      DO 115 I=1,121,NXSPCE
      DO 115 J=1,NYH,NYSPCE
      POINT(I,J)=1H+
      IF(I-121) 106,115,115
  106 DO 110 K=1,8
      JK=J+NSY(K)
      POINT(I,JK)=1H+
      IK=I+NSX(K)
  110 POINT(IK,J)=1H+
  115 CONTINUE
C
      DO 120 K=1,N
      AI=(XL(K)-XMIN)*XSPACE+0.5
      I=XINTF(AI)+1
      AJ=(YL(K)-YMIN)*YSPACE+0.5
      J=XINTF(AJ)+1
      POINT(I,J)=SYMBOL
  120 CONTINUE
C
      DO 135 JREV=1,NYH
      J=NYH+1-JREV
      IF(YNUMB(J)) 130,125,130
  125 WRITE OUTPUT TAPE 6, 500, (POINT(I,J),I=1,121)
      GO TO 135
  130 WRITE OUTPUT TAPE 6, 501, YNUMB(J),(POINT(I,J),I=1,121)
  135 CONTINUE
C
      XNUMB(1)=10.**(XMIN)
      NCYCLE=XINTF(XCYCLE)
      IH=NCYCLE+1
      DO 136 I=2,IH
      XNUMB(I)=XNUMB(I-1)*10.
```

```
  136 CONTINUE
      IF(NCYCLE-2) 145,150,140
  140 IF(NCYCLE-4) 155,160,165
  145 WRITE OUTPUT TAPE 6, 502, (XNUMB(I),I=1,IH)
      GO TO 200
  150 WRITE OUTPUT TAPE 6, 503, (XNUMB(I),I=1,IH)
      GO TO 200
  155 WRITE OUTPUT TAPE 6, 504, (XNUMB(I),I=1,IH)
      GO TO 200
  160 WRITE OUTPUT TAPE 6, 505, (XNUMB(I),I=1,IH)
      GO TO 200
  165 WRITE OUTPUT TAPE 6, 506, XNUMB(1),XNUMB(IH)
C
  500 FORMAT(8X,121A1)
  501 FORMAT(1PE8.1,(121A1))
  502 FORMAT(6X,E8.1,110X,E8.1)
  503 FORMAT(6X,E8.1,50X,E8.1,52X,E8.1)
  504 FORMAT(6X,E8.1,5X,2H 2,5X,2H 3,3X,2H 4,2X,2H 5,3H  6,3H  7,1X,E8.1
     1,2(7X,2H 2,5X,2H 3,3X,2H 4,2X,2H 5,3H  6,3H  7,1X,E8.1))
  505 FORMAT(6X,E8.1,20X,E8.1,3(22X,E8.1))
  506 FORMAT(6X,E8.1,110X,E8.1)
C
  200 RETURN
      END
```

```
C                         SUBROUTINE - LAGINT
C     A LAGRANGIAN INTERPOLATION METHOD USING ARBITRARY SET OF ORDINATES
C     REFERENCE - HILDEBRAND, INTRODUCTION TO NUMERICAL ANALYSIS, 1956
C                                 P60
C
      SUBROUTINE LAGINT(N,X,Y,NV,XV,YV)
C
      DIMENSION X(500),Y(500),XV(500),YV(500)
C
      DO 17 K=1,NV
C
C     CHECK IF XV(K) IS INSIDE THE RANGE OF X VALUES
C
      IF(XV(K)-X(1)) 13,4,3
    3 IF(XV(K)-X(N)) 4,4,14
C
C     CHECK IF XV(K) IS IDENTICAL TO ANY ONE OF THE VALUES OF X
C
    4 DO 5 I=1,N
      IF(XV(K)-X(I)) 6,15,5
    5 CONTINUE
C
C     MAIN COMPUTATION
C
    6 PAI=1.
      DO 7 I=1,N
    7 PAI=PAI*(XV(K)-X(I))
      BOT1=(XV(K)-X(1))
      DO 8 I=2,N
    8 BOT1=BOT1*(X(1)-X(I))
      TERM1=Y(1)/BOT1
      NL=N-1
      TERMIN=0.
      DO 11 J=2,NL
      BOTIN=(XV(K)-X(J))
      JL=J-1
      JH=J+1
      DO 9 I=1,JL
    9 BOTIN=BOTIN*(X(J)-X(I))
      DO 10 I=JH,N
   10 BOTIN=BOTIN*(X(J)-X(I))
      TERMIN=TERMIN+Y(J)/BOTIN
   11 CONTINUE
      BOTN=(XV(K)-X(N))
      DO 12 I=1,NL
   12 BOTN=BOTN*(X(N)-X(I))
      TERMN=Y(N)/BOTN
C
      SIGMA=TERM1+TERMIN+TERMN
      YV(K)=PAI*SIGMA
      GO TO 17
C
   13 WRITE OUTPUT TAPE 6,500,K,XV(K)
      GO TO 17
   14 WRITE OUTPUT TAPE 6, 501, K,XV(K)
      GO TO 17
   15 YV(K)=Y(I)
   17 CONTINUE
C
  500 FORMAT(/11X8H WHEN K=I3,8H, XV(K)=E14.7,37H IS LESS THAN THE SMALL
```

```
     1EST VALUE OF X/)
  501 FORMAT(/11X8H WHEN K=I3,8H, XV(K)=E14.7,38H IS LARGER THAN THE BIG
     1GEST VALUE OF X/)
C
   20 RETURN
      END
```

```
C                         SUBROUTINE - GAUSS
C
      SUBROUTINE GAUSS(M,A,X)
C
      DIMENSION A(50,51),B(50,51),X(50),AI(50,51)
C
      N=M+1
C
      DO 1 I=1,M
      DO 1 J=1,N
      AI(I,J)=A(I,J)
    1 CONTINUE
C
      DO 100 K=1,M
      DO 100 I=1,M
      IF(I-K) 100,5,40
    5 DO 30 J=1,N
      IF(J-I) 10,15,15
   10 B(I,J)=0.0
      GO TO 30
   15 IF(AI(I,I)) 16,20,16
   16 B(I,J)=AI(I,J)/AI(I,I)
      GO TO 30
   20 L=I+1
   21 IF(AI(L,L)) 25,35,25
   25 B(I,J)=AI(L,J)/AI(L,L)
      AI(L,J)=AI(I,J)
      GO TO 30
   35 L=L+1
      IF(L-M)21,21,36
   36 WRITE OUTPUT TAPE 6, 1003,K
      GO TO 160
   30 CONTINUE
      GO TO 100
   40 DO 80 J=1,N
      IF(J-(K+1))80,45,45
   45 AI(I,J)=AI(I,J)-B(K,J)*AI(I,K)
   80 CONTINUE
  100 CONTINUE
C
      X(M)=B(M,N)
      ML=M-1
      DO 120 I=1,ML
      K=M-I
      SUM=0.0
      KH=K+1
  105 DO 110 J=KH,M
  110 SUM=SUM+B(K,J)*X(J)
      X(K)=B(K,N)-SUM
  120 CONTINUE
C
 1003 FORMAT(1H010X,46H NO SOLUTIONS BECAUSE OF ZERO COEFFICIENTS ON I4,
     114HTH ELIMINATION)
  150 RETURN
  160 CALL EXIT
      END
```

```
C                          SUBROUTINE - LEASQU
C      POLYNOMIAL FIT BY THE LEAST-SQUARES METHOD
C
       SUBROUTINE LEASQU(N,X,Y,MDEGRE,C)
C
       DIMENSION A(50,51),X(200),Y(200),C(50),S(200),V(200)
C
       M=MDEGRE+1
       M1=M+1
       MH=2*MDEGRE+1
C
       DO 5 I=1,MH
       S(I)=0.0
       DO 5 K=1,N
       S(I)=S(I)+X(K)**(MH-I)
     5 CONTINUE
C
       DO 10 I=1,M
       V(I)=0.0
       DO 10 K=1,N
       V(I)=V(I)+Y(K)*(X(K)**(M-I))
    10 CONTINUE
C
       DO 20 I=1,M
       A(I,M1)=V(I)
       DO 20 J=1,M
       IJL=I+J-1
       A(I,J)=S(IJL)
    20 CONTINUE
C
       CALL GAUSS(M,A,C)
C
       RETURN
       END
```

```
C                       SUBROUTINE - MATMPY
C     COMPUTING THE PRODUCT OF TWO SQUARE MATRICES
C
      SUBROUTINE MATMPY(N,A,B,C)
C
      DIMENSION A(50,51),B(50,51),C(50,51)
C
      DO 30 I=1,N
      DO 30 J=1,N
      C(I,J)=0.0
      DO 20 K=1,N
      C(I,J)=C(I,J)+A(I,K)*B(K,J)
   20 CONTINUE
   30 CONTINUE
      RETURN
      END
```

```
C                        SUBROUTINE - MATINV
C     COMPUTING THE INVERSE OF A SQUARE MATRIX BY USE OF GAUSS-
C     ELIMINATION METHOD
C
      SUBROUTINE MATINV(M,A,B)
      DIMENSION A(50,51),B(50,51),X(50)
C
      N=M+1
C
      DO 20 I=1,M
   20 A(I,N)=0.0
      DO 30 I=1,M
      A(I,N)=1.0
      CALL GAUSS(M,A,X)
      DO 25 J=1,M
      B(J,I)=X(J)
   25 CONTINUE
      DO 26 K=1,M
      A(K,N)=0.0
   26 CONTINUE
   30 CONTINUE
C
      RETURN
      END
```

```
C                     SUBROUTINE - SIMPSN
C     A SUBROUTINE TO COMPUTE INTEGRAL OF A FUNCTION BY THE SIMPSONS
C     RULE
C
      SUBROUTINE SIMPSN(N,Y,H,A)
C
      DIMENSION Y(200)
C
      CHECK=FLOATF(N)/2.0
      ICHECK=CHECK
      IF(CHECK-FLOATF(ICHECK)) 1,15,1
C
    1 K=1
    2 NL=N-1
      NLL=N-2
      SUMEVE=0.0
      SUMODD=0.0
      DO 5 I=2,NL,2
      SUMEVE=SUMEVE+Y(I)
    5 CONTINUE
      DO 10 I=3,NLL,2
      SUMODD=SUMODD+Y(I)
   10 CONTINUE
C
      A=(Y(1)+SUMEVE*4.+SUMODD*2.+Y(N))*H/3.
C
      GO TO (100,20),K
C
   15 N=N-1
      K=2
      GO TO 2
C
   20 N=N+1
      ALAST=(Y(N-1)+Y(N))*H/2.0
      A=A+ALAST
C
  100 RETURN
      END
```

```
C                  SUBROUTINE - RUNMOD
C     A RUNGE-KUTTA METHOD MODIFIED BY GILL TO SOLVE AN ORDINARY
C     DIFFERENTIAL EQUATION
C     REFERENCE - RALSTON AND WILF, MATHEMATICAL METHOD FOR DIGITAL
C                                 COMPUTERS, 1960, P110
C
      SUBROUTINE RUNMOD(N,X,Y,H)
      DIMENSION X(500),Y(500)
C
      Q0=0.
      NL=N-1
      DO 2 I=1,NL
      U=X(I)
      V=Y(I)
      T1=H*FUNCT(U,V)
      Q1=Q0+3.*(T1-2.*Q0)/2.-T1/2.
      Y2=Y(I)+(T1-2.*Q0)/2.
      U=X(I)+H/2.
      V=Y2
      T2=H*FUNCT(U,V)
      Q2=Q1+3.*(1.-SQRTF(0.5))*(T2-Q1)-(1.-SQRTF(0.5))*T2
      Y3=Y2+(1.-SQRTF(0.5))*(T2-Q1)
      V=Y3
      T3=H*FUNCT(U,V)
      Q3=Q2+3.*(1.+SQRTF(0.5))*(T3-Q2)-(1.+SQRTF(0.5))*T3
      Y4=Y3+(1.+SQRTF(0.5))*(T3-Q2)
      U=X(I)+H
      V=Y4
      T4=H*FUNCT(U,V)
      Q4=Q3+(T4-2.*Q3)/2.-T4/2.
      Y(I+1)=Y4+(T4-2.*Q3)/6.
      X(I+1)=X(I)+H
      Q0=Q4
    2 CONTINUE
C
      RETURN
      END
```

APPENDIX 2. LISTING OF CONTROL PROGRAMS AND NUMERICAL RESULTS

```
CCCNVRT         CONTROL PROGRAM FOR SUBROUTINE CONVRT (CCNVRT)
C
C     CASE1 ... THE UNITS ARE INTRODUCED AS HOLLERITH ARGUMENTS
C     PROBLEM ... FIND A CONVERSION FACTOR FOR A UNIT CONVERSION
C                 FROM BTU/SEC TO FT-LB/SEC
C
      CALL CONVRT(0, 8H BTU/SEC, 12H (FT*LB)/SEC, FACTOR)
      WRITE OUTPUT TAPE 6, 100, FACTOR
C
  100 FORMAT(1H1/1H015X,8H RESULTS/1H010X,24H ORIGINAL UNIT = BTU/SEC/1H
     1010X,28H  DESIRED UNIT = (FT*LB)/SEC/1H010X,20H CONVERSION FACTOR
     2= E12.5)
C
      CALL EXIT
      END
```

```
        RESULTS
ORIGINAL UNIT = BTU/SEC
 DESIRED UNIT = (FT*LB)/SEC
CONVERSION FACTOR = 0.77820E 03
```

```
CCCNVRT        CONTROL PROGRAM FOR SUBROUTINE CONVRT (CCNVRT)
C
C      CASE2 ... THE UNITS ARE INTRODUCED AS INPUT DATA
C      PROBLEM ... FIND A CONVERSION FACTOR FOR A UNIT CONVERSION
C                  FROM CENTI POISE(CP) TO LB/(FT-HR)
C
      DIMENSION UNIT1(30),UNIT2(30)
      N=12
      READ INPUT TAPE 5, 100, (UNIT1(I),I=1,N)
      READ INPUT TAPE 5, 100, (UNIT2(I),I=1,N)
      CALL CONVRT(N,UNIT1,UNIT2,FACTOR)
      WRITE OUTPUT TAPE 6, 101, (UNIT1(I),I=1,N),(UNIT2(I),I=1,N),FACTOR
C
  100 FORMAT(12A6)
  101 FORMAT(1H1/1H015X,8H RESULTS/1H010X,17H ORIGINAL UNIT = 12A6/1H010
     1X,17H  DESIRED UNIT = 12A6/1H010X,20H CONVERSION FACTOR = E12.5)
C
      CALL EXIT
      END
```

```
*     DATA
  CP
  LB/(FT*HR)
```

RESULTS

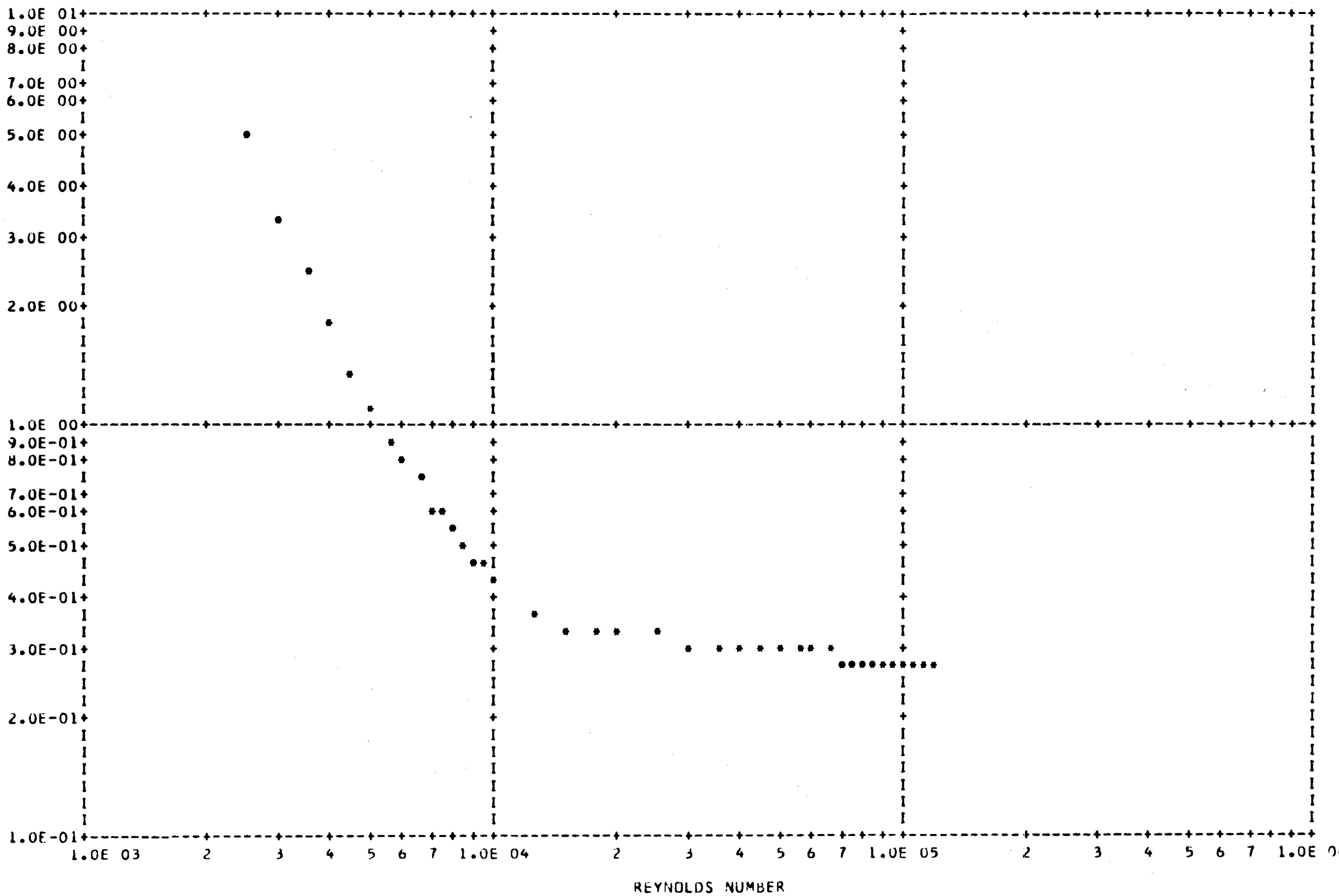
```
ORIGINAL UNIT =   CP

 DESIRED UNIT =   LB/(FT*HR)

CONVERSION FACTOR = 0.24195E 01
```

```
CCLNPLT         CONTROL PROGRAM FOR SUBROUTINE LNPLOT (CLNPLT)
C
C      PROBLEM ... TO PLOT A DISPERSION GROUP VS. REYNOLDS NUMBER ON
C                  A LOG - LOG SCALE
C
       DIMENSION REYN(200),DISP(200)
C
       WRITE OUTPUT TAPE 6, 1000
C
     5 REYN(1)=2500.
       DO 10 I=2,16
       REYN(I)=REYN(I-1)+500.0
    10 CONTINUE
C
       DO 11 I=17,20
       REYN(I)=REYN(I-1)+2500.
    11 CONTINUE
C
       DO 15 I=1,20
       TERM=0.63082532*(LOGF(REYN(I))**2)-12.55317450*(LOGF(REYN(I)))+61.
      125958824
       DISP(I)=EXPF(TERM)
    15 CONTINUE
C
       DO 20 I=21,40
       REYN(I)=REYN(I-1)+5000.
       F=0.0014+0.09*((1./REYN(I))**0.27)
       DISP(I)=3.57*SQRTF(F)
    20 CONTINUE
C
       N=40
       SYMBOL=1H*
       CALL LNPLOT(N,REYN,DISP,SYMBOL)
C
       WRITE OUTPUT TAPE 6, 1001
C
  1000 FORMAT(1H1/1H030X,47H A PLOT OF DISPERSION GROUP VS. REYNOLDS NUMB
      1ER////)
  1001 FORMAT(1H060X,16H REYNOLDS NUMBER)
C
       CALL EXIT
       END
```

A PLOT OF DISPERSION GROUP VS. REYNOLDS NUMBER



```
CCLAGIN         CONTROL PROGRAM FOR SUBROUTINE LAGINT(CLAGIN)
C
C      PROBLEM ... FIND MASS RATES FROM PRESSURE DROP DATA IN AN ORIFICE
C                  MANOMETER BY AN INTERPOLATION OF CALIBRATION CURVE
C                   (SEE TEXT FOR NUMERICAL VALUES)
C
      DIMENSION PDROP(20),RATE(20),PDROPV(20),RATEV(20)
C
      READ INPUT TAPE 5, 100, N,(PDROP(I),I=1,N)
      READ INPUT TAPE 5, 101, (RATE(I),I=1,N)
      READ INPUT TAPE 5, 100, NV,(PDROPV(I),I=1,NV)
C
      CALL LAGINT(N,PDROP,RATE,NV,PDROPV,RATEV)
C
      WRITE OUTPUT TAPE 6, 102
      WRITE OUTPUT TAPE 6, 103, (PDROP(I),RATE(I),I=1,N)
      WRITE OUTPUT TAPE 6, 104
      WRITE OUTPUT TAPE 6, 103, (PDROPV(I),RATEV(I),I=1,NV)
C
  100 FORMAT(I4/(12F5.1))
  101 FORMAT(12F5.2)
  102 FORMAT(1H1/1H010X,20H   CALIBRATION TABLE/1H012X,18H PRESSURE
     1MASS/15X,16H DROP         RATE/)
  103 FORMAT(15X,F5.1,6X,F5.2)
  104 FORMAT(////1H017X,8H RESULTS/1H012X,18H PRESSURE     MASS/15X,16H
     1DROP        RATE/)
C
      CALL EXIT
      END



*     DATA
   7
  1.0  2.0  3.0  4.0  5.0  6.0  7.0
 1.32 1.87 2.29 2.64 2.96 3.23 3.49
   4
  1.5  3.5  4.0  5.5
```

CALIBRATION TABLE

PRESSURE DROP	MASS RATE
1.0	1.32
2.0	1.87
3.0	2.29
4.0	2.64
5.0	2.96
6.0	3.23
7.0	3.49

RESULTS

PRESSURE DROP	MASS RATE
1.5	1.61
3.5	2.47
4.0	2.64
5.5	3.10

```
CCGAUSS         CONTROL PROGRAM FOR SUBROUTINE GAUSS (CGAUSS)
C
C     PROBLEM ... COMPUTE THE RATE OF PRODUCT(P),WASTE(W), AND AIR
C                 LEAVING ABSORBER (A) IN AN ACETONE RECOVERY SYSTEM
C                  (FOR EQUATIONS, SEE TEXT)
C
      DIMENSION COECNT(50,51),X(50)
C
      READ INPUT TAPE 5, 100, M,((COECNT(I,J),J=1,M),I=1,M)
    5 READ INPUT TAPE 5, 101, F
C
      N=M+1
      COECNT(1,N)=F*0.0295
      COECNT(2,N)=120.
      COECNT(3,N)=F*0.9705
C
      CALL GAUSS(M,COECNT,X)
C
      P=X(1)
      W=X(2)
      A=X(3)
C
      WRITE OUTPUT TAPE 6, 102, F
      WRITE OUTPUT TAPE 6, 103, ((COECNT(I,J),J=1,N),I=1,M)
      WRITE OUTPUT TAPE 6, 104, P,W,A
C
  100 FORMAT(I4/(3F5.2))
  101 FORMAT(F10.0)
  102 FORMAT(1H1/1H010X,21H THE RATE OF FEED IS F9.0,7H LB/HR.   )
  103 FORMAT(1H0/1H010X,23H ORIGINAL EQUATIONS ARE//(16X,F5.2,7H * P +
     1F5.2,7H * W + F5.2,7H * A = F8.1))
  104 FORMAT(1H0/1H010X,19H RESULTS (IN LB/HR)/1H015X,3H P=F8.1/16X,3H W
     1=F8.1/16X,3H A=F8.1)
C
      GO TO 5
      END


*     DATA
   3
 0.99 0.05 0.00
 0.01 0.95 0.00
 0.00 0.00 1.00
   50000.0
```

```
THE RATE OF FEED IS    50000. LB/HR.


ORIGINAL EQUATIONS ARE

     0.99 * P +  0.05 * W +  0.   * A =    1475.0
     0.01 * P +  0.95 * W +  0.   * A =     120.0
     0.   * P +  0.   * W +  1.00 * A =   48525.0


RESULTS (IN LB/HR)

     P=  1484.3
     W=   110.7
     A= 48525.0
```

```
CCLEASQ        CONTROL PROGRQM FOR SUBROUTINE LEASQU (CLEASQ)
C
C     PROBLEM ... FIND A STRAIGHT LINE FORMULA WHICH WILL BEST FIT THE
C                 DATA OBTAINED IN A CALIBRATION EXPERIMENT OF A FLOW
C                 METER
C                   (SEE TEXT FOR NUMERICAL VALUES)
C
      DIMENSION READNG(200),OBSRVD(200),C(50)
C
      READ INPUT TAPE 5, 100, N,(READNG(I),I=1,N)
      READ INPUT TAPE 5, 101, (OBSRVD(I),I=1,N)
C
      MDEGRE=1
      CALL LEASQU(N,READNG,OBSRVD,MDEGRE,C)
C
      WRITE OUTPUT TAPE 6, 102
      WRITE OUTPUT TAPE 6, 103, (READNG(I),OBSRVD(I),I=1,N)
      WRITE OUTPUT TAPE 6, 104, C(1),C(2)
C
  100 FORMAT(I4/(12F6.1))
  101 FORMAT(12F6.1)
  102 FORMAT(1H1/1H012X,18H CALIBRATION TABLE/1H010X,22H READINGS     OB
     1SERVED/13X,19H (GPH)        (GPH)//)
  103 FORMAT(12X,F6.1,7X,F6.1)
  104 FORMAT(1H0/1H0/1H015X,7H RESULT/1H010X,14H FLOW RATE = (E12.5,15H)
     1 * READING + (E12.5,1H)    )
C
      CALL EXIT
      END
```

```
*     DATA
  10
  20.0  30.0  40.0  50.0  60.0  70.0  80.0  90.0 100.0 120.0
  19.8  29.7  39.7  49.6  60.2  70.3  80.2  90.0 102.5 120.5
```

CALIBRATION TABLE

READINGS (GPH)	OBSERVED (GPH)
20.0	19.8
30.0	29.7
40.0	39.7
50.0	49.6
60.0	60.2
70.0	70.3
80.0	80.2
90.0	90.0
100.0	102.5
120.0	120.5

RESULT

FLOW RATE = (0.10161E 01) * READING + (-0.81428E 00)

```
CCMTMPY          CONTROL PROGRAM FOR SUBROUTINE MATMPY (CMTMPY)
C
C      PROBLEM ... FIND THE PRODUCT OF TWO MATRICES, A AND B
C                     (SEE TEXT FOR NUMERICAL VALUES)
C
       DIMENSION A(50,51),B(50,51),C(50,51)
C
       READ INPUT TAPE 5, 100, N
       READ INPUT TAPE 5, 101, ((A(I,J),J=1,N),I=1,N)
       READ INPUT TAPE 5, 101, ((B(I,J),J=1,N),I=1,N)
C
       CALL MATMPY(N,A,B,C)
C
       WRITE OUTPUT TAPE 6, 102, ((A(I,J),J=1,N),I=1,N)
       WRITE OUTPUT TAPE 6, 103, ((B(I,J),J=1,N),I=1,N)
       WRITE OUTPUT TAPE 6, 104, ((C(I,J),J=1,N),I=1,N)
C
   100 FORMAT(I4)
   101 FORMAT(4F10.5)
   102 FORMAT(1H1/1H010X,18H ORIGINAL MATRICES/1H010X,5H A = 4F10.5/(16X,
      14F10.5))
   103 FORMAT(1H0/1H010X,5H B = 4F10.5/(16X,4F10.5))
   104 FORMAT(////1H012X,7H RESULT/1H010X,19H THE PRODUCT MATRIX/1H010X,5
      1H C = 4F10.5/(16X,4F10.5))
C
       CALL EXIT
       END
```

```
       DATA
    4
    8.467       5.137       3.141       2.063
    5.137       6.421       2.617       2.003
    3.141       2.617       4.128       1.628
    2.063       2.003       1.628       3.446
    0.25087    -0.16214    -0.08116    -0.01760
   -0.16214     0.33074    -0.05994    -0.06686
   -0.08116    -0.05994     0.37988    -0.09604
   -0.01760    -0.06686    -0.09604     0.38497
```

ORIGINAL MATRICES

A =	8.46700	5.13700	3.14100	2.06300
	5.13700	6.42100	2.61700	2.00300
	3.14100	2.61700	4.12800	1.62800
	2.06300	2.00300	1.62800	3.44600

B =	0.25087	-0.16214	-0.08116	-0.01760
	-0.16214	0.33074	-0.05994	-0.06686
	-0.08116	-0.05994	0.37988	-0.09604
	-0.01760	-0.06686	-0.09604	0.38497

RESULT

THE PRODUCT MATRIX

C =	0.99997	-0.00003	-0.00002	0.00005
	-0.00003	0.99998	-0.00002	0.00004
	-0.00002	-0.00002	1.00000	0.00002
	0.00000	-0.00000	-0.00000	1.00002

```
CCMTINV         CONTROL PROGRAM FOR SUBROUTINE MATINV (CMTINV)
C
C     PROBLEM ... INVERT MATRIX A
C                   (SEE TEXT FOR NUMERICAL VALUES)
C
      DIMENSION A(50,51),B(50,51)
C
      READ INPUT TAPE 5, 100, N, ((A(I,J),J=1,N),I=1,N)
C
      CALL MATINV(N,A,B)
C
      WRITE OUTPUT TAPE 6, 101, ((A(I,J),J=1,N),I=1,N)
      WRITE OUTPUT TAPE 6, 102, ((B(I,J),J=1,N),I=1,N)
C
  100 FORMAT(I4/(4F8.5))
  101 FORMAT(1H1/1H010X,16H ORIGINAL MATRIX/1H010X,5H A = 4F10.5/(16X,4F
     110.5))
  102 FORMAT(////1H012X,7H RESULT/1H010X,15H INVERSE MATRIX/1H010X,5H B
     1= 4F10.5/(16X,4F10.5))
C
      CALL EXIT
      END
```

```
*     DATA
   4
 8.467   5.137   3.141   2.063
 5.137   6.421   2.617   2.003
 3.141   2.617   4.128   1.628
 2.063   2.003   1.628   3.446
```

ORIGINAL MATRIX

A =	8.46700	5.13700	3.14100	2.06300
	5.13700	6.42100	2.61700	2.00300
	3.14100	2.61700	4.12800	1.62800
	2.06300	2.00300	1.62800	3.44600

RESULT

INVERSE MATRIX

B =	0.25087	-0.16214	-0.08116	-0.01760
	-0.16214	0.33074	-0.05994	-0.06686
	-0.08116	-0.05994	0.37988	-0.09604
	-0.01760	-0.06686	-0.09604	0.38497

```
CCSMPSN          CONTROL PROGRAM FOR SUBROUTINE SIMPSN (CSMPSN)
C
C      PROBLEM ... FIND AN AREA UNDER THE CONCENTRATION CURVE OBTAINED
C                  BY LEVENSPIEL AND SMITH IN THEIR DISPERSION EXPERIMENT
C                    (SEE TEXT FOR CONCENTRATION DATA)
C                  ALSO, COMPUTE DIMENSIONLESS VARIANCE WHICH WILL BE
C                  USED TO DETERMINE A DISPERSION COEFFICIENT.
C                  SOME OTHER NECESSARY DATA ARE
C                    VELOCITY = 1.17 FT/SEC
C                    LENGTH   = 9.0 FT
C
      DIMENSION CONC(200),TIME(200),TCONC(200),TSQCON(200)
C
      READ INPUT TAPE 5, 100, N,VEL,AL
      READ INPUT TAPE 5, 101, (TIME(I),I=1,N)
      READ INPUT TAPE 5, 102, (CONC(I),I=1,N)
C
      H=TIME(2)-TIME(1)
C
      CALL SIMPSN(N,CONC,H,AREA)
C
      DO 10 I=1,N
      TCONC(I)=TIME(I)*CONC(I)
      TSQCON(I)=TIME(I)**2*CONC(I)
   10 CONTINUE
C
      CALL SIMPSN(N,TCONC,H,FIRSTM)
C
      CALL SIMPSN(N,TSQCON,H,SECNDM)
C
      VAR=(SECNDM/AREA-((FIRSTM/AREA)**2))*((VEL/AL)**2)
C
      WRITE OUTPUT TAPE 6, 103, VEL,AL
      WRITE OUTPUT TAPE 6, 104, (TIME(I),CONC(I),I=1,N)
      WRITE OUTPUT TAPE 6, 105, AREA,VAR
C
  100 FORMAT(I4,2F5.2)
  101 FORMAT(14F5.1)
  102 FORMAT(7F10.5)
  103 FORMAT(1H1/1H010X,18H EXPERIMENTAL DATA/1H012X,11H VELOCITY =F5.2,
     17H FT/SEC/13X,11H LENGTH   =F5.2,3H FT/1H0/1H013X,14H TIME     CON
     2C//)
  104 FORMAT(14X,F5.1,4X,F5.1)
  105 FORMAT(////1H010X,8H RESULTS/1H010X,30H AREA UNDER THE CONC. CURVE
     1 = E12.5/1H010X,30H DIMENSIONLESS VARIANCE      = E12.5)
C
      CALL EXIT
      END


*     DATA
  22 1.17 9.00
  0.0  2.0  4.0  6.0  8.0 10.0 12.0 14.0 16.0 18.0 20.0 22.0 24.0 26.0
 28.0 30.0 32.0 34.0 36.0 38.0 40.0 42.0
   0.00000  11.00000  53.00000  64.00000  58.00000  48.00000  39.00000
  29.00000  22.00000  16.00000  11.00000   9.00000   7.00000   5.00000
   4.00000   2.00000   2.00000   2.00000   1.00000   1.00000   1.00000
   1.00000
```

EXPERIMENTAL DATA

VELOCITY = 1.17 FT/SEC
LENGTH = 9.00 FT

TIME	CONC
0.	0.
2.0	11.0
4.0	53.0
6.0	64.0
8.0	58.0
10.0	48.0
12.0	39.0
14.0	29.0
16.0	22.0
18.0	16.0
20.0	11.0
22.0	9.0
24.0	7.0
26.0	5.0
28.0	4.0
30.0	2.0
32.0	2.0
34.0	2.0
36.0	1.0
38.0	1.0
40.0	1.0
42.0	1.0

RESULTS

AREA UNDER THE CONC. CURVE = 0.76400E 03

DIMENSIONLESS VARIANCE = 0.79132E 00

```
CCRUNMO          CONTROL PROGRAM FOR SUBROUTINE RUNMOD (CRUNMO)
C
C     PROBLEM ... FIND THE CONCENTRATION OF SALT IN A MIXING TANK AS
C                 A FUNCTION OF TIME
C                   (SEE TEXT FOR THE STATEMENT OF PROBLEM)
C
      DIMENSION TIME(200),CONC(200)
C
      N=181
      H=1.0
      TIME(1)=0.0
      CONC(1)=0.0
C
      CALL RUNMOD(N,TIME,CONC,H)
C
      WRITE OUTPUT TAPE 6, 100
      WRITE OUTPUT TAPE 6, 101, (TIME(I),CONC(I),I=1,N)
C
  100 FORMAT(1H1/1H015X,20H MIXING TANK PROBLEM/1H010X, 14H SALT 2 LB/GA
     1L/14X,18H 5 GAL/MIN ------+/30X,2H I/27X,5H I  I,7X,2H I/27X,5H I
     2 I,7X,2H I/27X,2H I,10X,2H I/27X,14H I----------I/27X,14H I V0=10
     30GAL I/27X,2H I,4X,2H C,4X,8H I-----+/27X,14H ------------4X,2H I
     4/45X,2H I/43X,9H C LB/GAL/42X,10H 1 GAL/MIN////1H021X,7H RESULT/1H
     5017X,5H TIME7X,5H CONC/18X,6H (MIN),4X,9H (LB/GAL)//)
  101 FORMAT(17X,F6.1,5X,F9.5)
C
      CALL EXIT
      END
```

```
C                FUNCTION RPOGRAM FOR MIXING TANK PROBLEM
C
      FUNCTION FUNCT(TIME,CONC)
C
      FUNCT=(10.-5.0*CONC)/(100.0+4.0*TIME)
C
      RETURN
      END
```

```
       MIXING TANK PRCBLEM

SALT 2 LB/GAL
   5 GAL/MIN -------+
                    I
                I   I         I
                I   I         I
                I             I
                I-------------I
                I  VC=1CCGAL  I
                I      C      I-----+
                ---------------     I
                                    I
                                 C LB/GAL
                                 1 GAL/MIN
```

RESULT

TIME (MIN)	CONC (LE/GAL)
0.	C.
1.0	C.09569
2.0	C.18344
3.0	C.26417
4.0	C.33866
5.0	0.4C760
6.0	0.47154
7.0	0.531C1
8.0	C.58645
9.0	0.63822
10.0	0.68668
11.0	0.73212
12.0	C.77481
13.0	C.81498
14.0	C.85284
15.0	C.88858
16.0	C.92236
17.0	C.95433
18.0	C.98464
19.0	1.01340
20.0	1.04C73
21.0	1.06673
22.0	1.09148
23.0	1.115C8
24.0	1.13760
25.0	1.15910
26.0	1.17966
27.0	1.19933
28.0	1.21817
29.0	1.23623
30.0	1.25355
31.0	1.27017
32.0	1.28614
33.0	1.3C149

34.0	1.31626
35.0	1.33048
36.0	1.34417
37.0	1.35736
38.0	1.37009
39.0	1.38237
40.0	1.39422
41.0	1.40567
42.0	1.41674
43.0	1.42744
44.0	1.43780
45.0	1.44782
46.0	1.45752
47.0	1.46692
48.0	1.47604
49.0	1.48487
50.0	1.49344
51.0	1.50176
52.0	1.50984
53.0	1.51768
54.0	1.52530
55.0	1.53270
56.0	1.53990
57.0	1.54691
58.0	1.55372
59.0	1.56035
60.0	1.56681
61.0	1.57309
62.0	1.57922
63.0	1.58519
64.0	1.59100
65.0	1.59668
66.0	1.60221
67.0	1.60761
68.0	1.61287
69.0	1.61801
70.0	1.62303
71.0	1.62794
72.0	1.63272
73.0	1.63740
74.0	1.64198
75.0	1.64645
76.0	1.65082
77.0	1.65509
78.0	1.65927
79.0	1.66336
80.0	1.66736
81.0	1.67128
82.0	1.67512
83.0	1.67887
84.0	1.68255
85.0	1.68615
86.0	1.68968
87.0	1.69314
88.0	1.69653
89.0	1.69986
90.0	1.70312
91.0	1.70631
92.0	1.70945
93.0	1.71252

94.0	1.71554
95.0	1.71850
96.0	1.72140
97.0	1.72426
98.0	1.72705
99.0	1.72980
100.0	1.73250
101.0	1.73515
102.0	1.73776
103.0	1.74032
104.0	1.74283
105.0	1.74530
106.0	1.74773
107.0	1.75011
108.0	1.75246
109.0	1.75477
110.0	1.75704
111.0	1.75927
112.0	1.76146
113.0	1.76362
114.0	1.76574
115.0	1.76783
116.0	1.76989
117.0	1.77191
118.0	1.77391
119.0	1.77587
120.0	1.77780
121.0	1.77970
122.0	1.78157
123.0	1.78341
124.0	1.78523
125.0	1.78702
126.0	1.78878
127.0	1.79051
128.0	1.79222
129.0	1.79391
130.0	1.79557
131.0	1.79721
132.0	1.79882
133.0	1.80041
134.0	1.80198
135.0	1.80352
136.0	1.80505
137.0	1.80655
138.0	1.80803
139.0	1.80950
140.0	1.81094
141.0	1.81236
142.0	1.81376
143.0	1.81515
144.0	1.81651
145.0	1.81786
146.0	1.81919
147.0	1.82051
148.0	1.82180
149.0	1.82308
150.0	1.82434
151.0	1.82559
152.0	1.82682
153.0	1.82804

154.0	1.82924
155.0	1.83042
156.0	1.83159
157.0	1.83275
158.0	1.83389
159.0	1.83502
160.0	1.83613
161.0	1.83723
162.0	1.83832
163.0	1.83939
164.0	1.84045
165.0	1.84150
166.0	1.84254
167.0	1.84356
168.0	1.84458
169.0	1.84558
170.0	1.84657
171.0	1.84754
172.0	1.84851
173.0	1.84947
174.0	1.85041
175.0	1.85135
176.0	1.85227
177.0	1.85318
178.0	1.85409
179.0	1.85498
180.0	1.85586

References:

1) Hamming, R. W., "Numerical Methods for Scientists and Engineers", McGraw-Hill (1962)

2) Hildebrand, F. B., "Introduction to Numerical Analysis", McGraw-Hill (1956)

3) Himmelblau, D. M., "Basic Principles and Calculations in Chemical Engineering", Prentice-Hall (1964)

4) Lapidus, L., "Digital Computation for Chemical Engineers", McGraw-Hill (1962)

5) Mickley, H. S., T. K. Sherwood, and C. E. Reed, "Applied Mathematics in Chemical Engineering" McGraw-Hill (1957)

6) Perry, J. H. ed., "Chemical Engineers' Handbook" 3rd edition McGraw-Hill (1950)

7) Ralston, A., and H. S. Wilf, "Mathematical Methods for digital Computers", Wiley (1962)

8) Richtmyer, R. D., "Difference Methods for Initial-Value Problems", Interscience (1962)