ORNL/TM-5457

Cy 69

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0284734 7

A Computer Model for the KALC Process Studies in the ORGDP Off-Gas Decontamination Pilot Plant

R. W. Glass
R. E. Barker

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
DOCUMENT COLLECTION
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this document, send in name with document and the library will arrange a loan.
UCN-7969
(3 3-67)

OAK RIDGE NATIONAL LABORATORY
OPERATED BY UNION CARBIDE CORPORATION FOR THE ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
Price: Printed Copy $4.50; Microfiche $2.25

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration/United States Nuclear Regulatory Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

ORNL/TM-5457
Dist. Category UC-77

Contract No. W-7405-eng-26

CHEMICAL TECHNOLOGY DIVISION

THORIUM UTILIZATION PROGRAM

Reprocessing Development - Task 200

A COMPUTER MODEL FOR THE KALC PROCESS STUDIES IN THE ORGDP OFF-GAS DECONTAMINATION PILOT PLANT

R. W. Glass and R. E. Barker

SEPTEMBER 1976

NOTICE This document contains information of a preliminary nature and was prepared primarily for internal use at the Oak Ridge National Laboratory. It is subject to revision or correction and therefore does not represent a final report.

OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37830
operated by
UNION CARBIDE CORPORATION
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0284734 7

TABLE OF CONTENTS

A COMPUTER MODEL FOR THE KALC PROCESS STUDIES IN THE ORGDP OFF-GAS DECONTAMINATION PILOT PLANT

R. W. Glass and R. E. Barker

ABSTRACT

A computer model of the KALC process is presented for the equipment configuration in use during HTGR off-gas studies at the ORGDP Off-Gas Decontamination Pilot Plant. The model is tailored to require input routinely available during such experimental studies. A program is included to provide McCabe-Thiele plots as an additional convenience.

---

1. INTRODUCTION

Mullins and Glass[1] have described a generic multicomponent, multicolumn, multistage equilibrium model for the KALC process. The present work has tailored the general model to the system configuration in use during the HTGR off-gas studies at the ORGDP Off-Gas Decontamination Pilot Plant.[2,3] Flexibility of the general program has been maintained in all of the important parameters, and the data input has been revised to allow fewer numbers to be included. The revised input is also in a form more convenient to engineers concerned with the operational and analytical efforts.

As an aid to the KALC/ORGDP program, a plotting package has been developed to utilize the output of the program and to generate plots of the McCabe-Thiele type for krypton and oxygen in each of the three columns (i.e., the absorber, the fractionator, and the stripper).

2. THE KALC/ORGDP PROGRAM

The KALC/ORGDP program is basically the same as the general program presented by Mullins and Glass.[1] However, it has two notable features: (1) a matrix algebra technique as described by Tierney and co-workers[4,5] is used for stage material and energy balances; and (2) a modified Redlich-Kwong[6] model is used for the vapor-liquid equilibria of the CO_2/Xe-O_2-CO-N_2-Kr system. The plant configuration treated by the KALC/ORGDP program is shown in Fig. 1.

2.1 Program Inputs

Input to the KALC/ORGDP program has been minimized as much as possible while maintaining flexibility. The required information is described in the paragraphs that follow.

Stage numbers. The total number of stages to be considered (N), the number of stages in the absorber (NA), the number of stages above the fractionator feed stage, excluding the fractionator condenser (NFR), the number of stages below the fractionator feed stage, excluding the fractionator reboiler (NFS), and the number of stages above the stripper feed stage, excluding the stripper condensers (NSR), must be specified. For the normal KALC/ORGDP configuration, NFR = 0 and NSR = 1.

Component specification. The number of components to be considered in the fluid phases must be specified as 6. The components are identified as follows:

(1) carbon dioxide,

(2) xenon,

ORNL-DWG 76-387

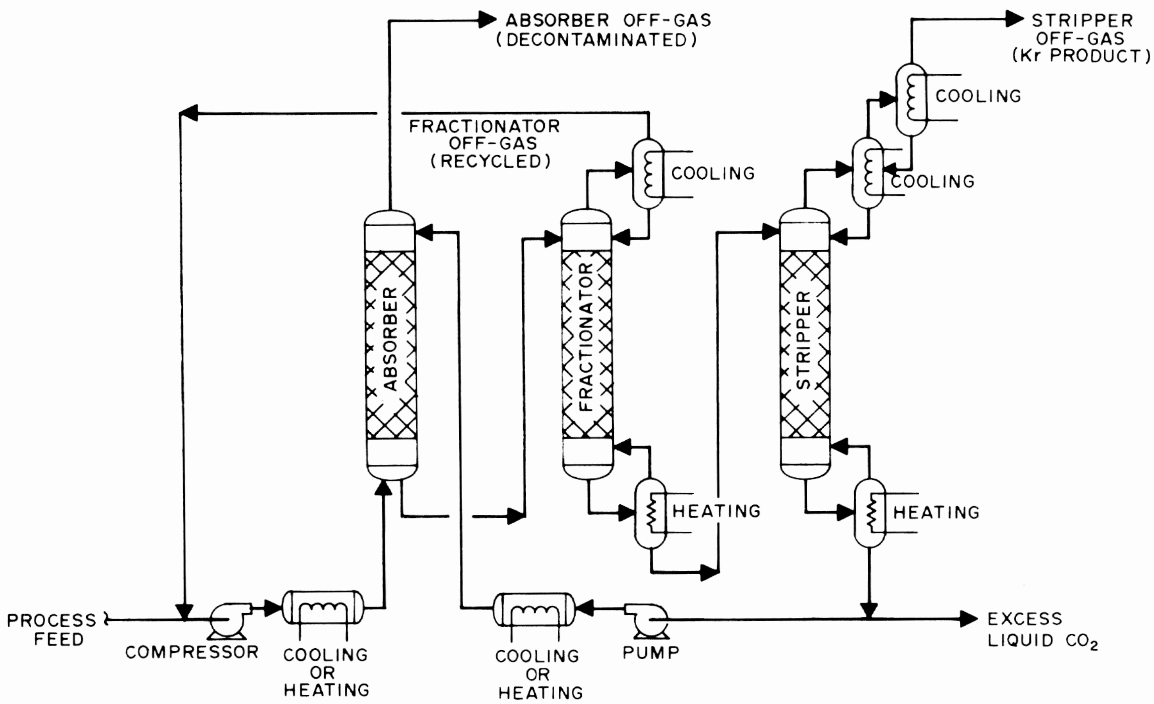


Fig. 1. Equipment configuration for computer model.

(3) oxygen,

(4) carbon monoxide,

(5) nitrogen,

(6) kryton.

Process feed gas. The composition of the process feed gas must be supplied in the form of mole fractions of components identified as above, beginning with xenon and ending with krypton. The mole fractions of oxygen and krypton must be nonzero since various output items require the presence of these two components.

System temperatures. Reasonable estimates of the following system temperatures are required (in °C):

(1) feed gas,

(2) absorber scrub liquid,

(3) absorber top stage,

(4) absorber bottom stage,

(5) fractionator condenser,

(6) fractionator feed (top) stage,

(7) fractionator reboiler stage,

(8) final stripper condenser stage,

(9) main stripper condenser stage,

(10) stripper reboiler stage.

The initial estimates given for temperatures (1), (2), (7), and (10) are important for the program calculation of the following items:

(1) absorber feed enthalpy,

(2) change in the fractionator recycle gas enthalpy,

(3) fractionator reboiler vapor rate,

(4) stripper reboiler vapor rate,

(5) scrub liquid cooling load.

Items (1), (2), and (5) are calculated and remain fixed throughout the program calculation; these heat quantities, rather than the temperatures *per se*, are treated by the program as constants.

System pressures. Although the program could accept different pressures for each stage, only the top and bottom stage pressures for each of the three columns are required [in atm (abs)]. The program linearly interpolates between the respective top and bottom stage pressures and apportions the results to the intermediate stages.

Recycle gas composition. In order to properly evaluate the recycle gas heat, the recycle gas composition for components 2,...6 must be specified as mole fractions. Zero values are permitted for any or all of these mole fractions.

Reboiler heats. Fractionator and stripper heating loads (in Btu/hr, positive for heat input) are required as inputs.

Condenser loads. Condenser loads are not required as inputs. The fractionator and final stripper condenser loads are effectively specified by fixing the respective off-gas rates (see subsection below). The main condenser load is specified by inputting a fraction of the stripper reboiler heat value to be used. Thus a fraction of 0.5 or 1.5 would indicate to the program that -50% or -150% of the stripper reboiler heat specified is to be assigned to the main condenser stage.

Flow rates. The flow rates of the following streams must be supplied as input to the program (in equivalent scfm, 70°F, 1 atm, real gas):

(1) process feed gas,

(2) absorber off-gas,

(3) fractionator recycle gas,

(4) stripper off-gas,

(5) scrub liquid.

The program treats flows (1), (3), (4), and (5) as constants throughout its calculation.

Order for inputting data. Appendix A is a listing of the KALC/ORGDP programs. A sample of the required input is given in the "data" portion of Appendix A. (Note that the data are input in free format, with the various items separated with a space.) The input is identified as follows:

Card 1: 42 6

Total number of stages = 42

Number of fluid components = 6

Card 2: 8 0 16 1

Number of stages in absorber = 8

Number of stages in fractionator above feed stage (excluding condenser) = 0

Number of stages in fractionator below feed stage (excluding reboiler) = 16

Number of stages below main stripper condenser on which stripper feed is introduced = 1

Card 3: 18.89 -27.78 -21 -22 -38 -24 -19 -24 -23 -22.78

Temperature of absorber feed gas as it enters the absorber = 18.89°C

Temperature of scrub liquid recycle as it enters the absorber = -27.78°C

Temperature of the topmost absorber packing = -21°C

Temperature of the bottommost absorber packing = -22°C

Temperature of the fractionator condenser = -38°C

Temperature of the fractionator feed stage = -24°C

Temperature of the fractionator reboiler = -19°C

Temperature of the final stripper condenser = -24°C

Temperature of the main stripper condenser = -23°C

Temperature of the stripper reboiler = -22.78°C.

Card 4: 22 22 20 20 18 18

Pressure at the top of the absorber column = 22 atm (abs)

Pressure at the bottom of the absorber column = 22 atm (abs)

Pressure at the top of the fractionator column = 20 atm (abs)

Pressure at the bottom of the fractionator column = 20 atm (abs)

Pressure at the top of the stripper column = 18 atm (abs)

Pressure at the bottom of the stripper column = 18 atm (abs)

Card 5: 1.0E-12 8.0E-2 1.0E-12 1.0E-12 1.0E-7

Feed gas composition (mole fractions):

Xe = 1.0E-12

O_2 = 8.0E-2

CO = 1.0E-12

N_2 = 1.0E-12

Kr = 1.0E-7

Card 6: 0.0 0.265 0.0 0.0 0.0

Fractionator recycle gas composition (mole fractions):

Xe = 0.0

O_2 = 0.265

CO = 0.0

N_2 = 0.0

Kr = 0.0

Card 7: 10300 16000 1.0

Heat input of fractionator reboiler = 10300 Btu/hr

Heat input of stripper reboiler = 16000 Btu/hr

Fraction of stripper reboiler heat value to be assumed by main stripper condenser = 1.0

This means that the program is to assume a constant value of (-1.0 * 16000 Btu/hr) as the main condenser duty.

Card 8: 6.6 3.2 2.4 0.061 125

Process feed gas flow rate = 6.6 scfm*

Absorber off-gas flow rate = 3.2 scfm*

Fractionator off-gas flow rate = 2.4 scfm*

Stripper off-gas flow rate = 0.061 scfm*

Scrub liquid flow rate = 125 scfm* (equivalent)

*Conditions: 70°F, 1 atm, real gas.

2.2 Model Calculations

Based on the data that are input to the program, certain initial calculations are performed to "fill up" the various program matrices for flow, temperatures, enthalpies, etc. Some of these data only need to be reasonable estimates, whereas other input (to be discussed shortly) should be carefully chosen.

Flow rate values that are input to the program, with the exception of the absorber off-gas flow rate, are treated as constants by the program. Thus the specification of the fractionator and final stripper condenser off-gas rates will result in condenser loads necessary to effect these rates.

Reboiler heat loads input are also assumed to be constants by the program. The effect of a scrub liquid cooler is accounted for by initial calculations using the scrub liquid temperature (at the point of entry into the absorber), the stripper reboiler temperature, and the recycle liquid flow rate. Basically, the program calculates the necessary enthalpy change required to heat (or cool) the specified amount of recycle liquid CO_2 from the temperature of the stripper reboiler to the temperature of the liquid entering the absorber. Once this enthalpy change is calculated from the initial input data, it is considered to be a constant and represents a processing condition.

Similarly, the net enthalpy change required for the fractionator off-gas as it is recycled to the absorber column is calculated from the values that are input for the fractionator off-gas composition, flow rate, temperature, etc.

Other temperatures and pressures are supplied to establish initial column profiles. Stage temperatures are variables in the computation, but stage pressures are constants.

2.3 Program Output

Once initial calculations have been performed, the KALC/ORGDP program proceeds to the iterative stage calculations. If reasonable and consistent data have been used as input, the stage program will converge in approximately seven to nine iterations. Convergence is based on both material and energy balance criteria.

The printed output consists of the following terms:

(1) equilibrium constants, parameters, etc., used by the program;

(2) a table of initial feeds and heats for each stage;

(3) a series of iteration tables indicating the values of temperature, flows, etc., for all stages;

(4) a table of feeds and heats, upon convergence;

(5) matrices of liquid and vapor streams, upon convergence;

(6) a table of stage temperatures, flows, enthalpies, and pressures, upon convergence;

(7) a table of vapor mole fractions for each component leaving each stage, upon convergence;

(8) a table of liquid mole fractions for each component leaving each stage, upon convergence;

(9) a table of y/x values for each component on each stage, upon convergence;

(10) a table of interstage L/V ratios, upon convergence;

(11) a table of interstage y/x ratios, upon convergence;

(12) a table of interstage absorption factors (Lx/Vy), upon convergence;

(13) a table of information concerning the general system, such as decontamination factors, concentration factors, etc., upon convergence.

The following notes are pertinent to the table that provides information for the general system:

(1) All heat loads are output in units of calories per 100 g-moles of process feed.

(2) The Krypton Process Decontamination Factor (PDF) is defined as the total amount of krypton entering in the feed stream, divided by the total krypton leaving the process (excluding the krypton released in the stripper off-gas).

(3) The Krypton Column Decontamination Factor (CKRDF) is defined as the total amount of krypton entering the absorber in the absorber feed gas stream, divided by the total krypton leaving the absorber in the absorber off-gas.

(4) The Krypton Process Concentration Factor (CFKR) is defined as the mole fraction of krypton in the stripper off-gas, divided by the mole fraction of krypton in the process feed gas.

(5) The Krypton Concentration Factor Relative to the Light Gas (CFKRLG) is defined as the ratio of the mole fraction of krypton in the stripper off-gas, divided by the sum of the mole fractions of all light gases present in the stripper off-gas, to the mole

fraction of krypton in the process feed gas, divided by the sum of the mole fractions of all light gases present in the process feed gas.

(6) The Fraction of Krypton Recovered (FLGR) is defined as the amount of krypton in the stripper off-gas, divided by the amount of krypton in the process feed gas.

(7) The Fraction of Krypton Vented (FKRV) is defined as the amount of krypton in the absorber off-gas, divided by the amount of krypton in the process feed gas.

(8) The Fraction of Krypton Bypassed (FKRB) is defined as the amount of krypton leaving the stripper reboiler as a liquid but not recycled to the absorber (i.e., present in the excess liquid CO_2), divided by the total amount of krypton present in the process feed gas.

(9) The Fraction of Light Gas Recovered (FLGR) is defined as the total amount of light gas present in the stripper off-gas, divided by the total amount of light gas present in the process feed gas.

(10) The Krypton Recycle Ratio (RKRR) is defined as the amount of krypton in the fractionator off-gas, divided by the total amount of krypton in the process feed gas.

(11) The Krypton Capture-to-Loss Ratio (CTLRKR) is defined as the krypton in the stripper off-gas, divided by the sum of the krypton present in the absorber off-gas plus that present in the excess CO_2 liquid.

(12) The Light Gas Recycle Ratio (RLGR) is defined as the total light gas in the fractionator off-gas, divided by the total light gas present in the process feed gas.

(13) The Light Gas Capture-to-Loss Ratio (CTLRLG) is defined as the total light gas present in the stripper off-gas, divided by the sum of the light gas in the absorber off gas plus that present in the excess liquid CO_2.

2.4 Performance Parameters Associated with Figure 2

The process performance parameters with reference to Fig. 2 are as follows:

Process DF for krypton (PDF)

$$\mathrm{PDF} \equiv \frac{F y_{Kr}^{F}}{A y_{Kr}^{A} + B x_{Kr}^{B}} .$$

Column DF for krypton (CKRDF)

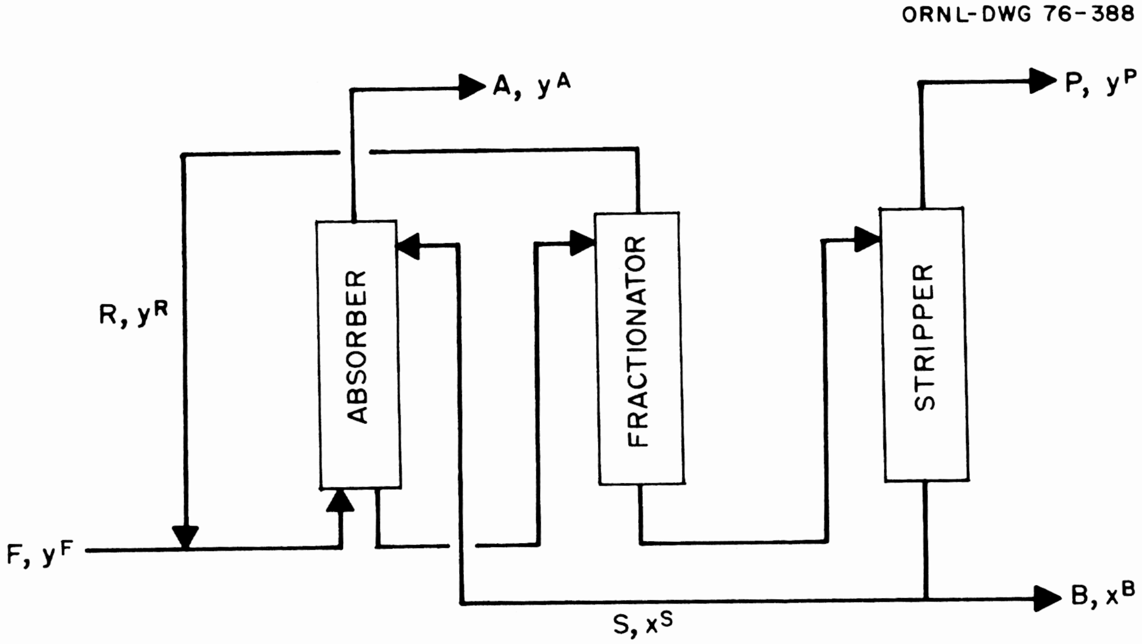
$$\mathrm{CKRDF} \equiv \frac{F y_{Kr}^{F} + R y_{Kr}^{R}}{A y_{Kr}^{A}} .$$

Krypton decontamination factor (CFKR)

$$\mathrm{CFKR} \equiv \frac{y_{Kr}^{P}}{y_{Kr}^{F}} .$$

Krypton-to-light gas concentration factor (CFKRLG)

$$\mathrm{CFKRLG} \equiv \frac{y_{Kr}^{P}}{y_{Kr}^{F}} \; \frac{\Sigma\, y_{LG}^{F}}{\Sigma\, y_{LG}^{P}} .$$



NOTES:

F = PROCESS FEED GAS RATE
A = ABSORBER OFF-GAS RATE
R = FRACTIONATOR OFF-GAS RATE
P = STRIPPER OFF-GAS RATE
S = SCRUB LIQUID RATE
B = EXCESS SCRUB LIQUID RATE
x = LIQUID MOLE FRACTIONS
y = VAPOR MOLE FRACTIONS

ASSUMPTIONS:

$$x^S = x^B$$

$$F = A + P + B$$

$$Fy^F = Ay^A + Py^P + Bx^B$$

Fig. 2. Identification of streams for performance parameters.

Fraction of light gas recovered (FLGR)

$$\mathrm{FLGR} \equiv \frac{P\Sigma y^{P}_{LG}}{F\Sigma y^{F}_{LG}} \ .$$

Fraction of krypton vented (FKRV)

$$\mathrm{FKRV} \equiv \frac{Ay^{A}_{Kr}}{Fy^{F}_{Kr}} \ .$$

Fraction of krypton bypassed (FKRB)

$$\mathrm{FKRB} \equiv \frac{Bx^{B}_{Kr}}{Fy^{F}_{Kr}} \ .$$

Fraction of light gas recovered (FLGR)

$$\mathrm{FLGR} \equiv \frac{P\Sigma y^{P}_{LG}}{F\Sigma y^{F}_{LG}} \ .$$

Krypton recycle ratio (RKRR)

$$\mathrm{RKRR} \equiv \frac{Ry^{R}_{Kr}}{Fy^{F}_{Kr}} \ .$$

Krypton capture-to-loss ratio (CTLRKR)

$$\mathrm{CTLRKR} \equiv \frac{Py^{P}_{Kr}}{Ay^{A}_{Kr} + Bx^{B}_{Kr}} \ .$$

Light gas recycle ratio (RLGR)

$$\mathrm{RLGR} \equiv \frac{R\Sigma y^{R}_{LG}}{F\Sigma y^{F}_{LG}} \ .$$

Light gas capture-to-loss ratio (CTLRLG)

$$\mathrm{CTLRLG} \equiv \frac{P\Sigma y^{P}_{LG}}{A\Sigma y^{A}_{LG} + B\Sigma x^{B}_{LG}} \ .$$

2.5 McCabe-Thiele Plots

In addition to the printed output described above, the KALC/ORGDP program also returns a deck of punched cards which may be used to construct McCabe-Thiele plots of krypton and oxygen for each of the three process columns. The plotting programs, complete with a typical data set, are listed in Appendix B. No data input other than the punched deck provided by the KALC/ORGDP program is necessary. Plots provided by the plotting routine listed in Appendix B are shown as Figs. 3-8. All of these are log-log plots except for oxygen in the absorber. Feed stages are indicated by a dashed line from the point on the 45° line which represents the feed composition to that stage. Three pieces of information are included on each plot to provide ready identification:

(1) The IBM-360 job notation is printed in the upper left corner (RWGK25 in Figs. 3-8).

(2) The column designation is printed in the upper right corner (SBS, FRC, and STR for absorber, fractionator, and stripper, respectively).

(3) The component name is printed in the lower right corner (KR and O2 for krypton and oxygen, respectively).

2.6 Concluding Notes

The most important single fact to be noted is that the KALC/ORGDP program represents a model and, as such, has the usual caveats associated with it. Although many program improvements are possible, these will be developed more effectively as a result of individual needs. The current version of the KALC/ORGDP program is written in the IBM 360/91

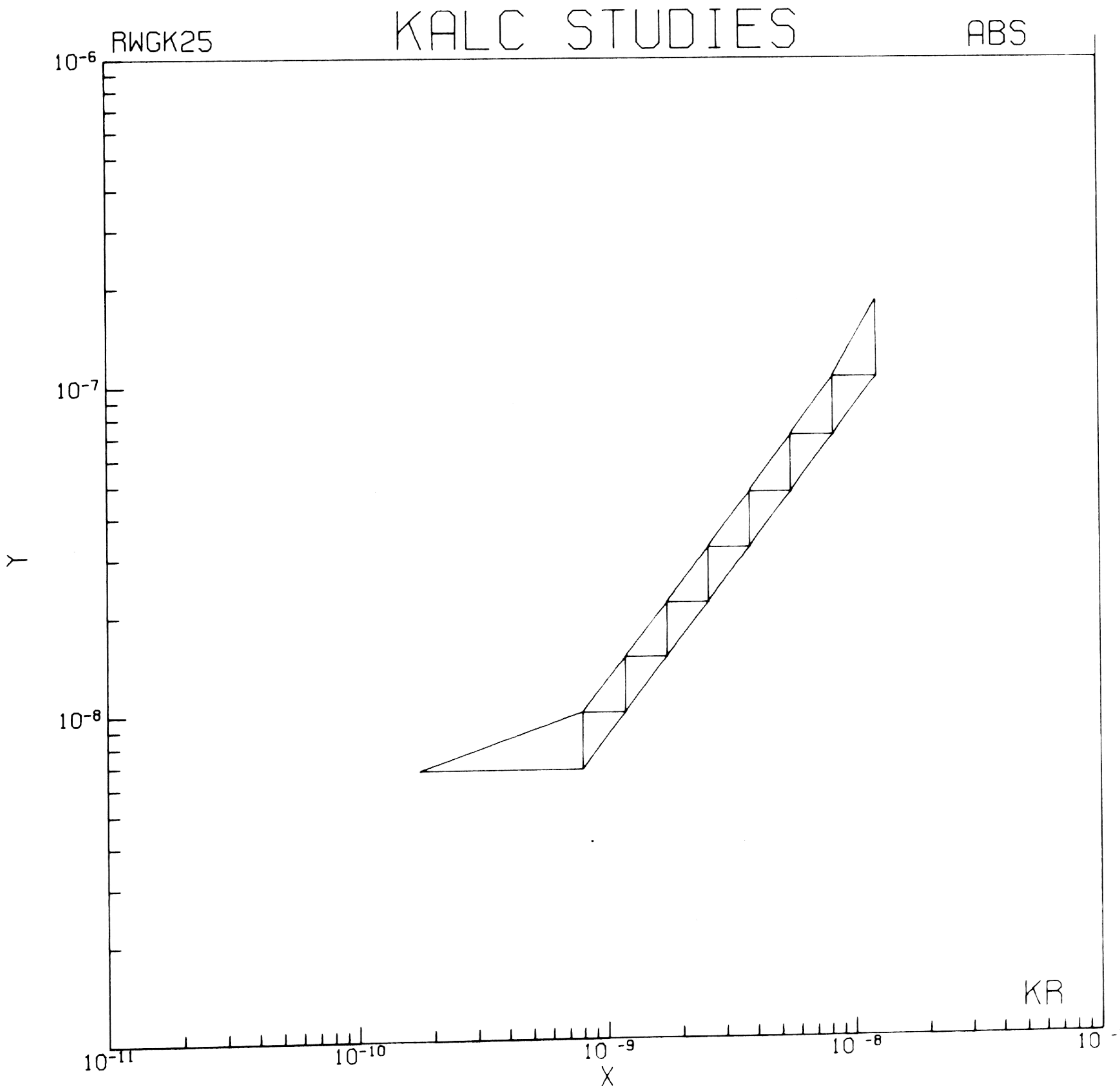


Fig. 3. McCabe-Thiele plot for krypton in the absorber (run RWGK25).

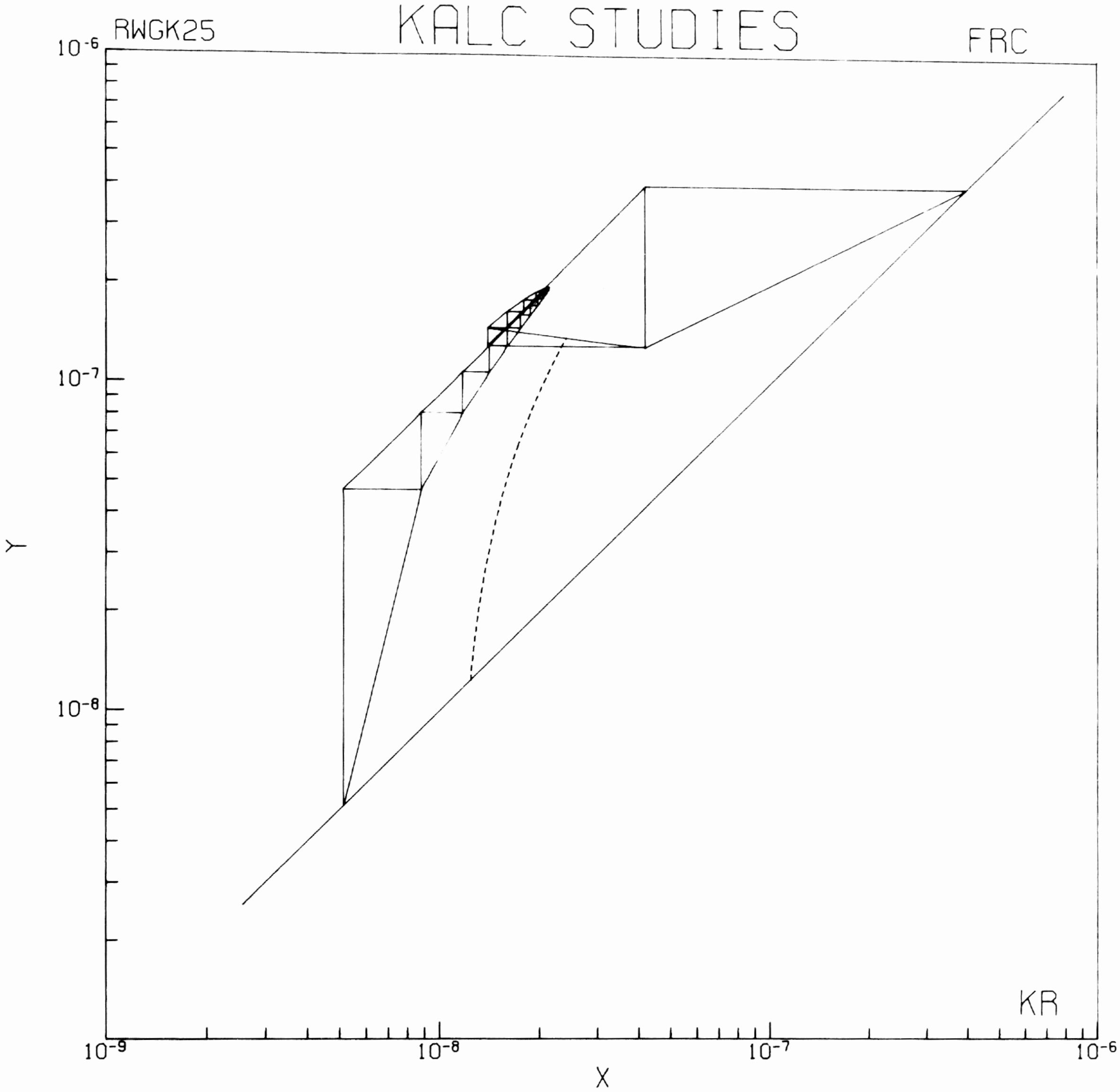


Fig. 4. McCabe-Thiele plot for krypton in the fractionator (run RWGK25).

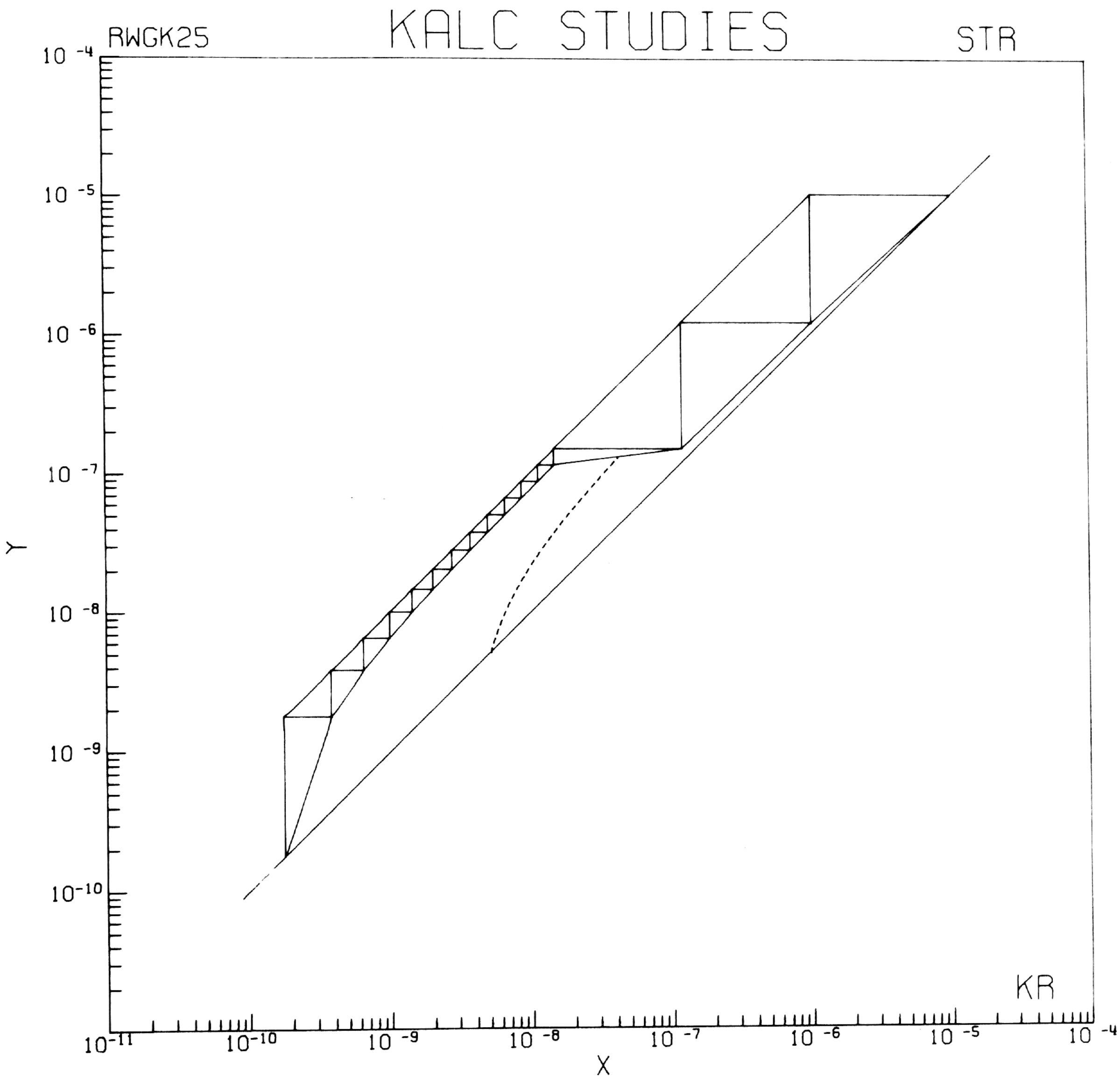


Fig. 5. McCabe-Thiele plot for krypton in the stripper (run RWGK25).

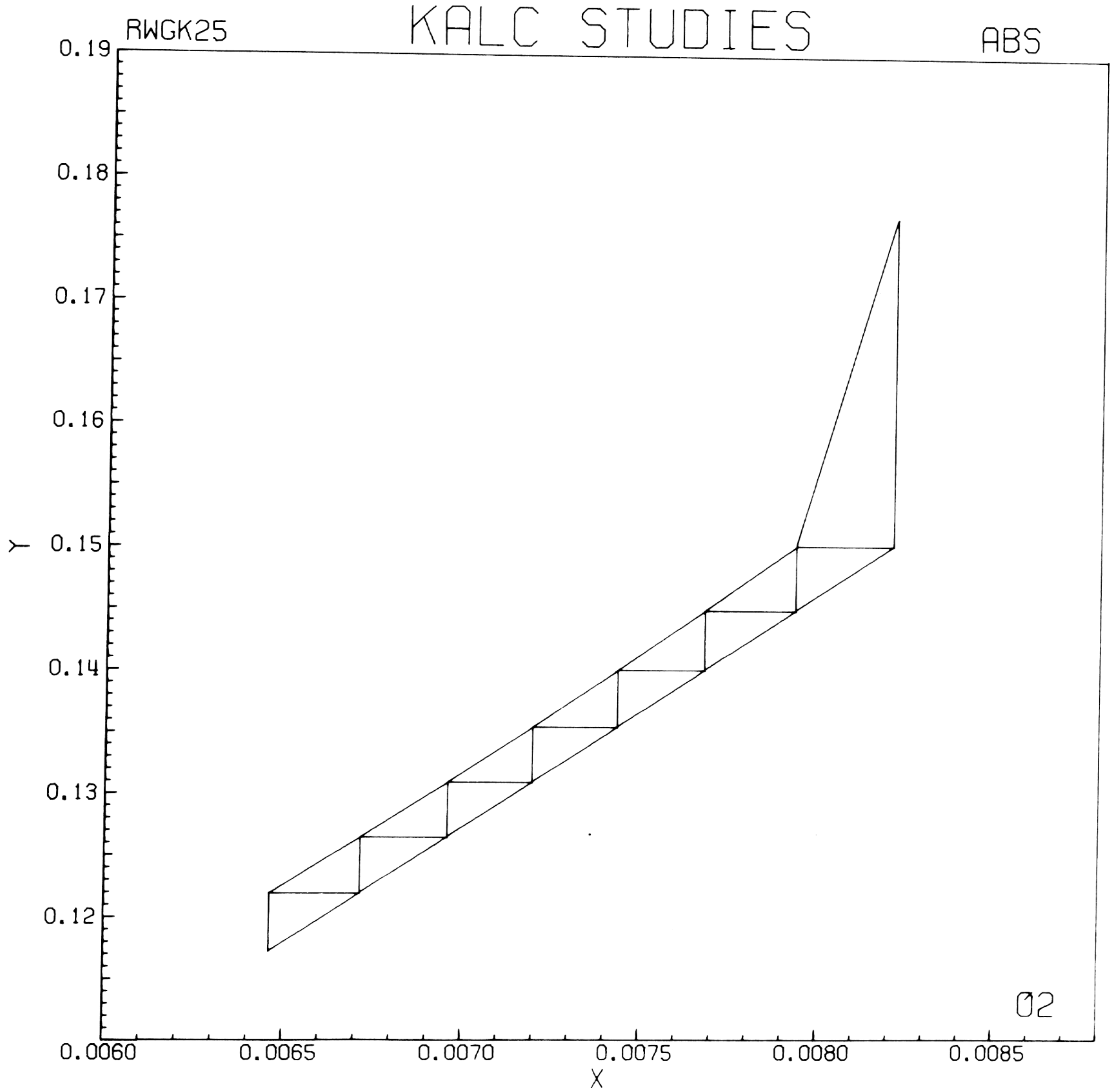


Fig. 6. McCabe-Thiele plot for oxygen in the absorber (run RWGK25).

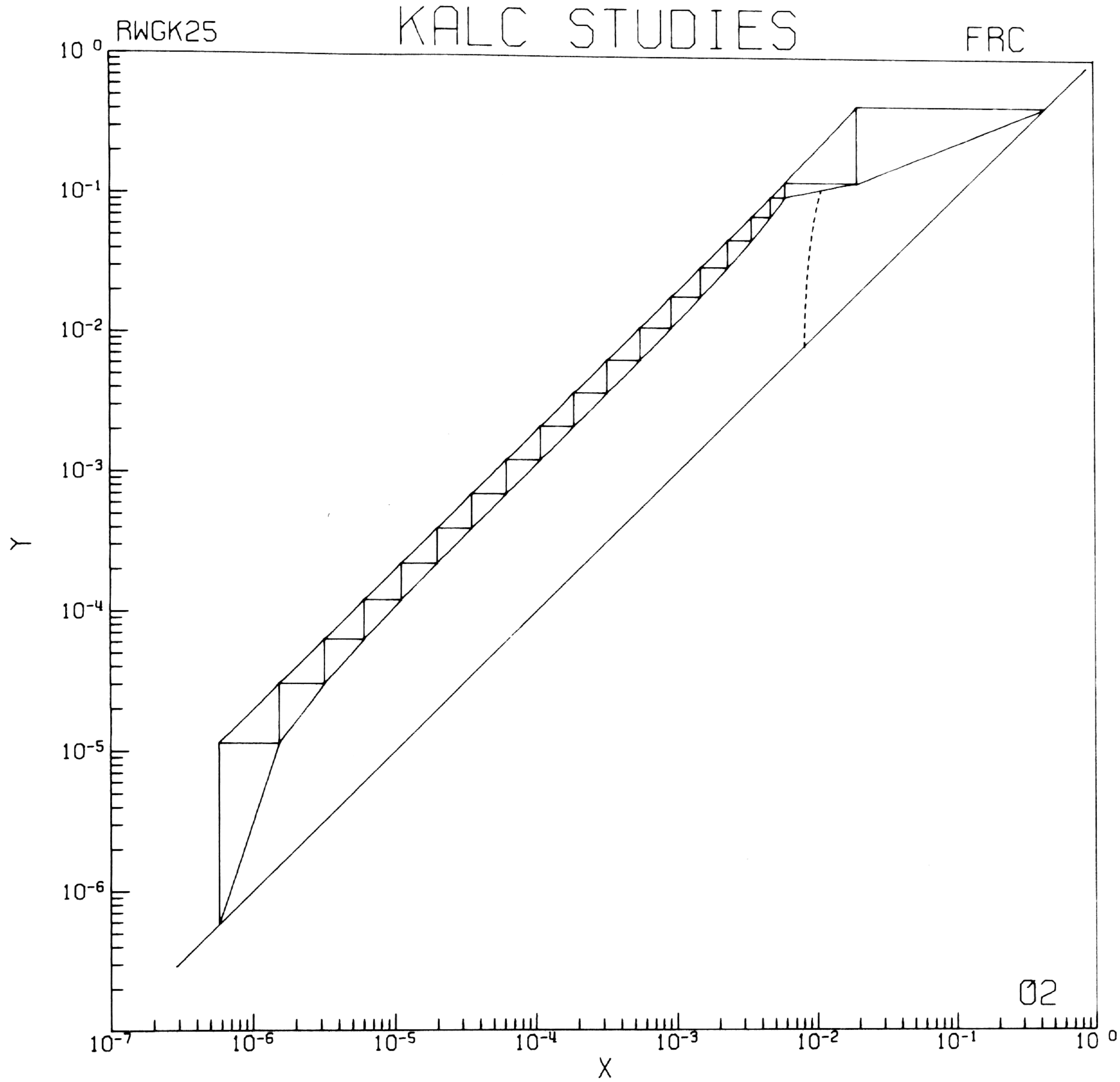


Fig. 7. McCabe-Thiele plot for oxygen in the fractionator (run RWGK25).

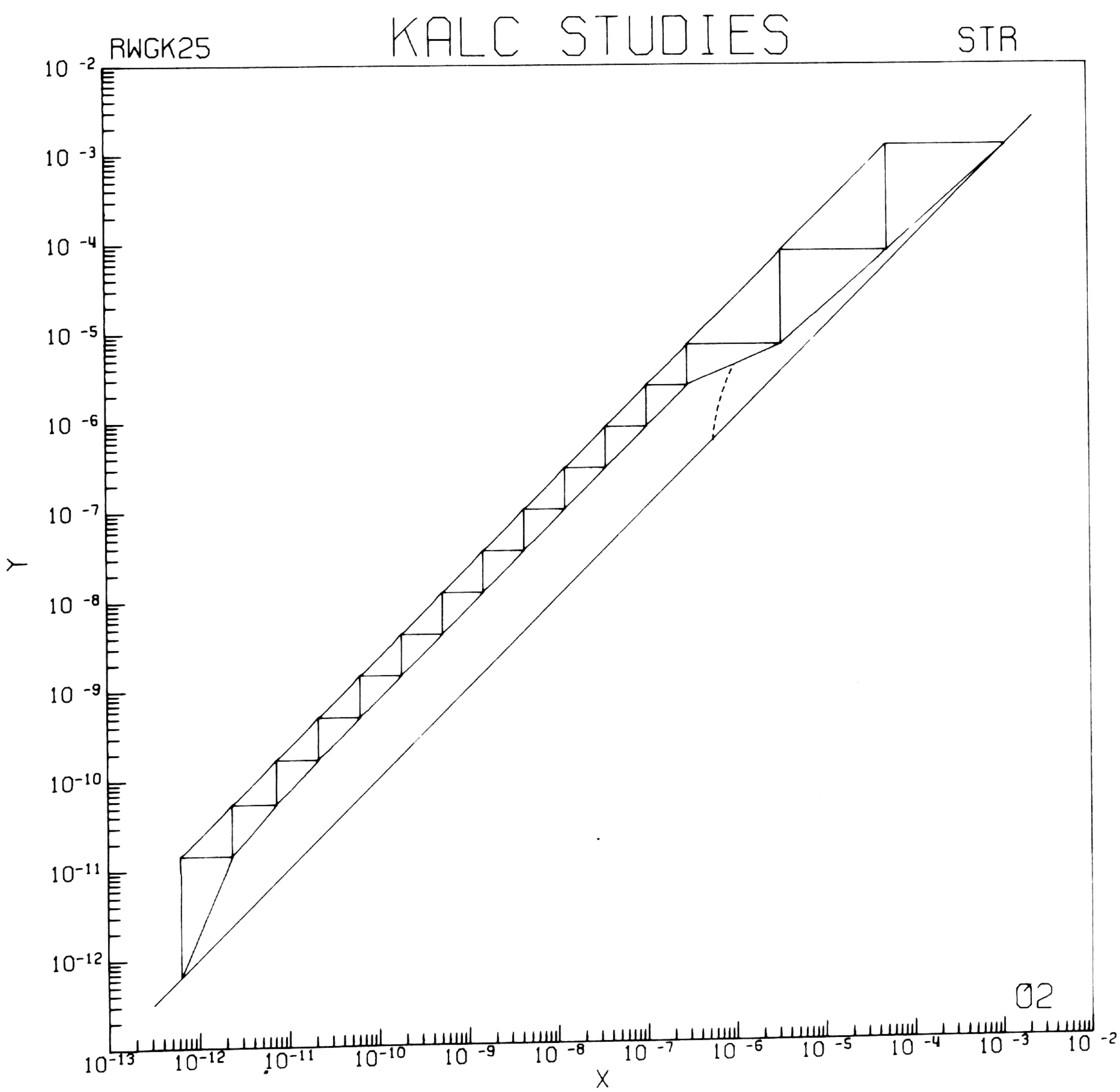


Fig. 8. McCabe-Thiele plot for oxygen in the stripper (run RWGK25).

(ORNL) PL/I language, PL/I optimizing compiler, version 1 R2.2 PT F60. The example program presented in Appendix A requires approximately 600 K storage (42 stages) and 16 min (nine iterations) to execute. Present convergence criteria are $ADE < 10^{+3}$ and $ADM < 10^{-4}$ for energy and material balances respectively. ADE is the sum of the absolute values of energy errors for all components on all stages. ADM is the sum of the absolute values of material errors for all components on all stages. ADE and ADM criteria must be met simultaneously for convergence.

Simple adjustments to the program might include a flooding curve correlation[7] for the packed columns, distributed condenser loads, additional process performance criteria, and so forth. Improved equilibria parameters are currently being evaluated and will be available for inclusion in the KALC/ORGDP program.

3. REFERENCES

1. J. C. Mullins and R. W. Glass, An Equilibrium Stage Model of the KALC Process, ORNL/TM-5099 (in press).

2. M. J. Stephenson and D. I. Dunthorn, System Design Description of the ORGDP LMFBR-HTGR Fuel Reprocessing Off-Gas Decontamination Plant, ORGDP-K-GD-873 (Sept. 7, 1973).

3. A. D. Ryon, KALC Campaign II at the ORGDP Off-Gas Pilot Plant, ORNL/TM-5298 (in press).

4. J. W. Tierney and J. A. Bruno, "Equilibrium Stage Calculations," A.I.Ch.E. J. 13, 556-63 (1967).

5. J. W. Tierney and J. L. Yanosik, "Simultaneous Flow and Temperature Correction in the Equilibrium Stage Problem," A.I.Ch.E. J. 15, 897-901 (1969).

6. J. M. Prausnitz, Molecular Thermodynamics of Fluid-Phase Equilibria, Prentice-Hall, Englewood Cliffs, N.J., 1969.

7. R. W. Glass et al., Krypton Absorption in Liquid CO_2 (KALC): Campaign II in the Experimental Engineering Section Off-Gas Decontamination Facility, ORNL/TM-5095 (February 1976).

APPENDIX A:

THE KALC/ORGDP PROGRAM

```
//XXXXXX JOB (XXXXX),'BIN X XXXXX',MSGLEVEL=1
/*ROUTE  PRINT LOCAL
//*CLASS CPU91=20M,R=600K,IO=16
/*OPTIONS FORMS=4000
/*ROUTE  XEQ   CPU1
// EXEC PLIXCLG,PARM.PLI='NEST',PARM.LKED='NOXREF,NOLIST',
// REGION.GO=600K,PARM.GO='REPORT'
//PLI.SYSLIN DD SPACE=(80,(1700,100))
//PLI.SYSIN DD *
             /* KALC COMPUTER STUDIES */
 STAGE:PROC OPTIONS(MAIN);
 SETUP:  PROC(N,M) ;
         /* SETUP FOR GENERAL KALC PROCESS           */
         /* NA = NO. OF STAGES IN ABS                */
         /* NA+1 = FRC CONDENSER                     */
         /* NFR = NO. OF STAGES IN FRC TOP           */
         /* NA+NFR+2 = FRC FEED PLATE                */
         /* NFS = NO. OF STAGES IN FRC BOTTOM        */
         /* NF = FRC REBOILER                        */
         /* NF+1 = STR FINAL CONDENSER               */
         /* NF+2 =  "  MAIN CONDENSER                */
         /* NF+NSR+2 = STR FEED PLATE (NSR>0)        */
         /* N = STR REBOILER = TOTAL NO. OF STAGES */
         F = 0;  Q,QF = 0;  A,B,VS,VP,LS,LP = 0;  YPFED,YRECY = 0;
         GET LIST(NA,NFR,NFS,NSR);  NF=NA+NFR+NFS+3;
         GET LIST(FGAST,SLTC,TC(1),TC(NA),TC(NA+1),TC(NA+2),
             TC(NF),TC(NF+1),TC(NF+2),TC(N));
         GET LIST(P(1),P(NA),P(NA+1),P(NF),P(NF+1),P(N));
         GET LIST((YPFED(J) DO J=2 TO M));  YPFED(1)=1-SUM(YPFED);
         GET LIST((YRECY(J) DO J=2 TO M));  YRECY(1)=1-SUM(YRECY);
         DO J=1 TO M;  F(NA,J)=100*YPFED(J);  END;
         GET LIST(QNFR,QN,XQN);
         GET LIST(FDRAT,ARATE,FRATE,SRATE,SOLRT);
         VS(1,1)=(100*ARATE)/FDRAT;
         VP(NA+1,NA+1)=-(100*FRATE)/FDRAT;
         VP(NF+1,NF+1)=-(100*SRATE)/FDRAT;
         LP(N,N)=-(100*SOLRT)/FDRAT;
         Q(NF,1)=(QNFR*355.94)/FDRAT;
         Q(NA+1,1)=-0.7*Q(NF,1);
         Q(N,1)=(QN*355.94)/FDRAT;
         Q(NF+2,1)=-XQN*Q(N,1);
         Q(NF+1,1)=0.15*Q(NF+2,1);
         VS(NF,NF)=VBOIL(Q(NF,1),TC(NF));
         VS(N,N)=VBOIL(Q(N,1),TC(N));
         VS(NF+2,NF+2)=(1.16-XQN)*VS(N,N);
         TK1=SLTC+273.2;  TK2=TC(N)+273.2;
         Q(1,1)=-LP(N,N)*(279.097*(TK1-TK2)-1.10491*(TK1**2-TK2**2)
                  +1.57081E-3*(TK1**3-TK2**3));
         TD = (TC(NA)-TC(1))/(NA-1);
         PD = (P(NA)-P(1))/(NA-1);
         DO I = 2 TO NA-1;
                 TC(I) = TC(1) + (I-1)*TD;
                  P(I) = P(1)+(I-1)*PD;  END;
         PD = (P(NF)-P(NA+1))/(NF-NA-1);
         DO I = NA+2 TO NF-1;
         P(I) = P(NA+1)+(I-NA-1)*PD;  END;
         TD = (TC(NF)-TC(NA+2))/(NF-NA-2);
         DO I = NA+2 TO NF-1;
```

```
                TC(I)=TC(NA+2)+(I-NA-2)*TD;   END;
        PD =  (P(N)-P(NF+1))/(N-NF-1);
        DO I = NF+2 TO N-1;
        P(I)  = P(NF+1)+(I-NF-1)*PD;  END;
        TD =  (TC(N)-TC(NF+2))/(N-NF-2);
        DO I = NF+2 TO N;
                TC(I)  = TC(NF+2)+ (I-NF-2)*TD;  END;
        DO I = 1 TO N-1; A(I,I)  = -1; B(I,I)=-1;
                A(I,I+1)  = 1; B(I+1,I)  = 1; END;
        A(N,N),B(N,N)  = -1;
        A(NF,NF+1),A(NA+1,NA+1),A(NA,NA+1),A(NF+1,NF+1)  = 0;
        B(NF+1,NF)  = 0;  B(NF+NSR+2,NF)  = 1;
        VS(NA,NA)=100.0-VP(NA+1,NA+1);
        LS(1,1)=-LP(N,N)+VS(NA,NA)-VS(1,1);
        DO I = 2 TO NA; LS(I,I)  = LS(1,1);
                VS(I,I)  = VS(NA,NA);  END;
        LS(NA+1,NA+1)  = VS(NF,NF)  + VP(NA+1,NA+1);
        DO I = NA+2 TO NA+NFR+2; LS(I,I)  = LS(NA+1,NA+1);
                VS(I,I)  = VS(NF,NF);  END;
        DO I = NA+NFR+2 TO NF;
                LS(I,I)  = LS(NA,NA)+LS(NA+1,NA+1);
                VS(I,I)  = VS(NF,NF);  END;
        LS(NF,NF)  = LS(NF,NF)-VS(NF,NF);  LP(NF,NF)  = 0;
        VP(NA,NA+1)=-VP(NA+1,NA+1);  LP(1,N)=-LP(NTN);
        LS(NF+1,NF+1)  = VS(NF+2,NF+2)  + VP(NF+1,NF+1);
        LS(NF+2,NF+2)=VS(N,N)+VP(NF+1,NF+1);
        DO I = NF+3 TO N-1;
                VS(I,I)=VS(N,N);  END;
        IF NSR>=2 THEN DO;
           DO I = NF+3 TO NF+NSR+1;
                  LS(I,I)=LS(NF+2,NF+2);  END;  END;
        DO I = NF+NSR+2 TO N;
                LS(I,I)=LS(NF,NF)+LS(NF+2,NF+2);  END;
        LS(N,N)=LS(N-1,N-1)-VS(N,N)+LP(N,N);
        VS(NA+1,NA+1)=0; /*FOR FIXED COND FLOW*/
        VS(NF+1,NF+1)=0; /*FOR FIXED COND FLOW*/
        B(NA+NFR+2,NA)  = 1; /*FOR BYPASSING CONDENSER*/
        B(NA+1,NA)  = 0; /*FOR BYPASSING CONDENSER*/
VBOIL:  PROC(Q,T);  DCL H,V;  DCL(Q,T,TK) BINARY FLOAT;  TK=T+40;
        H=137.8-0.729036*TK-1.42469E-3*TK*TK;
        H=24.45*H;  V=Q/H;  RETURN(V);  END VBOIL;
        END SETUP;
        DCL LOGIC(4);
        NH=1;
        GET LIST(N,M);
        NDIM=2*N;
        DCL (LOGP,LOGP1,K1) FLOAT;
        DCL (P1S,FGAST,SLTC) BINARY FLOAT;
        DCL TC(N) BINARY FLOAT CTL;
        ALLOCATE TC;
        DCL(YPFED(M),YRECY(M),DPHI(M),DDHBAR(M)) BINARY FLOAT CTL;
        ALLOCATE YPFED,YRECY,DPHI,DDHBAR;
        DCL (HE(M,M),HE1(M,M),HE2(M,M)) FLOAT CTL;
        ALLOCATE HE,HE1,HE2;
        DCL (PSI(NDIM,NDIM),TSDNN(N,N)) BINARY FLOAT(53) CTL;
        ALLOCATE TSDNN;
        DCL TNDIM(NDIM,1) BINARY FLOAT(53) CTL;
        ALLOCATE PSI;
```

```
ALLOCATE TNDIM;
DCL (PHIG(N,M),GAM(N,M),DELHBAR(N,M)) BINARY FLOAT CTL;
ALLOCATE PHIG,GAM,DELHBAR;
DCL (GSO(M),VOBAR(M),BETABAR(M),YFEED(M)) BINARY FLOAT CTL;
ALLOCATE GSO,YFEED;
DCL (L(N,N),V(N,N),Q(N,1),QF(N,1),ET(N,N),EV(N,N),JT(N,N),
JV(N,N),GT(N,N),GV(N,N),HT(N,N),HV(N,N),F(N,M),XX(N,M),YY(N,M),
ALPHA(M),BETA(M),GAMMA(M),
K(N,N),HS(N,N),GS(N,N),DK(N,N),DH(N,N),
DG(N,N),FF(N,1),X(N,1),Y(N,1),MO(N,N),XT(N,N),
SUMM(N,1),B(N,N),A(N,N),R(N,N),M1(N,N),M2(N,N),M3(N,N),
M4(N,N),XV(N,N),YV(N,N),
H(N,1),G(N,1),
DM(N,1),DE(N,1),UN(N,1),UM(M,1),W1(N,N),
W2(N,N),C(NDIM,1),
DEL(NDIM,1),VS(N,N),VP(N,N),LS(N,N),LP(N,N),ITZ(N,N),
ITB(N,N),NORM(N,N),IPSI(NDIM,NDIM),SUMX(N,1),Z(N,N),YT(N,N),
SUMY(N,1),TSNN(N,N),TSNM(N,M),TSN1(N,1),TSM(M),
P(N)) BINARY FLOAT CTL;
ALLOCATE   L,V,Q,QF,ET,EV,JT,JV,GT,GV,HT,HV,F,XX,YY,
K,HS,GS,DK,DH,DG,FF,
X,Y,MO,XT,SUMM,B,A,R,M1,M2,M3,M4,XV,YV,
H,G,DM,DE,UN,UM,W1,W2,
C,DEL,VS,VP,LS,LP,ITZ,ITB,TSNN,TSNM,TSN1,TSM,
NORM,IPSI,SUMX,Z,YT,SUMY,P;
DCL ADE,ADM,PDF,CKRDF,FKRR,FLGR,FKRB,RKRR,RLGR,
    CTLRKR,FKRV,CTLRLG,CFKR,CFKRLG;
DCL(D,DD,DDD,CON) BINARY FLOAT(53);
CON=0.0;
ALLOCATE ALPHA INIT(6.637,4.968,7.011,6.962,6.958,4.968),
         BETA INIT(1.395E-3,0.0,-8.1E-4,-9.5E-5,-3.5E-5,0.0),
         GAMMA INIT(2.045E-5,0.0,2.8E-6,3.5E-7,1.5E-7,0.0),
         VOBAR INIT(39.48,56.64,52.34,48.59,50.03,54.46),
         BETABAR INIT(0.00468,0.00547,-7.8E-5,-0.0181,
           0.00137,0.00894);
CALL SETUP(N,M); /*TO INITIALIZE THE MATRICES*/
LL=1;  CALL HENRYS(LL,M,NH,TK,HE);
LL=1; CALL RKWONG(LL,M,Y(1,*),P(1),TK,PHIG(1,*),DELHBAR(1,*),
DELH);
CALL ACTCO(LL,M,TK,XX(1,*),GAM(1,*));
PUT SKIP(2) EDIT('COMP','VOBAR','BETABAR','ALPHA','BETA',
'GAMMA') (X(10),A,X(4),A,X(3),A,X(3),A,X(5),A,X(8),A);
PUT SKIP(2) EDIT(((J,VOBAR(J),BETABAR(J),ALPHA(J),BETA(J),
GAMMA(J)) DO J = 1 TO M))
(SKIP,F(11),F(11,2),E(11,3),F(9,3),(2)E(12,3));
PUT SKIP(2) EDIT('FEED','COMP','RATE')
    (X(5),A,SKIP,X(5),A,X(4),A);
PUT SKIP EDIT((I,F(NA,I) DO I = 1 TO M))
    (SKIP,X(7),F(2),F(10,6));
TK=FGAST+273.15;
CALL RKWONG(3,M,YPFED,P(NA),TK,DPHI,DDHBAR,DELH);
CALL FINDH(FGAST,YPFED,100,DDHBAR,DELTAH);
QF(NA,1)=DELTAH;
CALL RKWONG(3,M,YRECY,P(NA),TK,DPHI,DDHBAR,DELH);
CALL FINDH(FGAST,YRECY,-VP(NA+1,NA+1),DDHBAR,DELTAH);
QF(NA,1)=QF(NA,1)+DELTAH;
TK=TC(NA+1)+273.15;
CALL RKWONG(3,M,YRECY,P(NA+1),TK,DPHI,DDHBAR,DELH);
```

```
        CALL FINDH(TC(NA+1),YRECY,-VP(NA+1,NA+1),DDHBAR,DELTAH);
        QF(NA,1)=QF(NA,1)-DELTAH;
START:  CALL MMGG(A,VS,N,N,N,TSNN)    ;
        V = TSNN+ VP  ;
        CALL MMGG(B,LS,N,N,N,TSNN)    ;
        L = TSNN + LP  ;
        MM=1;
        KOUNT=0;  CALL CHART;
CHART:  PROCEDURE;  PUT PAGE;
        PUT SKIP(5) EDIT('N','F','Q','QF') (X(5),A,(M)X(6),A,(M)X(6),
        X(4),A,X(10),A);
        PUT SKIP(2);
        DO I = 1 TO N;
        PUT SKIP EDIT(I,(F(I,J) DO J = 1 TO M),Q(I,1),QF(I,1))
             (X(4),F(2),(M)F(12,2),F(12,0),F(12,0));
             END;  END CHART;
        GO TO NOPRT;
        PUT PAGE;
        PUT SKIP(5)  EDIT('L')(X(34),A)  ;
        PUT SKIP(2);
        DO I = 1 TO N ;
        PUT SKIP(2)  EDIT((L(I,J)  DO J = 1 TO N ))  (F(10,2))  ;
        END ;
        PUT SKIP(5)  EDIT('V')(X(34),A)  ;
        PUT SKIP(2)  ;
        DO I = 1 TO N ;
        PUT SKIP(2)  EDIT((V(I,J)  DO J = 1 TO N))  (F(10,2))  ;
        END ;
         PUT PAGE ;
        PUT SKIP(5)  EDIT('LS')(X(34),A) ;
        PUT SKIP(2)  ;
        DO I = 1 TO N;
        PUT SKIP(2)  EDIT((LS(I,J)  DO J = 1 TO N))  (F(10,2))  ;
        END ;
        PUT SKIP(5)  EDIT('LP')(X(34),A)  ;
        PUT SKIP(2)  ;
        DO I = 1 TO N;
        PUT SKIP(2)  EDIT((LP(I,J)  DO J = 1 TO N))(F(10,2));
        END ;
        PUT PAGE ;
        PUT SKIP(5)  EDIT('A')(X(34),A);
        PUT SKIP(2);
        DO I = 1 TO N;
        PUT SKIP(2)  EDIT((A(I,J)  DO J= 1 TO N))(F(8,2));
        END ;
        PUT SKIP(5)  EDIT('B')(X(34),A)  ;
        PUT SKIP(2)  ;
        DO I = 1 TO N;
        PUT SKIP(2)  EDIT((B(I,J)  DO J = 1 TO N)) (F(8,2));
        END ;
        PUT PAGE;
        PUT SKIP(5)  EDIT('VS')(X(34),A)  ;
        PUT SKIP(2)  ;
        DO I = 1 TO N;
        PUT SKIP(2)  EDIT((VS(I,J)  DO J= 1 TO N))(F(10,2))  ;
        END ;
        PUT SKIP(5)  EDIT('VP')(X(34),A)  ;
        PUT SKIP(2)  ;
```

```
        DO I = 1 TO N;
        PUT SKIP(2) EDIT((VP(I,J) DO J = 1 TO N))(F(10,2)) ;
        END;
NOPRT:  PUT PAGE;
START1: ET,EV,JT,JV,GT,GV,HT,HV,R=0;
        KOUNT=KOUNT+1;
        IF KOUNT = 1 THEN DO ;
        TO = 233.15  ;
        LL = 2 ; CALL HENRYS(LL,M,NH,TO-.5,HE1) ;
        LL = 2 ; CALL HENRYS(LL,M,NH,TO+.5,HE2) ;
        DO I = 2 TO M ;
        GSO(I) = -(LOG(HE2(I,1))-LOG(HE1(I,1)))/(1/(TO+.5)-1/(TO-.5))
        *1.98726  ;
        END  ;
        TSM = 0 ; TSM(1) = 1 ; P1S= VPFUNCT(TO) ;
        LL = 2 ;
        CALL RKWONG(LL,M,TSM,P1S,TO,PHIG(1,*),DELHBAR(1,*),DELH);
        LL  = 3 ;
        CALL RKWONG(LL,M,TSM,P1S,TO,PHIG(1,*),DELHBAR(1,*),DELH);
        GSO(1) = DELH + 3369.2  ; /* HEAT OF VAP FROM PERRY*/
        PUT PAGE EDIT('IDEAL GAS ENTHALPIES AT -40 C') (X(5),A);
        PUT SKIP(2) EDIT('COMPONENT NO.','ENTHALPY, CAL/GMOL',
            (I,GSO(I) DO I = 1 TO M))
            (X(10),A,X(5),A,(M)(SKIP,X(15),F(2),X(12),F(12,2)));
        END  ;
        FF=0;
        DEL=0;
        PSI=0;
        K,HS,GS,DK,DH,DG=0;
        UM=1;
        UN=1;
        J=0;
        NORM=0;
        SUMY = 0    ;
        SUMX=0;
        H,G=0;
         IF KOUNT = 1 THEN DO;
        PHIG = 1; DELHBAR = 0; GAM = 1.0 ; END  ;
        ELSE DO ; LL= 2  ;
        CALL RKWONG(LL,M,YY(1,*),P(1),TK,PHIG(1,*),DELHBAR(1,*),DELH) ;
        DO I =  1 TO N ;
        TK = TC(I)  + 273.15    ;
        LL= 3;
        CALL RKWONG(LL,M,YY(I,*),P(I),TK,PHIG(I,*),DELHBAR(I,*),DELH) ;
        LL=2;   CALL ACTCO(LL,M,TK,XX(I,*),GAM(I,*));
        END ; END ;
START2: J=J+1;
        M0,M1,M2,M3,M4=0;
        DO I=1 TO N;
        FF(I,1)=F(I,J);
        TC1   =TC(I)+1.0;
        TK    =TC(I)+273.15;
        RT = 82.0574*TK    ;
        TK1   =TC(I)+274.15;
        TK2   =TC(I)+275.15;
        IF J=1 THEN DO;
        P1S = VPFUNCT(TK)   ;
        VBAR = EXP((TK-TO)*BETABAR(1))*VOBAR(1) ;
```

```
 TSM = 0 ; TSM(1) = 1 ; LL = 2 ;
CALL RKWONG(LL,M,TSM,P1S,TK,TSM,TSNM(I,*),DELH)    ;
LL  = 3 ;
CALL RKWONG(LL,M,TSM,P1S,TK,TSM,TSNM(I,*),DELH)    ;
FS = TSM(1)*P1S;
K(I,I) = GAM(I,1)*FS*EXP(VBAR*(P(I)-P1S)/RT)/(P(I)*PHIG(I,1));
DK(I,I) = DVPFUNCT(TK)/P(I) ;
END;
ELSE DO;
LL=2; CALL HENRYS(LL,M,NH,TK    ,HE);
LL=2; CALL HENRYS(LL,M,NH,TK1    ,HE1);
LL=2; CALL HENRYS(LL,M,NH,TK2    ,HE2);
K(I,I)=HE(J,1)/P(I) ;       /* FOR DERIVATIVE USE ONLY */
K1      =HE1(J,1)/P(I) ;
DK(I,I)=K1      -K(I,I);
VBAR = VOBAR(J)*EXP((TK-T0)*BETABAR(J))     ;
K(I,I) = K(I,I)*GAM(I,J)*EXP(VBAR    *P(I)/RT)/(      PHIG(I,J));
END;
GS(I,I) = ALPHA(J)*(TK-T0) + BETA(J)*(TK**2-T0**2)/2 +
GAMMA(J)*(TK**3-T0**3)/3 + GSO(J) -DELHBAR(I,J)   ;
DG(I,I)=ALPHA(J)+BETA(J)*TK+GAMMA(J)*TK*TK;
IF J=1 THEN DO;
HS(I,I) = 2.790974416E+02*(TK-T0)-2.209824996*(TK**2-T0**2)/2
+4.712433563E-03*(TK**3-T0**3)/3;
DH(I,I) = 2.790974416E+02-2.209824996*TK+4.712433563E-03*TK*TK;
END;
ELSE DO;
HS(I,I)=(LOG(HE1(J,1))-LOG(HE(J,1)))/(1/TK1    -1/TK    )*1.9872
+GS(I,I)+DELHBAR(I,J);
HS1     =(LOG(HE2(J,1))-LOG(HE1(J,1)))/(1/TK2    -1/TK1    )*
1.9872+GS(I,I)+DELHBAR(I,J);
DH(I,I)=HS1     -HS(I,I);
END;
END;
CALL MMGG(V,K,N,N,N,TSNN)   ;
Z = L + TSNN ;
TSDNN = Z ;
CALL MINV#(TSDNN,N,D,CON) ;
ITZ = -TSDNN ;
CALL MMGG(ITZ,FF,N,N,MM,X);
CALL MMGG(K,X,N,N,MM,Y);
DO I=1 TO N;
M0(I,I)=DK(I,I)*X(I,1);
END;
CALL MMGG(ITZ,V,N,N,N,TSNN ) ;
CALL MMGG(TSNN,M0,N,N,N,XT) ;
CALL MMGG(K,XT,N,N,N,TSNN) ;
YT = M0+TSNN ;
TSDNN = B ;
CALL MINV#(TSDNN,N,DD,CON) ;
ITB = TSDNN  ;
CALL MMGG(ITB,A,N,N,N,R);
DO I=1 TO N;
DO II=1 TO N;
M1(I,II)=R(I,II)*X(I,1);
M2(I,II)=A(I,II)*Y(II,1);
END;
M3(I,I)=DH(I,I)*X(I,1);
```

```
M4(I,I)=DG(I,I)*Y(I,1);
END;
CALL MMGG(B,M1,N,N,N,TSNN)   ;
TSNN = TSNN - M2  ;
ITZ = -ITZ  ;
CALL MMGG(ITZ,TSNN ,N,N,N,XV);
CALL MMGG(K,XV,N,N,N,YV  ) ;
ET=ET+XT-YT;
EV=EV+XV-YV;
CALL MMGG(HS,XT,N,N,N,TSNN) ;
HT=HT+TSNN+M3;
CALL MMGG(GS,YT,N,N,N,TSNN) ;
GT=GT+TSNN+M4;
CALL MMGG(HS,XV,N,N,N,TSNN) ;
HV=HV+TSNN ;
CALL MMGG(GS,YV,N,N,N,TSNN) ;
GV=GV+TSNN ;
DO I=1 TO N;
XX(I,J)=X(I,1);
YY(I,J)=Y(I,1);
END;
SUMX=SUMX+X;
SUMY = SUMY +Y     ;
CALL MMGG(HS,X,N,N,MM,TSN1 ) ;
H = H + TSN1 ;
CALL MMGG(GS,Y,N,N,MM,TSN1 ) ;
G = G + TSN1 ;
IF J<M THEN GO TO START2;
DO I=1 TO N;
NORM(I,J)=1/SUMX(I,1);
END;
CALL MMGG(ET,NORM,N,N,N,TSNN ) ;
ET = TSNN ;
TSNM = XX-YY;
CALL MMGG(TSNM ,UM,N,M,MM,DM);
CALL MMGG(L,H,N,N,MM,TSN1) ;
DE = TSN1+Q+QF ;
CALL MMGG(V,G,N,N,MM,TSN1) ;
DE = DE + TSN1 ;
ADM =SUM(DM**2);
ADE =SUM(DE**2);
ADM=SQRT(ADM );
ADE=SQRT(ADE );
CALL MMGG(L,HT,N,N,N,TSNN) ;
JT = TSNN ;
CALL MMGG(V,GT,N,N,N,TSNN) ;
JT = TSNN + JT ;
DO I=1 TO N; DO II=1 TO N;
W1(I,II)=R(I,II)*H(I,1);
W2(I,II)=A(I,II)*G(I,1);
END; END;
CALL MMGG(B,W1,N,N,N,TSNN) ;
JV = -TSNN ;
CALL MMGG(L,HV,N,N,N,TSNN) ;
JV = JV + TSNN ;
CALL MMGG(V,GV,N,N,N,TSNN) ;
JV = JV + TSNN + W2  ;
DO I=1 TO N; DO II=1 TO N;
```

```
          PSI(I,II)=ET(I,II);
          PSI(I,II+N)=EV(I,II);
          PSI(I+N,II)=JT(I,II);
          PSI(I+N,II+N)=JV(I,II);
          END;
          PSI(I,N+NA+1) = 0 ; /* FOR VARIABLE Q(NA+1) */
          PSI(I,N+NF+1) = 0   ;     /* FOR VARIABLE Q(NF+1)  */
          PSI(I+N,N+NA+1) = 0 ; /* FOR VARIABLE Q(NA+1)*/
          PSI(I+N,N+NF+1) = 0   ;     /* FOR VARIABLE Q(NF+1)  */
          DEL(I,1)=DM(I,1);
          DEL(I+N,1)=DE(I,1);
          END;
          PSI(N+NA+1,N+NA+1) = 1;/*FOR VARIABLE Q(NA+1)*/
          PSI(N+NF+1,N+NF+1) = 1   ; /* FOR VARIABLE Q(NF+1)    */
          CALL MINV#(PSI,NDIM,DDD,CON);
          IPSI=PSI; DEL = -DEL ;
          CALL MMGG(IPSI,DEL,NDIM,NDIM,MM,C)   ;
          DO I=1 TO N;
          TC(I) = TC(I) + C(I,1)     ;
          END;
          DO I = 1 TO NA, NA+2 TO NF, NF+2 TO N ; /* FOR VARIABLE COND*/
          VS(I,I)=VS(I,I)+C(I+N,1)    ;
          END;
          Q(NA+1,1)=Q(NA+1,1) + C(N+NA+1,1); /* FOR VARIABLE Q'S  */
          Q(NF+1,1) = Q(NF+1,1) +C(N+NF+1,1)     ;  /* FOR VARIABLE Q  */
          CALL MMGG(A,VS,N,N,N,TSNN)  ;
          V = TSNN + VP  ;
          CALL MMGG(F,UM,N,M,MM,SUMM)  ;
          CALL MMGG(V,UN,N,N,MM,TSN1) ;
          SUMM = SUMM + TSN1  ;
          CALL MMGG(LP,UN,N,N,MM,TSN1) ;
          SUMM = SUMM + TSN1 ; SUMM = -SUMM ;
          CALL MMGG(ITB,SUMM,N,N,MM,TSN1 ) ;
          DO I=1 TO N;
          LS(I,I)=TSN1(I,1)   ;
          END;
          CALL MMGG(B,LS,N,N,N,TSNN) ;
          L = TSNN + LP ;
          PUT PAGE DATA(ADE,ADM);
          PUT SKIP EDIT('ITERATION NO. =',KOUNT) (X(5),A,F(4)) ;
          PUT SKIP(2) EDIT('STAGE','T,C','L','V','H LIQ','H VAPOR',
          'P,ATM')
          (X(3),A,X(2),A,X(8),A,X(9),A,X(6),A,X(3),A,X(3),A);
          PUT SKIP;
          DO I=1 TO N;
          PUT SKIP EDIT(I,TC(I),-L(I,I),-V(I,I),H(I,1),G(I,1),P(I))
          (X(5),F(2),F(8,2),F(10,2),F(10,2),F(8,2),F(9,2),F(9,2)); END;
          IF ADM<1E-4 THEN IF ADE<1E+3 THEN GO TO PRINT;
          ELSE; ELSE;
          CALL MMGG(NORM,XX,N,N,M,TSNM) ;
          XX = TSNM ;
          DO I = 1 TO N ; NORM(I,I)=1/SUMY(I,1);END ;
          CALL MMGG(NORM,YY,N,N,M,TSNM);
          YY = TSNM  ;
          GO TO START1;
PRINT:    PUT SKIP(2) EDIT('CONVERGED')(X(10),A);
          SLGF,SLGA,SLGR,SLGP,SLGL = 0.0;
          DO J = 2 TO M; SLGF = SLGF +F(NA,J);
```

```
            SLGR = SLGR-V(NA+1,NA+1)*YY(NA+1,J);
            SLGA = SLGA-V(1,1)*YY(1,J);
            SLGL = SLGL-(L(N,N)+LP(1,N))*XX(N,J);
            SLGP = SLGP-V(NF+1,NF+1)*YY(NF+1,J);   END;
         FEED = 0.0; DO J = 1 TO M; FEED = FEED + F(NA,J); END;
         FKRR = (-V(NF+1,NF+1)*YY(NF+1,M))/F(NA,M);
         PDF = 1.0/(1.0-FKRR);   FLGR = SLGP/SLGF;
         FKRB = (-(L(N,N)+LP(1,N))*XX(N,M))/F(NA,M);
         RKRR = (-V(NA+1,NA+1)*YY(NA+1,M))/F(NA,M);
         FKRV = (-V(1,1)*YY(1,M))/F(NA,M);
         CFKR = (YY(NF+1,M)/F(NA,M))*FEED;
         CFKRLG = ((PDF-1.0)/PDF)*(SLGF/SLGP);
         CKRDF = (1.0+RKRR)/FKRV;   RLGR = SLGR/SLGF;
         CTLRKR = FKRR/(1.0-FKRR);   CTLRLG = SLGP/(1.0-SLGP);
FRMT1:   FORMAT(X(10),A(30),E(16,6));
FRMT2:   FORMAT(X(10),A(30),F(16,2));
FRMT3:   FORMAT(X(10),A(30),F(16,6));
         CALL CHART;
         PUT PAGE EDIT('LIQUID STREAMS') (X(50),A);  PUT SKIP(2);
         DO I = 1 TO N;
         PUT SKIP(2) EDIT((L(I,J) DO J = 1 TO N)) (F(10,2));  END;
         PUT PAGE EDIT('VAPOR STREAMS') (X(51),A);  PUT SKIP(2);
         DO I = 1 TO N;
         PUT SKIP(2) EDIT((V(I,J) DO J = 1 TO N)) (F(10,2));  END;
         PUT PAGE EDIT('STAGE','T,C','L','V','H LIQ','H VAPOR','P,ATM')
              (X(3),A,X(2),A,X(8),A,X(9),A,X(6),A,X(3),A,X(3),A);
         PUT SKIP;
         DO I=1 TO N;  IF I=NA+1|I=NF+1 THEN PUT SKIP;
         PUT SKIP EDIT(I,TC(I),-L(I,I),-V(I,I),H(I,1),G(I,1),P(I))
         (X(5),F(2),F(8,2),F(10,2),F(10,2),F(8,2),F(9,2),F(9,2));  END;
         PUT PAGE EDIT('COMPONENT') (X(5),(M/2)X(10),A);
         PUT SKIP EDIT('STAGE',(I DO I = 1 TO M),'SUM') (X( 5),A,X(5),
         (M) (F(2),X(9)),A);
         DO I=1 TO N;  IF I=NA+1|I=NF+1 THEN PUT SKIP;
         PUT SKIP EDIT(I,' Y=',(YY(I,J) DO J = 1 TO M),SUMY(I,1))
         (X(5),F(2),A,(M)E(11,4),F(8,5));  END;
         PUT PAGE EDIT('COMPONENT') (X(5),(M/2)X(10),A);
         PUT SKIP EDIT('STAGE',(I DO I = 1 TO M),'SUM')
              (X(5),A,X(5),(M)(F(2),X(9)),A);  DO I = 1 TO N;
         IF I=NA+1|I=NF+1 THEN PUT SKIP;
         PUT SKIP EDIT(I,' X=',(XX(I,J) DO J=1 TO M),SUMX(I,1))
              (X(5),F(2),A,(M)E(11,4),F(8,5));  END;
         PUT PAGE EDIT('COMPONENT') (X(5),(M/2)X(10),A);
         PUT SKIP EDIT('STAGE',(I DO I = 1 TO M))
              (X(5),A,X(5),(M)(F(2),X(9)));  DO I = 1 TO N;
         IF I=NA+1|I=NF+1 THEN PUT SKIP;
         PUT SKIP EDIT(I,' K=',(YY(I,J)/XX(I,J) DO J = 1 TO M))
              (X(5),F(2),A,(M)E(11,4));  END;
         PUT PAGE EDIT('STAGES')(X(4),A);
         DO I = 1 TO NA-1,NA+1 TO NF-1,NF+1 TO N-1;
         IF I=NA+1|I=NF+1 THEN PUT SKIP;
         PUT SKIP EDIT(I,'-',I+1,'L/V=',L(I,I)/V(I+1,I+1))
              (X(4),F(2),A,F(2),X(1),A,E(11,4));  END;
         PUT PAGE EDIT('COMPONENT') (X(5),(M/2)X(10),A);
         PUT SKIP EDIT('STAGES',(I DO I = 1 TO M))
              (X(4),A,X(5),(M)(F(2),X(9)));
         DO I = 1 TO NA-1,NA+1 TO NF-1,NF+1 TO N-1;
         IF I=NA+1|I=NF+1 THEN PUT SKIP;
```

```
          PUT SKIP EDIT(I,'-',I+1,'AVK=',((YY(I,J)+YY(I+1,J))/
              (XX(I,J)+XX(I+1,J)) DO J = 1 TO M))
              (X(4),F(2),A,F(2),X(1),A,(M)E(11,4));  END;
          PUT PAGE EDIT('COMPONENT') (X(5),(M/2)X(10),A);
          PUT SKIP EDIT('STAGES',(I DO I = 1 TO M))
              (X(4),A,X(5),(M)(F(2),X(9)));
          DO I = 1 TO NA-1,NA+1 TO NF-1,NF+1 TO N-1;
          IF I=NA+1|I=NF+1 THEN PUT SKIP;
          PUT SKIP EDIT(I,'-',I+1,'ABF=',((L(I,I)/V(I+1,I+1))/
              ((YY(I,J)+YY(I+1,J))/(XX(I,J)+XX(I+1,J)))
              DO J = 1 TO M))(X(4),F(2),A,F(2),X(1),A,(M)E(11,4));  END;
          PUT PAGE EDIT('ENERGY BALANCE ERROR',ADE)(R(FRMT1));
          PUT SKIP(2) EDIT('MATERIAL BALANCE ERROR',ADM)(R(FRMT1));
          PUT SKIP(2) EDIT('FEED STAGES',NA,NA+NFR+2,NF+NSR+2)
              (X(10),A(34),3 F(4));
          PUT SKIP(2) EDIT('FRC CONDENSER LOAD',Q(NA+1,1))(R(FRMT1));
          PUT SKIP(2) EDIT(' "  REBOILER HEAT',Q(NF,1))(R(FRMT1));
          PUT SKIP(2) EDIT('STR FINAL CONDENSER LOAD',Q(NF+1,1))
              (R(FRMT1));
          PUT SKIP(2) EDIT(' "  MAIN CONDENSER LOAD',Q(NF+2,1))
              (R(FRMT1));
          PUT SKIP(2) EDIT(' "  REBOILER HEAT',Q(N,1))(R(FRMT1));
          PUT SKIP(2) EDIT('SCRUB COOLING LOAD',Q(1,1))(R(FRMT1));
          PUT SKIP(2) EDIT('ABS FEED ENTHALPY',QF(NA,1))(R(FRMT1));
          PUT SKIP(2) EDIT('KRYPTON PROCESS DF',PDF)(R(FRMT2));
          PUT SKIP(2) EDIT('   "    COLUMN DF',CKRDF)(R(FRMT2));
          PUT SKIP(2) EDIT('   "    PROCESS CF',CFKR)(R(FRMT2));
          PUT SKIP(2) EDIT('   "    CF, REL/LG',CFKRLG)(R(FRMT2));
          PUT SKIP(2) EDIT('FRACTION KR RECOVERED',FKRR)(R(FRMT3));
          PUT SKIP(2) EDIT('   "     KR VENTED',FKRV)(R(FRMT3));
          PUT SKIP(2) EDIT('   "     KR BYPASSED',FKRB)(R(FRMT3));
          PUT SKIP(2) EDIT('   "     LIGHT GAS RECOVERED',FLGR)
              (R(FRMT3));
          PUT SKIP(2) EDIT('KRYPTON RECYCLE RATIO',RKRR)(R(FRMT3));
          PUT SKIP(2) EDIT('   "    CAP/LOSS RATIO',CTLRKR)(R(FRMT2));
          PUT SKIP(2) EDIT('LIGHT GAS RECYCLE RATIO',RLGR)(R(FRMT3));
          PUT SKIP(2) EDIT('   "      CAP/LOSS RATIO',CTLRLG)(R(FRMT3));
          PUT SKIP(2) EDIT('ABS FEED GAS TEMP, C',FGAST)(R(FRMT2));
          PUT SKIP(2) EDIT('ABS FEED LIQ TEMP, C',SLTC)(R(FRMT2));
          DO I = 6,3;
          YFEED(I)=(F(NA,I)+VP(NA,NA+1)*YY(NA+1,I))/(FEED+VP(NA,NA+1));
          END;   PUT FILE (PUNCH) SKIP EDIT
          (N,NA,NA+NFR+2,NF,NF+NSR+2,' 2',(YFEED(I) DO I = 6,3))
          (5 (F(2),X(6)),A(2),X(6),2 E(10,3));
          DO I = 1 TO N;  PUT FILE (PUNCH) SKIP EDIT
             (I,XX(I,6),YY(I,6))(X(5),F(2),2 E(15,4));  END;
          DO I =1 TO N;  PUT FILE (PUNCH) SKIP EDIT
             (I,XX(I,3),YY(I,3))(X(5),F(2),2 E(15,4));  END;
FINDH:    PROCEDURE(T,Y,V,DELHBAR,DELTAH);
          DCL H(6),I,T2;
          DCL (Y(*),V,T,DELHBAR(*)) BINARY FLOAT;
          T1=233.15;  T2=T+273.15;  DELTAH=0.0;
          H(1)=3513.05;  H(2)= 521.93;  H(3)=-773.62;
          H(4)=-194.51;  H(5)=-1199.32;  H(6)=-294.94;
          DO I=1 TO 6;
             DELTAH=DELTAH+V*Y(I)*(H(I)-DELHBAR(I)+
                   ALPHA(I)*(T2-T1)+BETA(I)*(T2*T2-T1*T1)/2.0+
                   GAMMA(I)*(T2**3-T1**3)/3.0);  END;  END FINDH;
```

```
VPFUNCT: PROC(T) ;
        /* CALCULATES VP OF CO2 FROM EQ OF MICHELS */
        DCL LOGVP FLOAT, T, VP  ;
        LOGVP = 24.61930 -  1353.202/T-8.142537*LOG10(T)+.006259156*T  ;
        VP = 10**LOGVP ;  RETURN(VP)  ;
        END VPFUNCT  ;
DVPFUNCT: PROC(T)    ;
        /* CALCULATES VP DERIVATIVE */
        DCL T, DPDT, LN10 FLOAT ;
        LN10 = LOG(10)      ;
        DPDT = VPFUNCT(T)   ;
        DPDT = DPDT*(LN10*1353.202/(T*T)-8.142537/T +LN10*.006259156) ;
        RETURN(DPDT) ;
        END DVPFUNCT  ;
ACTCO:  PROC(L,M,T,X,GAM)     ;
        DCL RT  ;
        DCL I,J,K ;
        DCL S(2) LABEL ;
        DCL A(M,M) CTL    ; DCL(GAM(*),X(*) )   BINARY FLOAT   ;
        GO TO S(L)   ;
S(1) :  ALLOCATE A ;    A = 0 ;
        A(3,3)=851.21;  A(4,4)=863.49;  A(5,5)=885.6;
        DO I = 2 TO M ;
        DO J = I+1 TO M ; A(I,J)  = (A(I,I) + A(J,J))/ 2 ;
        A(J,I) = A(I,J) ; END ;          END ;
        PUT SKIP(2) EDIT('MATRIX OF MARGULES CONSTANTS')
        (X(10),A)   ;
        PUT SKIP(1) EDIT((( A(I,J) DO I = 1 TO M) DO J = 1 TO M))
        (SKIP,X(8),(M)F(10,5)) ;
        RETURN  ;
S(2) :  RT = 1.98726*T  ;
        SUM1 = 0 ; SUM2 = 0 ;
        DO J = 2 TO M ;
        DO I = 2 TO M ;
        SUM1 = SUM1 + A(I,J)  * X(I)*X(J)  ; END ; END;
        DO K = 2 TO M ;
        SUM2 = 0 ; DO I = 2 TO M    ;
        SUM2 = SUM2 + X(I)  * A(I,K)  ; END ;
        GAM(K)  = EXP((SUM1-2*SUM2)/RT);
        END  ;
        GAM(1)  = EXP(SUM1/RT)   ;
        RETURN  ;
        END ACTCO  ;
/*GENERALIZED CORRELATION FOR HENRY'S CONSTANT, PRESTON & PRAUSNITZ*/
HENRYS: PROC(L,N,NH,T,H) ;
        DCL R,V,Q,K,X,Y,Z,U,TR,VR,T,I,J;
        DCL C(16) STATIC INITIAL(
        0.42457138, -0.97214424, -0.48309824, -0.12611504, 0.02042006,
        0.15446554, -0.06794337, 0.06654287,  0.30730388, 0.27319810,
        -0.32214081, -0.07442808,
        0.35402549, 0.05514906, 0.02432847, 0.86320302);
        DCL LOGQ FLOAT, LOGH FLOAT;
        DCL IDENT(6) CHAR(8) INIT('  CO2    ','  XE    ','  O2    ',
            '  CO    ','  N2    ','  KR    ');
        DCL (VS(N ), TC(N ), VC(N ), PC(N ),W(N )) CONTROLLED;
        DCL (TCI(N,N), VCI(N,N)) CONTROLLED;
        DCL H(*,*);
        DCL S(2) LABEL;
```

```
        R = 82.0574  ;
        GO TO S(L);
S(1):   ALLOCATE VS,TCI,VCI;
        ALLOCATE TC INIT(304.2,289.7,154.6,132.9,126.2,209.4),
           PC INIT(72.8,57.6,49.8,34.5,33.5,54.3),
           VC INIT(0.0,0.0,0.0,0.0,0.0,0.0),
           W  INIT(0.225,0.0,0.021,0.041,0.040,0.0);
        PUT SKIP EDIT('TC','PC','VC','W','COMP') (X(10),A,X(10),A,X(10),
        A,X(10),A,X(10),A);
        PUT SKIP;
        DO I=1 TO N ;
        IF VC(I)=0  THEN VC(I)=(R*TC(I)/PC(I))*(0.291-0.08*W(I));
        PUT SKIP EDIT(TC(I),PC(I),VC(I),W(I),IDENT(I)) (X(8),F(6,2),
            X(6),F(6,2),X(6),F(6,2),X(6),F(5,3),X(7),A(8));
        END;
        PUT SKIP(2);
        DO J=(NH+1) TO N;
        PUT SKIP EDIT('IDENT(',J,')',IDENT(J)) (X(8),A,F(2),A,A(8));
        END;
        PUT SKIP(2);
        PUT SKIP EDIT('MATRIX OF TCI') (X(15),A);
        PUT SKIP;
        TCI=0; VCI=0; H=0;
        TCI(2,1)=285.25;  TCI(3,1)=200.36;  TCI(4,1)=180.44;
        TCI(5,1)=197.66;  TCI(6,1)=237.59;  VCI(2,1)=95.58;
        VCI(3,1)=92.26;   VCI(4,1)=86.48;   VCI(5,1)=93.93;
        VCI(6,1)=93.83;
        DO I=1 TO NH;  DO J=(NH+1) TO N;
        IF VCI(J,I)=0  THEN VCI(J,I)=21.888+0.79827*VC(I);
        END; END;
        DO J=1 TO N;
        PUT SKIP EDIT((TCI(J,I) DO I=1 TO N))(X(10),(N)F(5,1));
        END;
        PUT SKIP(2);
        PUT SKIP EDIT('MATRIX OF VCI') (X(15),A);
        PUT SKIP;
        DO J=1 TO N;
        PUT SKIP EDIT((VCI(J,I) DO I=1 TO N))(X(10),(N)F(5,1));
        END;
        PUT PAGE;
        RETURN;
S(2):   DO I=1 TO NH; DO J=(NH+1) TO N;
        TR=T/TCI(J,I);
        CALL VOLSAT(T,TC(I));
        VR=VS(I)/VCI(J,I);
        C1T=C(1)+C(2)/TR+C(3)/(TR*TR)+C(4)/TR**3+C(5)/TR**5;
        C2T=C(6)+C(7)/TR;
        C3T=C(15)/TR;
        C4T=C(9)/TR**3+C(10)/TR**4+C(11)/TR**5;
        C5T=C(12)/TR**3+C(13)/TR**4+C(14)/TR**5;
        V=2*C1T/VR+3*C2T/(2*VR*VR)+4*C(8)/(3*VR**3)+6*C3T/(5*(VR**5));
        U=C4T+C5T/(VR*VR)-C5T/(2*C(16));
        X=EXP(-C(16)/(VR*VR))/(VR*VR);
        Y=C4T/(2*C(16))+C5T/(2*(C(16)**2));
        Z=EXP(-C(16)/(VR*VR))-1;
        LOGQ=V+U*X-Y*Z;
        Q=EXP(LOGQ);
        H(J,I)=Q*R*T/VS(I);
```

```
        END; END;
        RETURN;
/*CORRELATION FOR MOLAR LIQUID VOLUMES, LYCKMAN, ECKERT & PRAUSNITZ*/
VOLSAT: PROC(TT,TC);
        DCL TT,TC,TR1;
        DCL VR(3);
        DCL K;
        DCL A(3) STATIC INITIAL(0.11917,0.98465,-0.55314) ;
        DCL B(3) STATIC INITIAL(0.009513,-1.60378,-0.15793) ;
        DCL C(3) STATIC INITIAL(0.21091,1.82484,-1.01601)  ;
        DCL D(3) STATIC INITIAL(-0.06922,-0.61432,0.34095 ) ;
        DCL E(3) STATIC INITIAL(0.07480,-0.34546,0.46795 ) ;
        DCL F(3) STATIC INITIAL(-0.084476,0.087037,-0.239938) ;
        TR1=TT/TC;        DIF=1-TR1;
        DO K=1 TO 3;
        VR(K)=A(K)+B(K)*TR1+C(K)*TR1*TR1+D(K)*(TR1**3)+E(K)/TR1+F(K)*
          LOG(DIF);
        END;
        VS(I)=VC(I)*(VR(1)+VR(2)*W(I)+VR(3)*W(I)*W(I));
END VOLSAT;
END HENRYS;
RKWONG: PROC(L,M,Y,P,T,PHIG,DELHBAR,DELH);
        DCL P BINARY FLOAT   ;
        DCL RT  ;
        DCL EX,AM,BM      ;
        DCL R STATIC  ;
        DCL(Y(*), PHIG(*) ,VBAR(M),DELHBAR(*) ) BINARY FLOAT  ;
        DCL Z STATIC  ;
        DCL I, J ;
        DCL S(3) LABEL ;
        DCL (A(M,M),B(M), OMEGA(M) FLOAT,OMEGB(M) FLOAT,
        ZC(M,M), W(M), PC(M,M), VC(M,M), TC(M,M),K(M,M) FLOAT,
        IDENT(M) CHAR(8)) CTL;
        R = 0.0820574 ;        EX = 1/3  ;
        GO TO S(L) ;
S(1):   PUT SKIP(2) EDIT('INPUT PARAMETERS FOR R_K EQUATION',
        'OMEGA', 'OMEGB', 'W', 'TC,DEGK', 'PC,ATM', 'VC,LITER',
        'COMPONENT')
        (X(10), A, SKIP, X(15),A,X(5),A, X(8),A,X(4),A,X(5),
        A,X(2),A,X(1),A)  ;
        ALLOCATE A,B,ZC,PC,VC,TC,K;
        ALLOCATE OMEGA INIT(0.4470,0.4278,0.4278,0.4278,0.4278,0.4278),
          OMEGB INIT(0.0911,0.0867,0.0867,0.0867,0.0867,0.0867),
          W     INIT(0.225,0.0,0.021,0.041,0.040,0.0),
          IDENT INIT('  CO2  ','  XE  ','  O2  ','  CO  ',
                     '  N2  ','  KR  ');
        VC=0.0;  TC(1,1)=304.2;  PC(1,1)=72.8;  TC(2,2)=289.7;
        PC(2,2)=57.6;  TC(3,3)=154.6;  PC(3,3)=49.8;  TC(4,4)=132.9;
        PC(4,4)=34.5;  TC(5,5)=126.2;  PC(5,5)=33.5;
        TC(6,6)=209.4;  PC(6,6)=54.3;
        DO I = 1 TO M;
        PUT SKIP EDIT(OMEGA(I), OMEGB(I), W(I), TC(I,I),PC(I,I),
        VC(I,I),IDENT(I))
        (X(10),F(10,5),F(10,5),F(10,3),F(10,2),F(10,2),F(10,5),X(2),
        A(8));
        END ;
        K=0.0;
        PUT SKIP(2) EDIT('CORRECTIONS FOR GEOM MEAN,KIJ')(X(10),A) ;
```

```
        DO I = 1 TO M-1 ;
        PUT SKIP EDIT(I) (X(10),F(2)) ;
        DO J = I+1 TO M ;
        PUT EDIT(K(I,J)) (F(10,3));
        END ; END;
        DO I = 1 TO M ;
        DO J = I TO M ;
        ZC(I,J) = 0.291-0.08*(W(I) + W(J))/2 ;
        END ;
        IF VC(I,I) > 0 THEN ZC(I,I) = PC(I,I)*VC(I,I)/(R*TC(I,I)) ;
        ELSE ; VC(I,I) = ZC(I,I)*R*TC(I,I)/PC(I,I) ;
        B(I) = OMEGB(I)*R*TC(I,I)/PC(I,I) ;
        END ;
        DO I = 1 TO M ;
        DO J=I+1 TO M ;
        VC(I,J) =(((VC(I,I)**EX+ VC(J,J)**EX)/2))**3 ;
        TC(I,J) = SQRT(TC(I,I)*TC(J,J))*(1-K(I,J)) ;
        PC(I,J) = ZC(I,J)*R*TC(I,J)/VC(I,J) ;
        END ; END ;
        DO I = 1 TO M ; DO J = I TO M ;
        A(I,J) = (OMEGA(I) + OMEGA(J))*R*R*(TC(I,J)**2.5)/(2*PC(I,J));
        A(J,I) = A(I,J) ;
        END ; END;
        RETURN ;
S(2) : Z = 1.0 ;
        RETURN ;
S(3) :  AM = 0 ;
        RT = R*T ;
        BM = 0 ;
        DO I = 1 TO M ;
        BM = BM + Y(I)*B(I) ;
        DO J = 1 TO M ;
        AM = AM+Y(I)*Y(J)*A(I,J) ;
        END ; END ;
        CALL VOLG ;
        DELH = RT*(1.5*AM/(BM*RT*T**0.5)*LOG((VG+BM)/VG)+1-P*VG/RT) ;
        DELH = (1.98726/0.0820574)*DELH ;
        DO I = 1 TO M ;
        PHIG(I) = FUNCT(I) ;
        VBAR(I) = VFUNCT(I) ;
        DELHBAR(I) = HFUNCT(I) ;   END ;
FUNCT:  PROC(I) ;
        DCL X ;
        X = 0 ; DO J = 1 TO M ; X = X+Y(J)*A(J,I) ; END ;
        X = -2*X ;
        X = (X/(R*BM*T**1.5))*LOG((VG+BM)/VG) + LOG(VG/(VG-BM))
        + B(I)/(VG-BM) + AM*B(I)*(LOG((VG+BM)/VG)-BM/(VG+BM))/
        (R*T**1.5*BM*BM) - LOG(P*VG/(R*T)) ;
        X = EXP(X ) ;
        RETURN(X) ;
        END FUNCT ;
VFUNCT: PROC(I) ;
        DCL X,J,XX ;
        T12 = SQRT(T) ; RT = R*T ;
        X = 0 ;
        DO J = 1 TO M ; X = X + Y(J)*A(J,I) ; END ;
        X = -2*X/(T12*VG*(VG+BM)) ;
        X = X + (RT/(VG-BM))*(1+B(I)/(VG-BM)) ;
```

```
        X = X + AM*B(I)/(T12*VG*(VG+BM)**2) ;
        XX = RT/(VG-BM)**2   ;
        XX = XX-(AM/T12)*(2*VG+BM)/(VG*VG*(VG+BM)**2)   ;
        X = X/XX ;
        RETURN(X)   ;
        END VFUNCT ;
HFUNCT: PROC(I) ;
        DCL X,J     ;
        X = 0 ; DO J = 1 TO M ; X = X + Y(J)*A(J,I) ; END  ;
        X = (2*X-AM*B(I)/BM)/BM  ;
        X = X*LOG((VG+BM)/VG ) ;
        X = X + (AM/BM)*(VG*B(I)-VBAR(I)*BM)/(VG*(VG+BM)) ;
        X = 1.5*X/SQRT(T)    + RT -P*VBAR(I) ;
        X = X * 1.98726/R  ;
        RETURN(X)    ;
        END HFUNCT  ;
VOLG:   PROC; /* SOLVES RK USING CUBIC SOLUTION  */
        DCL A,B,C(4),D(4),PHI,RT,VGAS(3) ,A3,B3,L,M,N,MM,RR,K,T12      ;
        L = 2 ; M = 3 ; RT = R*T ; T12 = SQRT(T) ;  C(1) = 1 ;
        C(2) = -RT/P ;C(3)=-BM**2-RT*BM/P+AM/(T12*P) ; C(4)=-AM*BM/(P*
        T12) ; D(1) = 1 ; D(2) = 0 ; D(3) = (3*C(3)-C(2)**L)/3 ;
        D(4) = (27*C(4)-9*C(2)*C(3)+2*C(2)**M)/27      ;
        RR = (D(3)/3)**M + (D(4)/2)**L  ;  IF RR<0 THEN DO ; MM=1 ;
        PHI = ATAND(SQRT( -(D(3)**M/27 + D(4)**L/4) /(D(4)**L/4)))    ;
        DO K = 0 TO 2 ;
        VGAS(K+1) =2* SQRT(-D(3)/3)*COSD(PHI/3 + 120*K) ;
        VGAS(K+1) = VGAS(K+1) - C(2)/3  ;   END ;
        VG = MAX(VGAS(1),VGAS(2),VGAS(3) ) ;           END ;
        ELSE DO ; MM= 0 ; A3=(-D(4)/2+SQRT(RR)); B3=(-D(4)/2-SQRT(RR));
        A = ABS(A3)**(1/3) ;
        IF A3<0 THEN A = -A  ; B = ABS(B3)**(1/3) ; IF B3<0 THEN B=-B;
        VG = A + B -C(2)/3  ;     END  ;
        RETURN ; END     VOLG  ;
        END RKWONG      ;
MMGG:   PROCEDURE(A,B,K,L,M,C );
        /*  GENERAL MATRIX MULT OF C = A X B COPIED FROM PL1 SSR */
        DECLARE (A(*,*),B(*,*),C(*,*)) BINARY FLOAT      ;
        DECLARE S BINARY FLOAT(53)   ;
        DECLARE (K,L,M,I,J,N) BINARY FIXED  ;
        DECLARE ERROR EXTERNAL CHARACTER(1)  ;
        ERROR = 'D'   ;
        IF K > 0 THEN IF L > 0 THEN IF M > 0 THEN DO  ;
        I = 0  ;
NEXTI:  I = I + 1 ;
        J = 0 ;
NEXTJ:  J = J + 1 ;
        S = 0 ;
        DO N = 1 TO L ;  S = S+MULTIPLY(A(I,N),B(N,J),53);
        END  ;
        C(I,J) = S ;
        IF J < M THEN GO TO NEXTJ  ;
        ELSE IF I < K THEN GO TO NEXTI  ;
        ERROR = '0' ;
        END ;
      END      MMGG  ;
/*MINV# IS THE DOUBLE PRECISION MATRIX INVERSION SUBROUTINE FROM THE
  SCIENTIFIC SUBROUTINE PACKAGE.  THE CALCULATION OF THE DERTERMINANT
  HAS BEEN OMMITTED.*/
```

```
MINV#:  PROC(A,N,D,CON);
        DCL ERROR EXTERNAL CHARACTER(1);
        DCL (I,J,K,N,L(N),M(N)) FIXED BINARY;
        DCL (A(*,*),BIGA,HOLD,D,CON,S) BINARY FLOAT(53);
        ERROR='0';
        IF N<=0 THEN DO;
        ERROR='1';
        GO TO FIN;
        END;
        IF CON=0 THEN S=1.0E-15;
        ELSE S=CON;
        IF N=1 THEN DO;
        D=A(1,1);
        IF ABS(D)<=S THEN DO;
        ERROR='2';
        END;
        ELSE A(1,1)=1/D;
        GO TO FIN;
        END;
        D=1.0;
        DO K=1 TO N;
        L(K)=K;
        M(K)=K;
        BIGA=A(K,K);
        DO I=K TO N;
        DO J=K TO N;
        IF ABS(BIGA)<ABS(A(I,J)) THEN DO;
        BIGA=A(I,J);
        L(K)=I;
        M(K)=J;
        END;
        END;
        END;
        J=L(K);
        IF L(K)>K THEN DO;
        DO I=1 TO N;
        HOLD=-A(K,I);
        A(K,I)=A(J,I);
        A(J,I)=HOLD;
        END;
        END;
        I=M(K);
        IF M(K)>K THEN DO;
        DO J=1 TO N;
        HOLD=-A(J,K);
        A(J,K)=A(J,I);
        A(J,I)=HOLD;
        END;
        END;
        IF ABS(BIGA)<=S THEN DO;
        D=0.0;
        END;
        DO I=1 TO N;
        IF I ¬= K THEN A(I,K)=A(I,K)/(-A(K,K));
        END;
        DO I=1 TO N;
        IF I ¬= K THEN DO;
        DO J=1 TO N;
```

```
          IF J ¬= K THEN A(I,J)=A(I,K)*A(K,J)+A(I,J);
          END;
          END;
          END;
          DO J=1 TO N;
          IF J ¬= K THEN A(K,J)=A(K,J)/A(K,K);
          END;
 COMP:    /* CONTINUE */
          A(K,K)=1.0/A(K,K);
          END;
          K=N;
 LOOP:    K=K-1;
          IF K>0 THEN DO;
          I=L(K);
          IF I>K THEN DO;
          DO J=1 TO N;
          HOLD=A(J,K);
          A(J,K)=-A(J,I);
          A(J,I)=HOLD;
          END;
          END;
          J=M(K);
          IF J>K THEN DO;
          DO I=1 TO N;
          HOLD=A(K,I);
          A(K,I)=-A(J,I);
          A(J,I)=HOLD;
          END;
          END;
          GO TO LOOP;
          END;
 FIN:     RETURN;
 END MINV#;
 END STAGE;
/*
//GO.PUNCH DD SYSOUT=B,DCB=(RECFM=F,BLKSIZE=80)
//GO.PLIDUMP DD SYSOUT=A
//GO.SYSIN DD *
42 6
8 0 16 1
18.89 -27.78 -21 -22 -38 -24 -19 -24 -23 -22.78
22 22 20 20 18 18
1.0E-12 8.0E-2 1.0E-12 1.0E-12 1.0E-7
0.0 0.265 0.0 0.0 0.0
10300 16000 1.0
6.6 3.2 2.4 0.061 125
/*
//
```

APPENDIX B:

THE McCABE-THIELE PLOTTING PACKAGE

FOR THE KALC/ORGDP PROGRAM

```
//XXXXXX JOB (XXXXX),'BIN X XXXXX',CLASS=G,TYPRUN=HOLD
//*ROUTE PRINT LOCAL
//*CLASS CPU91=30S,R=270K,IO=10,SPECIAL=TAPE
// EXEC FORTHCLG,REGION.GO=256K
//FORT.SYSIN DD *
      DIMENSION X(100),Y(100),ISTG(100),YFEED(2)
      COMMON /XYT/ TX(100),TY(100),K,NAME(2),KOLM(2),JBNUM(2)
      DATA X/100*0.0/,Y/100*0.0/,ISTG/100*0/
      CALL QQQCPA(48,0.1)
      CALL QQQCPA(42,0.5)
      CALL QQQCPA(41,0)
      CALL JOBNUM(JBNUM)
      READ (5,100) N,NA,NFF,NF,NSF,NDS,YFEED(1),YFEED(2)
  100 FORMAT(6(I2,6X),2E10.3)
      DO 901 K=1,NDS
      READ (5,200,END=999) (ISTG(I),X(I),Y(I),I=1,N)
  200 FORMAT(5X,I2,2E15.4)
      DO 104 I=1,NA
      TX(I)=X(I)
  104 TY(I)=Y(I)
      CALL ABSORB(NA,X(N),YFEED(K))
      NT=NF-NA
      NTF=NFF-NA
      ZF=X(NA)
      DO 102 I=1,NT
      TX(I)=X(NA+I)
  102 TY(I)=Y(NA+I)
      CALL FRCSTR(NT,NTF,ZF,1)
      NT=N-NF
      NTF=NSF-NF
      ZF=X(NF)
      DO 103 I=1,NT
      TX(I)=X(NF+I)
  103 TY(I)=Y(NF+I)
      CALL FRCSTR(NT,NTF,ZF,2)
  901 CONTINUE
  999 CALL ADVANS
      STOP
      END
      SUBROUTINE FRCSTR(NT,NTF,ZF,KFS)
      COMMON /XYT/ TX(100),TY(100),K,NAME(2),KOLM(2),JBNUM(2)
      COMMON /BM3/ XM3,B3
      DIMENSION BUFFER(4000),TIX(100),TIY(100)
      TX(NT+1)=TX(NT)
      TY(NT+1)=TX(NT+1)
      TX(NT+2)=TY(1)
      TY(NT+2)=TX(NT+2)
      TX(NT+3)=TX(NT)/2.0
      TY(NT+3)=TX(NT+3)
      TX(NT+4)=TY(1)*2.0
      TY(NT+4)=TX(NT+4)
      TX(NT+5)=SQRT(TX(NTF-1)/TX(NTF))*TX(NTF)
      TY(NT+5)=SQRT(TY(NTF)/TY(NTF+1))*TY(NTF+1)
      TX(NT+6)=ZF
      TY(NT+6)=ZF
      NI=NT+6
      CALL SETPLT(TX,TY,NI,'LOG ','MECH',-1.,-1.,BUFFER,4000,'BARKER$')
      CALL CNVRT(TIX,TIY,TX,TY,NI)
```

```
      XM3=(TIY(NT+5)-TIY(NT+6))/(TIX(NT+5)-TIX(NT+6))
      B3=TIY(NT+6)-XM3*TIX(NT+6)
      CALL QQLINE(TX(NT+3),TY(NT+3),TX(NT+4),TY(NT+4),0)
      CALL QQLINE(TX(NT+2),TY(NT+2),TX(1),TY(1),0)
      NI=NT+1
      CALL LINPLT(TIX,TIY,NI,-1,0)
      NI=NT-1
      DO 100 I=1,NI
      CALL QQLINE(TX(I),TY(I),TX(I),TY(I+1),0)
  100 CALL QQLINE(TX(I),TY(I+1),TX(I+1),TY(I+1),0)
      CALL QQLINE(TX(NT+2),TY(NT+2),TX(1),TY(2),0)
      DO 200 I=1,NI
  200 CALL QQLINE(TX(I),TY(I+1),TX(I+1),TY(I+2),0)
      XMAXQ=AMAX1(TIX(NT+6),TIX(NT+5))
      XMINQ=AMIN1(TIX(NT+6),TIX(NT+5))
      CAIL PLTFNC(0.01,XMINQ,XMAXQ,2)
      CALL TITLE(0.4,'KALC STUDIES$','X$','Y$')
      CALL QQQSYM(8.5,10.1,0.25,KOLM(KFS),0.0,4)
      CALL QQQSYM(9.0,0.25,0.25,NAME(K),0.0,4)
      CALL QQQSYM(0.1,10.1,0.2,JBNUM,0.0,8)
      RETURN
      END
      SUBROUTINE ABSORB(NA,X0,YFEED)
      COMMON /XYT/ TX(100),TY(100),K,NAME(2),KOIM(2),JBNUM(2)
      DIMENSION BUFFER(4000),TIX(100),TIY(100)
      REAL MULT(2) /2.0,1.05/
      DO 100 I=1,NA
      TX(NA-I+2)=TX(NA-I+1)
  100 TY(NA-I+2)=TY(NA-I+1)
      IF (K .EQ. 1) TX(1)=X0
      TX(NA+2)=TX(NA+1)
      TY(NA+2)=YFEED
      TY(NA+3)=YFEED*MULT(K)
      TX(NA+3)=TX(NA+1)*MULT(K)
      TY(NA+4)=TY(1)/MULT(K)
      TX(NA+4)=TX(1)/MULT(K)
      NI=NA+4
      IF(K .EQ. 2)  GO TO 120
      CALL SETPLT(TX,TY,NI,'LOG ','MECH',-1.,-1.,BUFFER,4000,'BARKER$')
      GO TO 150
  120 CALL SETPLT(TX,TY,NI,'LINE','MECH',-1.,-1.,BUFFER,4000,'BARKER$')
  150 NI=NA+2
      CALL LINPLT(TX,TY,NI,-1,0)
      NI=NA+4
      CALL CNVRT(TIX,TIY,TX,TY,NI)
      DO 200 I=1,NA
  200 CALL QQLINE(TX(I),TY(I+1),TX(I+1),TY(I+2),0)
      DO 300 I=2,NA
      CALL QQLINE(TX(I),TY(I),TX(I),TY(I+1),0)
  300 CALL QQLINE(TX(I),TY(I+1),TX(I+1),TY(I+1),0)
      CALL TITLE(0.4,'KALC STUDIES$','X$','Y$')
      CALL QQQSYM(8.5,10.1,0.25,' ABS',0.0,4)
      CALL QQQSYM(9.0,0.25,0.25,NAME(K),0.0,4)
      CALL QQQSYM(0.1,10.1,0.2,JBNUM,0.0,8)
      RETURN
      END
      SUBROUTINE CNVRT(TIX,TIY,TX,TY,N)
      DIMENSION TIX(100),TIY(100),TY(100),TX(100),TTT(100)
```

```
      DO 100 I=1,N
      CALL QQINCH(TX(I),TY(I),TIX(I),TIY(I))
      TTT(I)=TX(I)
      TX(I)=TIX(I)
      TIX(I)=TTT(I)
      TTT(I)=TY(I)
      TY(I)=TIY(I)
      TIY(I)=TTT(I)
  100 CONTINUE
      RETURN
      END
      REAL FUNCTION F(X)
      COMMON /BM3/ XM3,B3
      F=X*XM3+B3
      RETURN
      END
      BLOCK DATA
      COMMON /XYT/ TX(100),TY(100),K,NAME(2),KOLM(2),JBNUM(2)
      DATA TX/100*0./,TY/100*0./,NAME/' KR ',' O2 '/,KOLM/' FRC',' STR'/
      END
//LKED.SYSLIB DD DSNAME=LABLIB,DISP=SHR
// DD DSNAME=A8.G1.P31394.C10207.CHANDLER.GRAPHICS,
// VOLUME=REF=ZZZZZZ,DISP=SHR
//GO.PLOTTAPE DD UNIT=TAPE7,LABEL=(,NL),DISP=OLD,VOLUME=SER=25
//GO.FT51F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=1100)
//GO.FT05F001 DD *
42      8       10      27      30       2        1.792E-07 1.768E-01
      1     7.9405E-10     6.6738E-09
      2     1.1843E-C9     9.9571E-09
      3     1.7458E-09     1.4683E-08
      4     2.5630E-09     2.1564E-08
      5     3.7656E-09     3.1693E-08
      6     5.5567E-09     4.6784E-08
      7     8.2587E-09     6.9558E-08
      8     1.2393E-08     1.0442E-07
      9     4.1813E-08     3.9703E-07
     10     1.3995E-08     1.2882E-07
     11     1.5993E-08     1.4692E-07
     12     1.7941E-08     1.6448E-07
     13     1.9536E-08     1.7879E-07
     14     2.0633E-08     1.8858E-07
     15     2.1246E-08     1.9401E-07
     16     2.1464E-08     1.9589E-07
     17     2.1378E-08     1.9503E-07
     18     2.1051E-08     1.9201E-07
     19     2.0515E-08     1.8709E-07
     20     1.9772E-08     1.8031E-07
     21     1.8807E-08     1.7151E-07
     22     1.7585E-08     1.6035E-07
     23     1.6052E-08     1.4638E-07
     24     1.4141E-08     1.2895E-07
     25     1.1763E-08     1.0726E-07
     26     8.8054E-C9     8.0293E-08
     27     5.1299E-09     4.6777E-08
     28     1.0231E-06     1.0321E-05
     29     1.1990E-07     1.2095E-06
     30     1.4727E-08     1.4856E-07
     31     1.1276E-08     1.1375E-07
```

32	8.6060E-09	8.6811E-08
33	6.5398E-09	6.5969E-08
34	4.9411E-09	4.9842E-08
35	3.7041E-09	3.7364E-08
36	2.7470E-09	2.7709E-08
37	2.0065E-09	2.0240E-08
38	1.4335E-09	1.4460E-08
39	9.9015E-10	9.9879E-09
40	6.4713E-10	6.5277E-09
41	3.8172E-10	3.8505E-09
42	1.7637E-10	1.7791E-09
1	6.4619E-03	1.1716E-01
2	6.7149E-03	1.2191E-01
3	6.9573E-03	1.2646E-01
4	7.1949E-03	1.3094E-01
5	7.4326E-03	1.3543E-01
6	7.6764E-03	1.4005E-01
7	7.9323E-03	1.4491E-01
8	8.2081E-03	1.5017E-01
9	1.9701E-02	4.4285E-01
10	6.0446E-03	1.2335E-01
11	4.7705E-03	9.6673E-02
12	3.4769E-03	6.9970E-02
13	2.3554E-03	4.7122E-02
14	1.5055E-03	2.9986E-02
15	9.2326E-04	1.8335E-02
16	5.5101E-04	1.0922E-02
17	3.2326E-04	6.4000E-03
18	1.8762E-04	3.7119E-03
19	1.0812E-04	2.1383E-03
20	6.1965E-05	1.2252E-03
21	3.5313E-05	6.9811E-04
22	1.9971E-05	3.9479E-04
23	1.1156E-05	2.2053E-04
24	6.0971E-06	1.2052E-04
25	3.1947E-06	6.3147E-05
26	1.5304E-06	3.0250E-05
27	5.7616E-07	1.1388E-05
28	5.3368E-05	1.2004E-03
29	3.3957E-06	7.6361E-05
30	3.0057E-07	6.7589E-06
31	1.0432E-07	2.3459E-06
32	3.6209E-08	8.1424E-07
33	1.2568E-08	2.8261E-07
34	4.3618E-09	9.8083E-08
35	1.5137E-09	3.4039E-08
36	5.2519E-10	1.1810E-08
37	1.8210E-10	4.0948E-09
38	6.3012E-11	1.4169E-09
39	2.1681E-11	4.8753E-10
40	7.3350E-12	1.6494E-10
41	2.3559E-12	5.2977E-11
42	6.2773E-13	1.4116E-11

/*
//

ORNL/TM-5457
UC-77 — Gas-Cooled Reactor Technology

INTERNAL DISTRIBUTION

1.	R. E. Barker	45.	L. E. McNeese
2-3.	R. A. Bradley	46-50.	J. C. Mullins
4.	R. E. Brooksbank	51.	J. P. Nichols
5.	K. B. Brown	52-53.	K. J. Notz
6.	W. D. Burch	54.	R. L. Philippone
7.	W. L. Carter	55.	H. Postma
8.	J. H. Coobs	56.	A. D. Ryon
9.	F. L. Culler	57.	C. D. Scott
10.	F. E. Dearing	58.	J. D. Sease
11.	R. S. Eby	59.	J. W. Snider
12.	D. E. Ferguson	60.	M. J. Stephenson
13.	L. M. Ferris	61.	D. B. Trauger
14.	M. J. Feldman	62.	W. E. Unger
15.	C. W. Forsberg	63.	V. C. A. Vaughen
16-21.	V. L. Fowler	64.	B. L. Vondra
22.	T. M. Gilliam	65.	M. E. Whatley
23-32.	R. W. Glass	66-67.	R. G. Wymer
33.	W. S. Groenier	68.	O. O. Yarbro
34.	P. A. Haas	69-70.	Central Research Library
35.	F. E. Harrington	71.	Document Reference Section
36.	B. A. Hannaford	72-77.	Laboratory Records
37.	D. J. Inman	78.	Laboratory Records, R.C.
38-39.	P. R. Kasten	79.	ORNL Patent Section
40.	R. K. Kibbe	80.	W. K. Davis (consultant)
41.	J. A. Klein	81.	J. C. Frye (consultant)
42.	K. H. Lin	82.	C. H. Ice (consultant)
43-44.	A. L. Lotts	83.	R. B. Richards (consultant)

EXTERNAL DISTRIBUTION

84. Research and Technical Support Division, ERDA, ORO
85. Director, Reactor Division, ERDA, ORO
86-87. Director, Nuclear Fuel Cycle and Production, ERDA, Washington, D.C. 20545
88-89. Director, Nuclear Research and Applications, ERDA, Washington, D.C. 20545
90-256. Given distribution as shown in TID-4500 under Gas-Cooled Reactor Technology category (25 copies - NTIS)
257-294. U.S./German HTR Research Exchange Arrangement

OAK RIDGE NATIONAL LABORATORY
OPERATED BY
UNION CARBIDE CORPORATION
NUCLEAR DIVISION

POST OFFICE BOX X
OAK RIDGE, TENNESSEE 37830

November 12, 1976

To: Recipients of Subject Report

Report No.: ORNL/TM-5457 Classification: Unclassified

Author(s): R. W. Glass and R. E. Barker

Subject: A Computer Model for the KALC Process Studies in the ORGDP Off-Gas Decontamination Pilot Plant

Several changes should be made in the KALC/ORGDP Program listed in Appendix A of the subject report. Please make pen or pencil corrections on your copy in accordance with the instructions given on the attached sheet. Apologies are extended to the reader for his inconvenience in this matter.

J. L. Langford
J. L. Langford, Supervisor
Laboratory Records Department
Information Division

JLL:we

Attachment (1)

UCN-6638
(3 5-65)

CHANGES TO BE MADE IN ORNL/TM-5457

Page 27, line 40 should read:

```
H=137.8-0.729036*TK-1.42469E-3*TK*TK-8.115044E-5*TK**3;
```

Page 30, lines 16 and 17 should be deleted.

Page 30, line 39 should read:

```
ELSE DØ;
```

Page 30, line 40 should be deleted.

Page 31, line 1 should read:

```
TSM=0; TSM(1)=1;
```

Page 31, line 2 should be deleted.

Page 32, line 50 should read:

```
W2(I,II)=A(I,II)*G(II,1);
```

Page 34, line 47 should read:

```
PUT SKIP EDIT(I,' K(EQ)=',(YY(I,J)/XX(I,J)DØ J=1 TØ M))
```

Page 34, line 52 should read:

```
PUT SKIP EDIT(I,'-',I+1,'L/V(ØP)=',L(I,I)/V(I+1,I+1))
```

Page 35, line 1 should read:

```
PUT SKIP EDIT(I,'-',I+1,'K(ØP)=',(YY(I+1,J)/
```

Page 35, line 2 should read:

```
XX(I,J) DØ J=1 TØ M))
```

Page 35, line 9 should read:

```
PUT SKIP EDIT(I,'-',I+1,'ABF(ØP)=',((L(I,I)/V(I+1,I+1))/
```

Page 35, line 10 should read:

```
(YY(I+1,J)/XX(I,J))
```