

Lawrence Berkeley National Laboratory

Recent Work

Title

NUMERICAL SOLUTION OF COUPLED, ORDINARY DIFFERENTIAL EQUATIONS

Permalink

<https://escholarship.org/uc/item/7qj7k1bb>

Author

Newman, John.

Publication Date

1967-08-01

C.2

University of California

Ernest O. Lawrence Radiation Laboratory

NUMERICAL SOLUTION OF COUPLED,
ORDINARY DIFFERENTIAL EQUATIONS

John Newman

August 1967

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.
For a personal retention copy, call
Tech. Info. Division, Ext. 5545*

Berkeley, California

308

UCRL-17739
C.2

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal 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. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

UCRL-17739
UC-32 Math. and Comp.
TID-4500 (50th Ed.)

UNIVERSITY OF CALIFORNIA
Lawrence Radiation Laboratory
Berkeley, California

AEC Contract No. W-7405-eng-48

NUMERICAL SOLUTION OF COUPLED,
ORDINARY DIFFERENTIAL EQUATIONS

John Newman

August 1967

Printed in the United States of America
Available from
Clearinghouse for Federal Scientific and Technical Information
National Bureau of Standards, U.S. Department of Commerce
Springfield, Virginia 22151
Price: Printed Copy \$3.00; Microfiche \$0.65

Contents

	page
Abstract	1
Introduction	2
Solution of Coupled, Linear, Difference Equations	4
Linearization of Nonlinear Problems	7
Distillation	9
Conclusions	10
Acknowledgement	11
References	12
Appendix A. Program for coupled, linear difference equations	13
Appendix B. Effect of ionic migration on limiting currents	16
Appendix C. Distillation calculations by the Thiele- Geddes method	19
Appendix D. Distillation calculations by linearization	28

Numerical Solution of Coupled,
Ordinary Differential Equations

John Newman

Inorganic Materials Research Division,
Lawrence Radiation Laboratory, and
Department of Chemical Engineering
University of California, Berkeley

August, 1967

Abstract

A wide variety of problems involving ordinary differential equations can be linearized about a trial solution and then put into finite difference form. The resulting coupled, tridiagonal matrices can be solved readily on a high speed, digital computer, and the nonlinear problem can then be solved by iteration.

Introduction

Many problems in the physical sciences can be reduced to ordinary differential equations. The availability of high-speed digital computers and a generalized method of solution allows many such problems to be treated without the drastic approximations frequently needed to obtain analytic solutions.

Many of these problems are nonlinear and involve several dependent variables, but by a proper linearization of such problems a convergent iteration scheme frequently results, although convergence cannot generally be assured. Consequently, we shall first show how to solve coupled, linear differential equations. This will be followed by an illustration of the linearization method for a particular problem.

Types of problems

The author has had many opportunities to apply the present method of calculation. The effect of migration on limiting currents (Newman (1966)) results from the interaction of several diffusing ionic species through the electric potential. Okada and others (1959) discovered a non-iterative method for such problems. However, the present method is more generally applicable, and no special procedure has to be devised for each case.

In hydrodynamics and mass transfer, a transformation will frequently reduce the applicable partial differential equations to ordinary differential equations. This is true for mass transfer to a rotating disk, to a growing mercury drop, and to "penetration" problems, that is, transient diffusion into a stagnant, semi-infinite region. Thus, these several cases could be treated to determine the effect of ionic migration on limiting currents.

However, in these situations one can also obtain a detailed verification of the mass-transfer equations, taking into account variation of physical properties and a non-zero interfacial velocity as well as ionic migration (Newman and Hsueh (1967), Hsueh and Newman (1967)). The numerical method could be applied to multicomponent diffusion problems, but the necessary knowledge of the concentration dependence of the multicomponent diffusion coefficients is lacking in most cases.

The one-dimensional, macroscopic model of porous electrodes (Newman and Tobias (1962), Grens and Tobias (1964)) provides a large class of problems involving coupled, nonlinear differential equations, at least in the steady or pseudo-steady state and also just after the current has been applied but before the composition has begun to change appreciably.

Occasionally one runs into a single, nonlinear differential equation. The linearization method outlined here was used to solve the first hydrodynamic equation of the Blasius series

$$f_1'^2 - f_1 f_1'' = 1 + f_1''' \quad (1)$$

by Newman (1966), although this problem had already been solved by other methods. A nonlinear equation encountered by Smyrl and Newman (1967) was also attacked in this manner.

Two problems currently being worked on are free convection at the limiting current in the presence of a supporting electrolyte and the calculation of liquid junction potentials. For free convection the concentration of the supporting electrolyte affects the density as well as the ionic migration of the reactant. Models of liquid junctions such as restricted diffusion and free diffusion from an initially sharp boundary also yield to the present method of analysis.

The analysis of a distillation process involving a number of stages produces a set of equations representing material balances and enthalpy balances. These equations are similar to the finite difference approximation to differential equations in that each equation can involve unknown quantities for the stage above and the stage below the one in question. The equations are nonlinear and coupled and involve unknowns like the temperature, compositions, and flow rates for each stage. The present method is applicable to such problems. For a restricted class of distillation problems, more efficient methods have already been developed (Newman (1963)), but the present method is more general and allows greater flexibility.

Solution of Coupled, Linear, Difference Equations

At each point j , except $j = 1$ and $j = j_{\max}$, there are n equations of the form

$$\sum_{k=1}^n A_{i,k}(j)C_k(j-1) + B_{i,k}(j)C_k(j) + D_{i,k}(j)C_k(j+1) = G_i(j) . \quad (2)$$

The unknowns are C_1 through C_n . The subscript i denotes the equation number, and each of the equations can involve all of the unknowns C_k , through the sum. $A_{i,k}$, $B_{i,k}$, and $D_{i,k}$ are coefficients of the unknowns at the mesh points $j-1$, j , and $j+1$, and G_i contains all terms independent of the unknowns C_k . The equations are linear, that is, the coefficients A , B , and D are independent of the unknowns.

Equation (2) is sufficiently general to cover the finite-difference form of most coupled, linear, differential equations. For example, a second derivative is frequently approximated by

$$C_k'' = \frac{C_k(j+1) + C_k(j-1) - 2C_k(j)}{h^2} + o(h^2), \quad (3)$$

where h is the mesh distance.

At $j=1$ the equations are

$$\sum_{k=1}^n B_{i,k}(j) C_k(j) + D_{i,k}(j) C_k(j+1) + X_{i,k}(j) C_k(j+2) = G_i(j). \quad (4)$$

There is no point below $j=1$, so $A_{i,k}$ does not appear. However, in order to allow the treatment of complex boundary conditions, the third term involving the unknowns at $j+2$ has been added. In this case, point 1 is an image point. In other words, it is occasionally desirable to use the difference approximation to the differential equation at the boundary point even though this involves an imaginary image point outside the domain of interest. The finite difference form of the boundary conditions is then written as an extra set of equations which should be sufficient to eliminate the image points.

For $j=1$, let $C_k(j)$ take the form

$$C_k(j) = \xi_k(j) + \sum_{\ell=1}^n E_{k,\ell}(j) C_\ell(j+1) + x_{k,\ell} C_\ell(j+2). \quad (5)$$

Substitution into equation (4) shows that ξ_k , $E_{k,\ell}$, and $x_{k,\ell}$ satisfy the equations

$$\left. \begin{aligned} \sum_k B_{i,k}(j) \xi_k(j) &= G_i(j) \\ \sum_k B_{i,k}(j) E_{k,\ell}(j) &= -D_{i,\ell}(j) \\ \sum_k B_{i,k}(j) x_{k,\ell}(j) &= -X_{i,\ell} \end{aligned} \right\} j = 1, \quad (6)$$

which all have the same matrix of coefficients $B_{i,k}$ and which can be readily solved.

For the remaining points, except $j=j_{\max}$, the unknowns C_k assume the form

$$C_k(j) = \xi_k(j) + \sum_{\ell=1}^n E_{k,\ell}(j) C_\ell(j+1) . \quad (7)$$

Substitution of equation (7) into equation (2) to eliminate first $C_k(j-1)$ and then $C_k(j)$ and setting the remaining coefficient of each $C_k(j+1)$ equal to zero yield a set of equations for the determination of ξ_k and $E_{k,\ell}$:

$$\sum_k b_{i,k}(j) \xi_k(j) = G_i(j) - \sum_\ell A_{i,\ell}(j) \xi_\ell(j-1) , \quad (8)$$

$$\sum_k b_{i,k}(j) E_{k,m}(j) = - D_{i,m}(j) , \quad (9)$$

where

$$b_{i,k}(j) = B_{i,k}(j) + \sum_\ell A_{i,\ell}(j) E_{\ell,k}(j-1) . \quad (10)$$

The solution of these linear equations at each point j is again straightforward, but the point to the left must be calculated first since $\xi_k(j-1)$ appears on the right side of equation (8) and $E_{\ell,k}(j-1)$ appears in the matrix of coefficients $b_{i,k}$. The equations for $j=2$ actually take a slightly different form since equation (5) should be used instead of equation (7) to eliminate $C_k(1)$ from equation (2).

At $j=j_{\max}$ the equations are

$$\sum_{k=1}^n Y_{i,k} C_k(j-2) + A_{i,k}(j) C_k(j-1) + B_{i,k}(j) C_k(j) = G_i(j) , \quad (11)$$

where the coefficients $Y_{i,k}$ allow the introduction of complex boundary

conditions at the right in the same way that $X_{i,k}$ does at the left. If $C_k(j-2)$ and $C_k(j-1)$ are eliminated by means of equation (7), then the values of $C_k(j)$ can be determined from the resulting equations

$$\sum_k b_{i,k}(j)C_k(j) = G_i(j) - \sum_{\ell} Y_{i,\ell} \xi_{\ell}(j-2) - \sum_{\ell} \left[A_{i,\ell}(j) + \sum_m Y_{i,m} E_{m,\ell}(j-2) \right] \xi_{\ell}(j-1) \quad (12)$$

where

$$b_{i,k}(j) = B_{i,k}(j) + \sum_{\ell} \left[A_{i,\ell}(j) + \sum_m Y_{i,m} E_{m,\ell}(j-2) \right] E_{\ell,k}(j-1) \quad (13)$$

Having in hand $C_k(j)$ for $j=j_{\max}$, one is now in a position to determine $C_k(j)$ in reverse order in j from equation (7) and finally to determine $C_k(1)$ from equation (5).

Linearization of Nonlinear Problems

The preceding section shows how to solve coupled, linear, difference equations. Often, however, one is faced with a set of coupled, non-linear, differential equations. A variety of experience shows that iteration with a linearized form of the equations frequently converges to the correct result.

The equations

$$D_i c_i'' + z_i u_i F (c_i \Phi'' + c_i' \Phi') = 0 \quad (14)$$

represent Fick's second law for one dimensional, steady migration and diffusion of ionic species in a stagnant medium (Newman (1966)). Here c_i is the concentration, Φ is the electric potential, D_i is the diffusion coefficient, u_i is the mobility, z_i is the charge number, and F is

Faraday's constant. The diffusion term is already linear, but the migration terms are nonlinear. Equations (14) can be linearized by assuming that one has nearly correct values of c_i and Φ , say c_i^0 and Φ^0 , and that the change in these quantities during one iteration is relatively small. Then we can write, for example,

$$c_i \Phi'' = (c_i^0 + \Delta c_i)(\Phi^0'' + \Delta \Phi'') \approx c_i^0 \Phi^0'' + \Phi^0'' \Delta c_i + c_i^0 \Delta \Phi'' , \quad (15)$$

where the term quadratic in the small quantities Δc_i and $\Delta \Phi''$ has been neglected. Replacing Δc_i by $c_i - c_i^0$ and $\Delta \Phi''$ by $\Phi'' - \Phi^0''$, we can then write

$$c_i \Phi'' \approx c_i \Phi^0'' + c_i^0 \Phi'' - c_i^0 \Phi^0'' . \quad (16)$$

The linearized form of equations (14) can now be written as

$$D_i c_i'' + z_i u_i F (c_i \Phi^0'' + c_i^0 \Phi'' + c_i' \Phi^{0'} + c_i^{0'} \Phi') = z_i u_i F (c_i^0 \Phi^0'' + c_i^{0'} \Phi^{0'}) . \quad (17)$$

These form a set of coupled, linear, differential equations. The finite difference form is

$$\begin{aligned} & c_i(j-1)[D_i - \frac{1}{2}z_i u_i F h \Phi^{0'}] + c_i(j)[-2D_i + z_i u_i F h^2 \Phi^0''] \\ & + c_i(j+1)[D_i + \frac{1}{2}z_i u_i F h \Phi^{0'}] + \Phi(j-1)[z_i u_i F c_i^0 - \frac{1}{2}z_i u_i F h c_i^{0'}] \\ & + \Phi(j)[-2z_i u_i F c_i^0] + \Phi(j+1)[z_i u_i F c_i^0 + \frac{1}{2}z_i u_i F h c_i^{0'}] \\ & = z_i u_i F [c_i^0 \Phi^0'' h^2 + c_i^{0'} \Phi^{0'} h^2] . \end{aligned} \quad (18)$$

The coefficients $A_{i,k}$ in equation (2) then become the coefficients of c_i and Φ at $j-1$, etc., and Φ becomes one of the unknowns C_k . The term on the right in equation (18) becomes $G_i(j)$.

The one remaining equation for this problem would be the electro-neutrality relation

$$\sum_k z_k c_k = 0 , \quad (19)$$

which is already linear and does not involve the unknowns at $j-1$ or $j+1$.

For porous electrodes the equations might involve a reaction term involving exponentials, like

$$J = e^{-\phi} - c_3 e^{\phi}, \quad (20)$$

where ϕ is a dimensionless potential and c_3 is a reactant concentration. The linearized form of this equation would be

$$J \approx -\phi [e^{-\phi^0} + c_3^0 e^{\phi^0}] - c_3^0 e^{\phi^0} + (1 + \phi^0) e^{-\phi^0} + c_3^0 \phi^0 e^{\phi^0}. \quad (21)$$

A remark is in order for first and third order differential equations. For the purpose of computation, the third order equation (1) could be replaced by two second order equations (with $C_1' = f_1$ and $C_2 = f_1'$) or a first order equation and a second order equation (with $C_1 = f_1$ and $C_2 = f_1'$). In this way the finite difference forms still involve only the points at $j-1$, j , and $j+1$. For a first order equation it is probably better to use a backward difference rather than a central difference. The order of the approximation will still be h^2 if the coefficient takes on its average value:

$$KC_1' \approx \frac{1}{2} [K(j) + K(j-1)] [C_1(j) - C_1(j-1)] / h + O(h^2). \quad (22)$$

Distillation

The numerical procedure described here was developed primarily for problems in hydrodynamics and electrolytic mass transfer. However, its application to distillation calculations has also been tested and compared with the method developed by Newman (1963). One first notices a difference in the storage requirements. The present method requires

at least the storage of the E array. A large number of components is frequently involved in distillation, and the E array requires 55,200 storage locations for the calculation of 20 components and 100 stages, compared to 10,000 locations for the largest array in the earlier method.

The calculation time was perceptibly longer for the more general method. The earlier program is a slight modification of the Thiele-Geddes method and takes advantage of the fact that one set of variables, the temperatures on the stages, controls all the compositions if the flow rates are fixed. The subsequent adjustment of the flow rates to satisfy the enthalpy balances is less certain, whereas the general method simultaneously treats the enthalpy balances in the linearization.

The general method has two distinct advantages. First, it is easier to introduce composition-dependent activity coefficients for the liquid phase without upsetting the convergence characteristics. Second, there can be greater flexibility in the specification of conditions of column operation. In the standard extensions of the Thiele-Geddes method, the bottom-product flow rate and the reflux flow rate are specified. With the more general method one should be able to replace one or both of these conditions by conditions such as a given mole fraction of a component in the top or bottom product, a given product amount of a component, a fixed reboiler or condenser temperature, or a fixed reboiler or condenser heat load.

Conclusions

The procedure outlined here for solving coupled, nonlinear, difference equations by linearization and subsequent iteration has proved

useful in a number of problems. For a given problem it may be possible to devise a more efficient method, but usually with considerable expense of personnel effort. The method outlined, it should be emphasized, is very general and flexible.

Acknowledgement

This work was supported by the United States Atomic Energy Commission.

References

Edward A. Grens II and Charles W. Tobias. "Analysis of the Dynamic Behavior of Flooded Porous Electrodes." Ber. Bunsenges. Physik. Chem., 68, 236-249 (1964).

L. Hsueh and J. Newman. "Mass Transfer and Polarization at a Rotating Disk Electrode." Electrochim. Acta, 12, 429-438 (1967).

John S. Newman and Charles W. Tobias. "Theoretical Analysis of Current Distribution in Porous Electrodes." J. Electrochem. Soc., 109, 1183-1191 (1962).

John S. Newman. "Temperature Computed for Distillation." Hydrocarbon Process. Petrol. Refiner, 42, no.4, 141-144 (1963).

John Newman. "Blasius Series for Heat and Mass Transfer." Intern. J. Heat Mass Transfer, 9, 705-709 (1966).

John Newman. "Effect of Ionic Migration on Limiting Currents." Ind. Eng. Chem. Fundamentals, 5, 525-529 (1966).

J. Newman and L. Hsueh. "The Effect of Variable Transport Properties on Mass Transfer to a Rotating Disk." Electrochim. Acta, 12, 417-427 (1967).

Shinzo Okada, Shiro Yoshizawa, Fumio Hine, and Kameo Asada. "Effect of Migration on Polarographic Limiting Current." J. Electrochem. Soc. Japan (Overseas Ed.), 27, No.1-3, E51-E52 (1959).

William H. Smyrl and John Newman. "Double Layer Structure at the Limiting Current." Trans. Faraday Soc., 63, 207-216 (1967).

Appendix A. Program for coupled, linear difference equations.

The next two pages give subroutines implementing the solution of coupled, linear, difference equations as outlined in the body of this report. To save storage space, the arrays A, B, D, and G are to be supplied by the main program for each value of j , and the subroutine BAND(J) is to be called for each value of j . The values of X are to be supplied for $j=1$, and the values of Y for $j=NJ=j_{\max}$. The values of X are not to be disturbed for any intermediate calculations between $j=1$ and $j=j_{\max}$. The dimensions have been selected for $n=6$, the number of unknown variables at each mesh point, and $j_{\max}=103$, the number of mesh points including image points, if any. These can be changed appropriately for a particular problem. The second dimension of the D array is to be $2n+1$, although values need to be supplied only for the original n by n array. The second dimension of the E array is $n+1$ since ξ_k is stored here. An example of a main program for the effect of ionic migration on limiting currents is given in appendix B.

The subroutine MATINV is used to solve the linear equations (6), (8), (9), and (12) which arise at each value of j . If, at any value of j , the determinant of the matrix of these equations is found to be zero, this fact is reported in the output. This usually indicates that all the equations have not been programmed or that they are not all independent. It can also indicate that the equations for $j=1$ are not sufficient to determine the image points although the equations for $j=1$ and $j=2$ would be sufficient to determine both the boundary point and the image point. In rare instances it may indicate that the trial solution is inadequate and gives a zero determinant.

```

SUBROUTINE BAND(J)
DIMENSION C(6,103),G(6),A(6,6),B(6,6),D(6,13),E(6,7,103),X(6,6),
1Y(6,6)
COMMON A,B,C,D,G,X,Y,N,NJ
101 FORMAT (15HODETERM=0 AT J=,I4)
IF (J-2) 1,6,8
1 NP1= N + 1
DO 2 I=1,N
D(I,2*N+1)= G(I)
DO 2 L=1,N
LPN= L + N
2 D(I,LPN)= X(I,L)
CALL MATINV(N,2*N+1,DETERM)
IF (DETERM) 4,3,4
3 PRINT 101, J
4 DO 5 K=1,N
E(K,NP1,1)= D(K,2*N+1)
DO 5 L=1,N
E(K,L,1)= -D(K,L)
LPN= L + N
5 X(K,L)= -D(K,LPN)
RETURN
6 DO 7 I=1,N
DO 7 K=1,N
DO 7 L=1,N
7 D(I,K)= D(I,K) + A(I,L)*X(L,K)
8 IF (J-NJ) 11,9,9
9 DO 10 I=1,N
DO 10 L=1,N
G(I)= G(I) - Y(I,L)*E(L,NP1,J-2)
DO 10 M=1,N
10 A(I,L)= A(I,L) + Y(I,M)*E(M,L,J-2)
11 DO 12 I=1,N
D(I,NP1)= -G(I)
DO 12 L=1,N
D(I,NP1)= D(I,NP1) + A(I,L)*E(L,NP1,J-1)
DO 12 K=1,N
12 B(I,K)= B(I,K) + A(I,L)*E(L,K,J-1)
CALL MATINV(N,NP1,DETERM)
IF (DETERM) 14,13,14
13 PRINT 101, J
14 DO 15 K=1,N
DO 15 M=1,NP1
15 E(K,M,J)= -D(K,M)
IF (J-NJ) 20,16,16
16 DO 17 K=1,N
17 C(K,J)= E(K,NP1,J)
DO 18 JJ=2,NJ
M= NJ - JJ + 1
DO 18 K=1,N
C(K,M)= E(K,NP1,M)
DO 18 L=1,N
18 C(K,M)= C(K,M) + E(K,L,M)*C(L,M+1)
DO 19 L=1,N
DO 19 K=1,N
19 C(K,1)= C(K,1) + X(K,L)*C(L,3)
20 RETURN
END

```

```
SUBROUTINE MATINV(N,M,DETERM)
DIMENSION ID(6),B(6,6),D(6,13),A(6,6),C(6,103)
COMMON A,B,C,D
DETERM= 1.0
DO 1 I=1,N
1 ID(I)= 0.0
DO 18 NN=1,N
BMAX= 0.0
DO 6 I=1,N
IF (ID(I)) 2,2,6
2 DO 5 J=1,N
IF (ID(J)) 3,3,5
3 IF (ABSF(B(I,J)) - BMAX) 5,5,4
4 BMAX= ABSF(B(I,J))
IROW= I
JCOL= J
5 CONTINUE
6 CONTINUE
IF (BMAX) 7,7,8
7 DETERM= 0.0
RETURN
8 ID(JCOL)= 1
IF (JCOL-IROW) 9,12,9
9 DO 10 J=1,N
SAVE= B(IROW,J)
B(IROW,J)= B(JCOL,J)
10 B(JCOL,J)= SAVE
DO 11 K=1,M
SAVE= D(IROW,K)
D(IROW,K)= D(JCOL,K)
11 D(JCOL,K)= SAVE
12 F= 1.0/B(JCOL,JCOL)
DO 13 J=1,N
13 B(JCOL,J)= B(JCOL,J)*F
DO 14 K=1,M
14 D(JCOL,K)= D(JCOL,K)*F
DO 18 I=1,N
IF (I-JCOL) 15,18,15
15 F= B(I,JCOL)
DO 16 J=1,N
16 B(I,J)= B(I,J) - F*B(JCOL,J)
DO 17 K=1,M
17 D(I,K)= D(I,K) - F*D(JCOL,K)
18 CONTINUE
RETURN
END
```

Appendix B. Effect of ionic migration on limiting currents.

This program gives an example of the use of the subroutines of appendix A to solve a particular problem, that of the effect of ionic migration on limiting currents (Newman (1966)). Each iteration begins at statement 8 and involves setting up the coefficients A, B, D, and G for each value of j followed by calling subroutine BAND(J) for each value of j. In the program U(I) is proportional to $z_i u_i$, and the electric potential is the n-th unknown variable, the other unknowns being the n-1 species concentrations. MODE is 1 for a Nernst stagnant diffusion layer, 2 for a growing mercury drop or unsteady diffusion into a stagnant fluid, and 3 for a rotating disk. H is the mesh size, and CRO is the concentration of the reactant at the electrode (equal to zero at the limiting current).

```

PROGRAM MIGR(INPUT,OUTPUT)
C PROGRAM FOR EFFECT OF MIGRATION ON LIMITING CURRENT
DIMENSION A(6,6),B(6,6),C(6,103),D(6,13),G(6),X(6,6),Y(6,6),U(6),
1V(103),DIF(6),Z(6),S(6),CIN(6),REF(6)
COMMON A,B,C,D,G,X,Y,N,NJ
101 FORMAT (2I4,E8.4)
102 FORMAT (4E8.4,A6)
103 FORMAT (4HONJ=,I4,5H, H=,F6.4/34HOSPECIES U DIF Z
1 S/(3X,A6,2F8.5,2F5.1))
104 FORMAT (32H1NERNST STAGNANT DIFFUSION LAYER)
105 FORMAT (32H1GROWING DROP OR PLANE ELECTRODE)
106 FORMAT (14H1ROTATING DISK)
107 FORMAT (5E8.4)
108 FORMAT (30H0THE NEXT RUN DID NOT CONVERGE)
109 FORMAT (1H0,26X,F10.6/(3X,A6,2F9.5))
110 FORMAT (35HOSPECIES CINF CZERO AMP)
READ 101, MODE,NJ,H
GO TO (1,2,3),MODE
1 H= 1.0/(NJ-2)
CONST= 0.0
PRINT 104
GO TO 99
2 CONST= 2.0
PRINT 105
GO TO 99
3 CONST= 3.0
PRINT 106
99 READ 101, N,J,CRO
IF (N) 4,4,5
4 STOP
5 NM1= N - 1
NM2= N - 2
READ 102, (U(I),DIF(I),Z(I),S(I),REF(I),I=1,NM1)
PRINT 103, NJ,H,(REF(I),U(I),DIF(I),Z(I),S(I),I=1,NM1)
PRINT 110
98 READ 107, (CIN(I),I=1,NM1)
CIN(N)= 0.0
IF (CIN(1)) 99,6,6
6 DO 7 J=1,NJ
V(J)= CONST*(H*(NJ-J-1)**(MODE-1)*DIF(NM1)
C(NM1,J)= CRO + (CIN(NM1) - CRO)*(NJ-J-1)/(NJ-2)
C(N,J)= 0.0
DO 7 I=1,NM2
7 C(I,J)= CIN(I)
JCOUNT = 0
AMP= 0.0
8 JCOUNT= JCOUNT + 1
J= 0
DO 9 I=1,N
DO 9 K=1,N
Y(I,K)= 0.0
9 X(I,K)= 0.0
10 J= J + 1
DO 11 I=1,N
G(I)= 0.0
DO 11 K=1,N
A(I,K)= 0.0
B(I,K)= 0.0
11 D(I,K)= 0.0

```

```

      IF (J-1) 12,12,14
12 DO 13 I=1,N
      B(I,I)= 1.0
13 G(I)= CIN(I)
      CALL BAND(J)
      GO TO 10
14 DO 15 K=1,NM1
15 B(N,K)= Z(K)
      IF (J-NJ) 16,18,18
16 DO 17 I=1,NM1
      PP= U(I)/DIF(I)*(C(N,J+1)-C(N,J-1))/2.0
      PPP= U(I)/DIF(I)*(C(N,J+1)+C(N,J-1)-2.0*C(N,J))
      CP= (C(I,J+1) - C(I,J-1))/2.0
      A(I,I)= - 1.0 + PP/2.0 - H*V(J)/2.0/DIF(I)
      B(I,I)= 2.0 - PPP
      D(I,I)= - 1.0 - PP/2.0 + H*V(J)/2.0/DIF(I)
      A(I,N)= U(I)/DIF(I)*(CP/2.0 - C(I,J))
      B(I,N)= 2.0*U(I)/DIF(I)*C(I,J)
      D(I,N)= - U(I)/DIF(I)*(CP/2.0 + C(I,J))
17 G(I)= - PPP*C(I,J) - PP*CP
      CALL BAND(J)
      GO TO 10
18 DO 19 I=1,NM2
      PP= U(I)/DIF(I)*(C(N,NJ)-C(N,NJ-2))
      Y(I,I)= - 1.0
      A(I,I)= PP
      B(I,I)= 1.0
      Y(I,NM1)= S(I)*DIF(NM1)/S(NM1)/DIF(I)
      B(I,NM1)= - Y(I,NM1)
      Y(I,N)= (S(I)/S(NM1)*U(NM1)*CRO-U(I)*C(I,NJ-1))/DIF(I)
      B(I,N)= - Y(I,N)
19 G(I)= PP*C(I,NJ-1)
      G(NM1)= CRO
      A(NM1,NM1)= 1.0
      CALL BAND(J)
      AMPO= AMP
      AMP= (U(NM1)*CRO*(C(N,NJ-2)-C(N,NJ))+DIF(NM1)*(C(NM1,NJ-2)-C(NM1,
      NJ)))/2.0/H/(GIN(NM1)-CRO)/DIF(NM1)
      IF (ABS(AMP-AMPO)-0.00001 *ABS(AMP)) 22,22,20
20 IF (JCOUNT-10) 8,8,21
21 PRINT 108
22 PRINT 109, AMP,(REF(I),C(I,1),C(I,NJ-1),I=1,NM1)
      GO TO 98
      END

```


Appendix C. Distillation calculations by the Thiele-Geddes method.*

This section contains the FORTRAN statements for a tested computer program employing the method discussed in "Temperature Computed for Distillation" (Newman (1963)). The program is designed to handle distillation calculations for a fractionating column with as many as 86 plates and 20 components, where the equilibrium constants may be considered to be independent of composition. A total or a partial condenser may be used, and the possibility of a feed, a side draw of liquid, and a side draw of vapor on each stage has been included. A two-product condenser can be achieved by a liquid draw from the condenser.

Equilibrium constants in the form of either power series in temperature or exponential functions may be used. These are put into a subroutine so that they can be changed without much trouble.

This program is improved over the earlier program* in the calculation of compositions and in the re-estimation of flow rates. For the compositions, with the flow rates and temperature fixed as in the Thiele-Geddes method, the standard method for a tri-diagonal matrix works for both light and heavy components and avoids the necessity for calculating heavy components from the bottom, and vice versa. The adjustment of the flow rates from the heat balances was also a source of difficulty, which arises because the molal enthalpies depend on temperature and composition and really are not known. A crude but effective method of interpolation, extrapolation, and damping has therefore been added.

* This section obsoletes and supersedes the report "Computer Program for Distillation" dated April 26, 1963.

The basic problem should specify all feeds, number and arrangement of plates, and the amount of reflux, bottom product, and side-stream draws. For problems with other variables specified, a specification subroutine would have to be used. Only a dummy subroutine of this type has been written. The arrays SPECS and IN are intended for transmitting data to the SPEC subroutine without modifying the main program.

The significance of the input data is outlined below.

NC - number of components.

NS - number of stages including reboiler and condenser.

NF - number of feeds.

JCOTYP = 0 for partial condenser, = 2 for total condenser.

KTYP = 0 for exponential equilibrium constants, = 2 for power-series equilibrium constants.

LIM - maximum number of times that the temperature-correction procedure can be applied.

NDRAW - number of stages on which side draws occur.

IN(1) to IN(5) - not used.

NPROB - problem number for identification of output.

AK, BK, CK, DK are parameters in the expression for equilibrium constants.

For the power-series expression,

$$K_i = AK_i + (BK_i)T + (CK_i)T^2 + (DK_i)T^3.$$

For the exponential expression,

$$K_i = \exp \left\{ \frac{AK_i}{T+DK_i} + BK_i + (CK_i)(T + DK_i) \right\}.$$

AHL, BHL, CHL are parameters in a power-series expression for the enthalpy of a liquid stream. Per mole of mixture,

$$h = \sum_i x_i (AHL_i + BHL_i T + CHL_i T^2) .$$

AHV, BHV, CHV are parameters in a power-series expression for the enthalpy of a vapor stream. Per mole of mixture

$$H = \sum_i y_i (AHV_i + BHV_i T + CHV_i T^2) .$$

AL are estimated molal flow rates of the liquid stream leaving each stage (the last one being the reflux).

T are estimated temperatures for each plate.

J is the feed stage.

HF is the total enthalpy of the feed.

FX is the feed rate for each component.

The last three are repeated for each feed stage.

JD is the number of a stage with a side draw.

SL and SV are molal flow rates for the liquid and vapor side streams.

The last three (JD, SL, and SV) are repeated for each stage with a side draw.

SUMERR - error limit on the sum of mole fractions. The sum of the mole fractions can differ from unity by no more than SUMERR.

HETERR - error limit on heat balance. The error in the heat balance on any stage, divided by the average of the condenser and reboiler loads, must be less than HETERR.

SUBCOL- subcooling of reflux for total condenser (degrees below bubble point).

DTLIM - limit on the temperature correction, degrees.

SPECS(1) to SPECS(4) - not used.

CHECK is 011111+1 in columns 65 to 72. This is used to make sure that the correct number of cards has been read.

Any number of problems may be run without stopping.

In the output, J is the stage number, T is the temperature, AL is the liquid flow (or reflux for a total condenser), SUMX is the sum of the mole fractions, X are component mole fractions. The component flow rates are then listed for the bottom product, the top product, and any side streams.

```

PROGRAM DIST(INPUT,OUTPUT)
C PROGRAM FOR FRACTIONATING COLUMN WITH SIDE-STREAM DRAWS
DIMENSION AK(20),BK(20),CK(20),DK(20),AHL(20),BHL(20),CHL(20),AHV
1(20),BHV(20),CHV(20),X(20, 88),EQ(20, 88),T( 88),AL( 88),V( 88),SL
2( 88),SV( 88),QERR( 88),OLDDL( 88),ERR( 88),HL( 88),HV( 88),A( 88,
3 88),IN(5),SPECS(4),F( 88),HF( 88),FX(20, 88)
COMMON AK,BK,CK,DK,KTYP,EQ,T,X,NC,N,QC,QR,QNET,CHECK,DL,OK,ITERAT,
1ITHEAT,ITEMP,NIT,ERR,AL,V,SL,SV,NS,NF,NDRAW,JCOTYP,IN,SPECS,F,HF,
2FX,DTLIM,SUMERR,HETERR,SUBCOL,NPROB,HL,HV,OLDDL,QERR,A
101 FORMAT (74H1COMPONENTS STAGES FEEDS COTYP KTYP LIMIT DR
1AWS INSTRUCTIONS,26X,7HPROBLEM/(13I8))
102 FORMAT (118H0 I AK(I) BK(I) CK(I) DK(I)
1 AHL(I) BHL(I) CHL(I) AHV(I) BHV(I) CHV(I)/
2(I3,7E12.4,3E11.4))
103 FORMAT (60H0 STAGE FEED ENTHALPY COMPO
1NENTS/I8,E22.7,6E15.7/(8E15.7))
104 FORMAT (40H0 STAGE LIQUID DRAW VAPOR DRAW/(I8,2E17.6))
105 FORMAT (69H0 SUMERR HETERR SUBCOL DTLIM
1 SPECS/(8E15.7))
1 FORMAT (9E8.4)
2 FORMAT (13I4)
3 READ 2, NC,NS,NF,JCOTYP,KTYP,LIM,NDRAW,(IN(I),I=1,5),NPROB
IF (NC) 98,98,99
98 STOP
99 READ 1, (AK(I),BK(I),CK(I),DK(I),AHL(I),BHL(I),CHL(I),AHV(I),
1BHV(I),CHV(I),I=1,NC)
READ 1, (AL(J),J=1,NS)
READ 1, (T(J),J=1,NS)
PRINT 101, NC,NS,NF,JCOTYP,KTYP,LIM,NDRAW,(IN(I),I=1,5),NPROB
PRINT 102, (I,AK(I),BK(I),CK(I),DK(I),AHL(I),BHL(I),CHL(I),AHV(I),
1BHV(I),CHV(I),I=1,NC)
DO 4 J=1,NS
SL(J)= 0.0
SV(J)= 0.0
F(J)= 0.0
HF(J)= 0.0
DO 4 I=1,NC
4 FX(I,J)= 0.0
DO 6 JF=1,NF
READ 2, J
READ 1, HF(J),(FX(I,J),I=1,NC)
DO 5 I=1,NC
5 F(J)= F(J) + FX(I,J)
6 PRINT 103, J,F(J),HF(J),(FX(I,J),I=1,NC)
IF (NDRAW) 9,9,7
7 DO 8 J=1,NDRAW
READ 2, JD
READ 1, SL(JD),SV(JD)
8 PRINT 104, JD,SL(JD),SV(JD)
9 READ 1, SUMERR,HETERR,SUBCOL,DTLIM,(SPECS(I),I=1,4),CHECK
PRINT 105, SUMERR,HETERR,SUBCOL,DTLIM,(SPECS(I),I=1,4)
ITERAT= 0
IF (ABSF(CHECK - 11.111) - 0.01) 10,10,3
10 ITERAT= ITERAT + 1
JC= -1
ITHEAT= - 1
ITEMP= 0
L= NS - 1
N= NS
    
```

```

      IF (JCOTYP - 1) 13,11,11
11  N= L
      DO 12 I=1,NC
12  EQ(I,NS)= 1.0
13  NM1= N - 1
      V(1)= AL(2) - AL(1) - SL(1) - SV(1) + F(1)
      DO 14 J=2,L
14  V(J)= AL(J+1) + V(J-1) - AL(J) - SL(J) - SV(J) + F(J)
      V(NS)= V(L) - AL(NS) - SL(NS) - SV(NS) + F(NS)
15  CALL EQUIL(200)
      DO 17 I=1,NC
      ERR(1)= 1.0/(AL(1) + SL(1) + (V(1)+SV(1))*EQ(I,1))
      X(I,1)= FX(I,1)*ERR(1)
      DO 16 J=2,NS
      ERR(J)= 1.0/(AL(J) + SL(J) + (V(J)+SV(J))*EQ(I,J)
      1 - ERR(J-1)*V(J-1)*EQ(I,J-1)*AL(J))
16  X(I,J)= (FX(I,J) + V(J-1)*EQ(I,J-1)*X(I,J-1))*ERR(J)
      DO 17 JD=1,L
      J= NS - JD
17  X(I,J)= X(I,J) + ERR(J)*X(I,J+1)*AL(J+1)
      DO 18 J=1,NS
      ERR(J)= - 1.0
      DO 18 I=1,NC
18  ERR(J)= ERR(J) + X(I,J)
      DO 19 J=1,NS
      IF (ABSF(ERR(J)) - SUMERR) 19,19,35
19  CONTINUE
      ITHEAT= ITHEAT + 1
      IF (JCOTYP - 1) 21,20,20
20  CALL EQUIL(NS)
      T(NS)= T(NS) - SUBCOL
21  DO 22 J=1,NS
      HL(J)= 0.0
      HV(J)= 0.0
      DO 22 I=1,NC
      HL(J)= HL(J) + X(I,J)*(AHL(I)+BHL(I))*T(J)+CHL(I)*T(J)**2)
22  HV(J)= HV(J) + X(I,J)*EQ(I,J)*(AHV(I)+BHV(I))*T(J)+CHV(I)*T(J)**2)
      IF (JCOTYP - 1) 24,23,23
23  HV(NS)= HL(NS)
24  QNET= HV(L)*V(L) - HL(NS)*AL(NS)
      QC= QNET - SL(NS)*HL(NS) - (V(NS)+SV(NS))*HV(NS) + HF(NS)
      IF (JC) 226,224,225
224  DAMP= DAMP*(SUM/FLOAT(L-1) + 1.0)
      EPSP= 0.0
      JC= 1
      GO TO 228
225  DAMP= DAMP*EPS1/(EPS1 - EPSP)
      IF (ABSF(DAMP - 0.5) - 0.5) 227,227,226
226  DAMP= 1.0
227  JC= 0
      SUM= 0.0
228  EPS1= EPS
      EPS= 0.0
      DO 32 K=2,L
      J= L - K + 2
      QNET= QNET + HV(J)*SV(J) + HL(J)*SL(J) - HF(J)
      QER= QNET + HL(J)*AL(J) - HV(J-1)*V(J-1)
      IF (JC) 27,27,25
25  EPSP= EPSP + QERR(J)*(QER/ABSF(QERR(J)))

```

```
IF (ABSF(QERR(J) - QER) - ABSF(QERR(J) + QER)/3.0) 27,26,26
26 DL= QER*(OLDDL(J)/(QERR(J) - QER))
GO TO 31
27 DL= QER/(HV(J-1) - HL(J))*DAMP
CHECK= AL(J)
IF (V(J-1) - AL(J)) 28,29,29
28 CHECK= V(J-1)
29 IF (ABSF(DL) - 0.25*CHECK) 31,31,30
30 SUM= SUM + 0.25*CHECK/ABSF(DL) - 1.0
DL=0.25*CHECK*DL/ABSF(DL)
31 OLDDL(J)= DL
QERR(J)= QER
AL(J)= AL(J) + DL
32 EPS= EPS + ABSF(QER)
33 QR= QNET + (AL(1)+SL(1))*HL(1) + SV(1)*HV(1) - HF(1)
DO 34 J=2,L
IF (ABSF(QERR(J)) - 0.5*HETERR*ABSF(QC + QR)) 34,34,13
34 CONTINUE
CALL OUTPUT(1)
OK= 0.0
CALL SPEC
IF (OK - 1.0) 3,3,10
35 IF (ITEMP - LIM) 37,36,36
36 CALL OUTPUT(2)
GO TO 3
37 DO 38 J=1,N
A(NS,J)= 0.0
HV(J)= - (ERR(J) + 1.0)*LOGF(ERR(J) + 1.0)
DO 38 K=1,N
38 A(J,K)= 0.0
DO 44 I=1,NC
ERR(I)= 1.0/(AL(1) + SL(1) + (V(1)+SV(1))*EQ(I,1))
DO 39 J=2,NS
39 ERR(J)= 1.0/(AL(J) + SL(J) + (V(J)+SV(J))*EQ(I,J)
1 - ERR(J-1)*V(J-1)*EQ(I,J-1)*AL(J))
DO 44 K=1,N
DO 40 J=1,NS
40 HL(J)= 0.0
CHECK= -X(I,K)*EQT(AK(I),BK(I),CK(I),DK(I),T(K),EQ(I,K),KTYP)
HL(K)= CHECK*(V(K) + SV(K))*ERR(K)
IF (K - NS) 41,43,43
41 HL(K+1)= - CHECK*V(K)
DO 42 J=K,L
42 HL(J+1)= (HL(J+1) + V(J)*EQ(I,J)*HL(J))*ERR(J+1)
43 A(NS,K)= A(NS,K) + HL(NS)
DO 44 JF=2,NS
J= NS - JF + 1
HL(J)= HL(J) + ERR(J)*HL(J+1)*AL(J+1)
44 A(J,K)= A(J,K) + HL(J)
DO 46 I=1,NM1
IP1= I + 1
A(I,I)= 1.0/A(I,I)
HV(I)= HV(I)*A(I,I)
DO 45 K=IP1,N
45 A(I,K)= A(I,K)*A(I,I)
DO 46 J=IP1,N
HV(J)= HV(J) - HV(I)*A(J,I)
DO 46 K=IP1,N
46 A(J,K)= A(J,K) - A(I,K)*A(J,I)
```

```

HV(N)= HV(N)/A(N,N)
DO 47 I=1,NM1
  J= NM1 + 1 - I
  JP1= J + 1
  DO 47 K=1,J
47 HV(K)= HV(K) - A(K,JP1)*HV(JP1)
  DO 49 J=1,N
  IF (ABSF(HV(J)) - DTLIM) 49,49,48
48 HV(J)= DTLIM*HV(J)/ABSF(HV(J))
49 T(J)= T(J) + HV(J)
  ITTEMP= ITTEMP + 1
  GO TO 15
END

```

```

C SUBROUTINE EQUIL(N)
  SUBROUTINE FOR EQUILIBRIUM CONSTANTS AND BUBBLE POINT
  DIMENSION AK(20),BK(20),CK(20),DK(20),EQ(20, 88),T( 88),X(20, 88)
  COMMON AK,BK,CK,DK,KTYP,EQ,T,X,NC,L,A,B,C,D,SUMX,SUMD
  IF (N - 150) 1,1,7
  1 IF (KTYP - 1) 2,2,4
  2 SUMX= 0.0
  SUMD= 0.0
  DO 3 I=1,NC
  A= EXPF(AK(I)/(T(N)+DK(I)) + BK(I) + CK(I)*(T(N)+DK(I)))
  SUMX= SUMX + X(I,N)*(A - 1.0)
  3 SUMD= SUMD + X(I,N)*A*(CK(I) - AK(I)/((T(N)+DK(I))**2))
  T(N)= T(N) - SUMX/SUMD
  IF (ABSF(SUMX) - 0.00001) 11,11,2
  4 A= 0.0
  B= 0.0
  C= 0.0
  D= 0.0
  DO 5 I=1,NC
  A= A + (AK(I) - 1.0)*X(I,N)
  B= B + BK(I)*X(I,N)
  C= C + CK(I)*X(I,N)
  5 D= D + DK(I)*X(I,N)
  6 SUMX= A + B*T(N) + C*T(N)**2 + D*T(N)**3
  SUMD= B + 2.0*C*T(N) + 3.0*D*T(N)**2
  T(N)= T(N) - SUMX/SUMD
  IF (ABSF(SUMX) - 0.00001) 11,11,6
  7 DO 10 J=1,L
  DO 10 I=1,NC
  IF (KTYP - 1) 8,8,9
  8 EQ(I,J)= EXPF(AK(I)/(T(J)+DK(I)) + BK(I) + CK(I)*(T(J)+DK(I)))
  GO TO 10
  9 EQ(I,J)= AK(I) + BK(I)*T(J) + CK(I)*T(J)**2 + DK(I)*T(J)**3
  10 CONTINUE
  11 RETURN
  END

```

```

C FUNCTION EQT(A,B,C,D,T,EQ,K)
  FUNCTION FOR DERIVATIVES OF EQUILIBRIUM CONSTANTS
  IF (K - 1) 1,1,2
  1 EQT= EQ*(C - A/(T+D)**2)
  GO TO 3
  2 EQT= B + 2.0*C*T + 3.0*D*T**2
  3 RETURN
  END

```



```

SUBROUTINE OUTPUT(JOUT)
  DIMENSION AK(81),EQ(20, 88),T( 88),AL( 88),X(20, 88),STAND(4),
  ITERAT(4),SUMX( 88),V( 88),SL( 88),SV( 88)
  COMMON AK,EQ,T,X,NC,N,QC,QR,STAND,ITERAT,SUMX,AL,V,SL,SV,NS
110 FORMAT (11HITERATIONS,4I8)
111 FORMAT (118HO J      T(J)          AL(J)          SUMX - 1.0
  1X(1,J)      X(2,J)      X(3,J)      X(4,J)      X(5,J) /
  2(I3,5E15.6,3E14.6))
112 FORMAT (119HO J      X(6,J)      X(7,J)      X(8,J)
  1X(9,J)      X(10,J)      X(11,J)      X(12,J)      X(13,J) /
  2 (I3,5E15.6,3E14.6))
113 FORMAT (105HO J      X(14,J)      X(15,J)      X(16,J)
  1X(17,J)      X(18,J)      X(19,J)      X(20,J) / (I3,5E15.6,
  22E14.6))
114 FORMAT (34HOTOP PRODUCT AMOUNTS BY COMPONENTS/E63.6,E15.6,3E14.6/
  1(E18.6,4E15.6,3E14.6))
115 FORMAT (41HOHEAT BALANCE SATISFIED, CONDENSER LOAD =,E14.6,19H ,
  1REBOILER LOAD =,E14.6)
134 FORMAT (15HOBOTTOM PRODUCT/E63.6,E15.6,3E14.6/(E18.6,4E15.6,3E14.6
  1))
135 FORMAT (20HOVAPOR DRAW ON STAGE,I4/E63.6,E15.6,3E14.6/(E18.6,4E
  115.6,3E14.6))
136 FORMAT (21HOLIQUID DRAW ON STAGE,I4/E63.6,E15.6,3E14.6/(E18.6,4E
  115.6,3E14.6))
117 PRINT 110, (ITERAT(I),I=1,3)
  IF (NC-20) 118,120,120
118 DO 119 I=NC,19
  DO 119 J=1,NS
119 X(I+1,J)= 0.0
120 PRINT 111, (J,T(J),AL(J),SUMX(J),(X(I,J),I=1,5),J=1,NS)
  IF (NC-5) 123,123,121
121 PRINT 112, (J,(X(I,J),I=6,13),J=1,NS)
  IF (NC-13) 123,123,122
122 PRINT 113, (J,(X(I,J),I=14,20),J=1,NS)
123 DO 124 I=1,NC
124 SUMX(I)= X(I,1)*AL(1)
  PRINT 134, (SUMX(I),I=1,NC)
  DO 125 I=1,NC
125 SUMX(I)= X(I,NS)*EQ(I,NS)*V(NS)
  PRINT 114, (SUMX(I),I=1,NC)
  DO 131 J=1,NS
  IF (SL(J)) 128,128,126
126 DO 127 I=1,NC
127 SUMX(I)= X(I,J)*SL(J)
  PRINT 136, J,(SUMX(I),I=1,NC)
128 IF (SV(J)) 131,131,129
129 DO 130 I=1,NC
130 SUMX(I)= X(I,J)*EQ(I,J)*SV(J)
  PRINT 135, J,(SUMX(I),I=1,NC)
131 CONTINUE
  GO TO (132,133),JOUT
132 PRINT 115, QC,QR
133 RETURN
END
SUBROUTINE SPEC
C DUMMY SPECIFICATION SUBROUTINE
  OK= 0.0
  RETURN
END

```

Appendix D. Distillation calculations by linearization

The program in this section solves the same problem as the program in appendix C, but by the method outlined in the body of this report instead of the Thiele-Geddes method. The input data are the same as for the last program, but some quantities have a different significance, as follows:

LIM is a limit on the total number of iterations.

IN(5) controls the number of times that temperature corrections are made without changes in the liquid and vapor flow rates (see the statement just before statement 39). A value of 0 or 1 for IN(5) should be satisfactory.

SUMERR is used to check convergence. The mole fraction of each component in the reboiler must change by less than SUMERR between one iteration and the next in order to satisfy the convergence criterion.

HETERR is not used.

The output is also similar to that of the preceding program.

The dimensions in the program correspond to 10 components and 40 stages (including the reboiler and condenser). For problems outside these limits, the dimensions can be changed appropriately. The BAND(J) and MATINV subroutines of appendix A have not been repeated here. The number of unknowns on each stage is taken to be $n = NC + 3$. The three additional unknowns are proposed changes in the temperature, liquid flow rate, and vapor flow rate.

The flow rate of the bottom product is controlled (that is, left unchanged) by statement 41, and that of the reflux by statement 52.

These represent the two remaining degrees of freedom after the number

of stages, nature of feed, etc., have been specified. At the top or bottom of the column one might wish to control one of the following:

1. Bottom product amount or top product amount.
2. Reflux or vapor flow rate from the reboiler.
3. The mole fraction of a component in the top or bottom product.
4. The flow rate of a component in the top or bottom product.
5. The reboiler or condenser temperature.
6. The heat load for the condenser or the reboiler.

On page 36 we have given FORTRAN statements for implementing the first five of these possibilities at both the top and the bottom of the column. Statement 41 and the following statement should be replaced by statements 41 through 48, and for the top, statement 52 and the following statement should be replaced by statements 52 through 62. These added statements use IN(1) through IN(4) and SPECS(1) and SPECS(2) to decide which specification to use, which component to control, and what value to achieve.

These additional, more flexible specifications must be used with caution. First, they can contradict each other. One can't specify both the top and bottom product amounts independently. Second, one must stay within the range of possible operating conditions of the column. For example, the reboiler temperature can be no higher than the boiling point of the heaviest component. Other limitations may be found which are more subtle than these examples.

```

PROGRAM DIST(INPUT,OUTPUT)
C PROGRAM FOR FRACTIONATING COLUMN WITH SIDE-STREAM DRAWS
DIMENSION A(13,13),B(13,13),C(13,40),D(13,27),G(13),X(13,13),Y(13,
113),AK(10),BK(10),CK(10),DK(10),AHL(10),BHL(10),CHL(10),AHV(10),BH
2V(10),CHV(10),F(40),HF(40),FX(10,40),IN(5),SPECS(4),SL(40),SV(40)
3,ERR(40),SAVE(10),T(40),AL(40),V(40)
COMMON A,B,C,D,G,X,Y,NP3,NS,AK,BK,CK,DK,KTYP,NC,JCOTYP,ITERAT,NP1,
1NP2,QC,QR,SL,SV,ERR,T,AL,V,F,HF,FX,IN,SPECS
101 FORMAT (74H1COMPONENTS STAGES FEEDS COTYP KTYP LIMIT DR
1AWS INSTRUCTIONS,26X,7HPROBLEM/(13I8))
102 FORMAT (118H0 I AK(I) BK(I) CK(I) DK(I)
1 AHL(I) BHL(I) CHL(I) AHV(I) BHV(I) CHV(I)/
2(I3,7E12.4,3E11.4))
103 FORMAT (60H0 STAGE FEED ENTHALPY COMPO
INENTS/I8,E22.7,6E15.7/(8E15.7))
104 FORMAT (40H0 STAGE LIQUID DRAW VAPOR DRAW/(I8,2E17.6))
105 FORMAT (69H0 SUMERR HETERR SUBCOL DTLIM
1 SPECS/(8E15.7))
1 FORMAT (9E8.4)
2 FORMAT (13I4)
3 READ 2, NC,NS,NF,JCOTYP,KTYP,LIM,NDRAW,(IN(I),I=1,5),NPROB
IF (NC) 98,98,99
98 STOP
99 READ 1, (AK(I),BK(I),CK(I),DK(I),AHL(I),BHL(I),CHL(I),AHV(I),
1BHV(I),CHV(I),I=1,NC)
NP1= NC + 1
NP2= NC + 2
NP3= NC + 3
READ 1, (AL(J),J=1,NS)
READ 1, (T(J),J=1,NS)
PRINT 101, NC,NS,NF,JCOTYP,KTYP,LIM,NDRAW,(IN(I),I=1,5),NPROB
PRINT 102, (I,AK(I),BK(I),CK(I),DK(I),AHL(I),BHL(I),CHL(I),AHV(I),
1BHV(I),CHV(I),I=1,NC)
DO 4 J=1,NS
SL(J)= 0.0
SV(J)= 0.0
F(J)= 0.0
HF(J)= 0.0
DO 4 I=1,NC
4 FX(I,J)= 0.0
DO 6 JF=1,NF
READ 2, J
READ 1, HF(J),(FX(I,J),I=1,NC)
DO 5 I=1,NC
5 F(J)= F(J) + FX(I,J)
6 PRINT 103, J,F(J),HF(J),(FX(I,J),I=1,NC)
IF (NDRAW) 9,9,7
7 DO 8 J=1,NDRAW
READ 2, JD
READ 1, SL(JD),SV(JD)
8 PRINT 104, JD,SL(JD),SV(JD)
9 READ 1, SUMERR,HETERR,SUBCOL,DTLIM,(SPECS(I),I=1,4),CHECK
PRINT 105, SUMERR,HETERR,SUBCOL,DTLIM,(SPECS(I),I=1,4)
L= NS - 1
DO 11 I=1,NC
11 SAVE(I)= 0.0
ITERAT= -1
IF (ABS(F(CHECK - 11.111) - 0.01) 12,12,3
12 ITERAT= ITERAT + 1

```

```

V(1)= AL(2) - AL(1) - SL(1) - SV(1) + F(1)
DO 10 J=2,L
10 V(J)= AL(J+1) + V(J-1) - AL(J) - SL(J) - SV(J) + F(J)
V(NS)= V(L) - AL(NS) - SL(NS) - SV(NS) + F(NS)
DO 17 I=1,NC
EQ= EQUIL(I,T(1))
ERR(1)= 1.0/(AL(1)+SL(1) + (V(1)+SV(1))*EQ)
C(I,1)= FX(I,1)*ERR(1)
DO 16 J=2,NS
EQB= EQ
EQ= EQUIL(I,T(J))
IF (J=NS) 15,13,13
13 IF (JCOTYP) 15,15,14
14 EQ= 1.0
15 ERR(J)=1.0/(AL(J)+SL(J)+(V(J)+SV(J))*EQ-ERR(J-1)*V(J-1)*EQB*AL(J))
16 C(I,J)= (FX(I,J) + V(J-1)*EQB*C(I,J-1))*ERR(J)
DO 17 JD=1,L
J= NS - JD
17 C(I,J)= C(I,J) + ERR(J)*C(I,J+1)*AL(J+1)
DO 19 J=1,NS
SUMX= 0.0
DO 18 I=1,NC
18 SUMX= SUMX + C(I,J)
DO 19 I=1,NC
19 C(I,J)= C(I,J)/SUMX
IF (ITERAT-LIM) 20,20,22
20 DO 21 I=1,NC
IF (ABS(C(I,1)-SAVE(I))-SUMERR) 21,21,27
21 CONTINUE
22 HL= 0.0
HV= 0.0
HLU= 0.0
TU= T(2)
DO 23 I=1,NC
HL= HL+C(I,1)*(AHL(I)+BHL(I)*T(1)+CHL(I)*T(1)**2)
HV= HV+C(I,1)*EQUIL(I,T(1))*(AHV(I)+BHV(I)*T(1)+CHV(I)*T(1)**2)
23 HLU= HLU + C(I,2) *(AHL(I)+BHL(I)*TU+CHL(I)*TU**2)
QR= (V(1)+SV(1))*HV + (AL(1)+SL(1))*HL - HLU*AL(2) - HF(1)
HL= 0.0
HV= 0.0
HVB= 0.0
TB= T(L)
DO 24 I=1,NC
HL= HL+C(I,NS)*(AHL(I)+BHL(I)*T(NS)+CHL(I)*T(NS)**2)
HV=HV+C(I,NS)*EQUIL(I,T(NS))*(AHV(I)+BHV(I)*T(NS)+CHV(I)*T(NS)**2)
24 HVB= HVB + C(I,L)*EQUIL(I,TB)*(AHV(I)+BHV(I)*TB+CHV(I)*TB**2)
IF (JCOTYP) 26,26,25
25 HV= HL
26 QC= HVB*V(L)-HL*(AL(NS)+SL(NS))-HV*(V(NS)+SV(NS)) + HF(NS)
CALL OUTPUT
GO TO 3
27 DO 28 I=1,NC
28 SAVE(I)= C(I,1)
J= 0
DO 29 I=1,NP3
DO 29 K=1,NP3
Y(I,K)= 0.0
29 X(I,K)= 0.0
30 J= J + 1

```

```

DO 31 I=1, NP3
G(I)= 0.0
DO 31 K=1, NP3
A(I,K)= 0.0
B(I,K)= 0.0
31 D(I,K)= 0.0
G(NP1)= 1.0
DO 38 I=1, NC
B(NP1,I)= 1.0
EQ= EQUIL(I,T(J))
B(I,NP1)= EQDT(I,T(J))*C(I,J)*(V(J)+SV(J))
IF (J-NS) 32,33,33
32 D(I,I)= - AL(J+1)
D(I,NP2)= - C(I,J+1)
GO TO 35
33 IF (JCOTYP) 35,35,34
34 EQ= 1.0
B(I,NP1)= 0.0
B(NP1,I)= EQUIL(I,T(J)+SUBCOL)
B(NP1,NP1)= B(NP1,NP1) + C(I,J)*EQDT(I,T(J)+SUBCOL)
35 IF (J-1) 37,37,36
36 TB= T(J-1)
EQB= EQUIL(I,TB)
A(I,I)= - EQB*V(J-1)
A(I,NP1)= - EQDT(I,TB)*C(I,J-1)*V(J-1)
A(I,NP3)= - EQB*C(I,J-1)
37 B(I,I)= AL(J) + SL(J) + EQ*(V(J)+SV(J))
B(I,NP2)= C(I,J)
B(I,NP3)= EQ*C(I,J)
38 G(I)= FX(I,J)
IF (ITERAT-IN(5)) 39,39,40
39 NP3= NP3 - 2
CALL BAND(J)
NP3= NP3 + 2
IF (J-NS) 30,65,65
40 B(NP3,NP2)= 1.0
B(NP3,NP3)= 1.0
G(NP3)= F(J) - SL(J) - SV(J) - AL(J) - V(J)
D(NP3,NP2)= - 1.0
A(NP3,NP3)= - 1.0
IF (J-1) 41,41,49
41 B(NP2,NP2)= 1.0
G(NP3)= G(NP3) + AL(J+1)
CALL BAND(J)
GO TO 30
49 IF (J-NS) 50,52,52
50 TU= T(J+1)
G(NP3)= G(NP3) + AL(J+1) + V(J-1)
DO 51 I=1, NC
EQ= EQUIL(I,T(J))
EQB= EQUIL(I,TB)
HVIB= AHV(I) + BHV(I)*TB + CHV(I)*TB**2
HVI= AHV(I) + BHV(I)*T(J) + CHV(I)*T(J)**2
HLIU= AHL(I) + BHL(I)*TU + CHL(I)*TU**2
HLI= AHL(I) + BHL(I)*T(J) + CHL(I)*T(J)**2
B(NP2,NP2)= B(NP2,NP2) + HLI*C(I,J)
B(NP2,NP3)= B(NP2,NP3) + HVI*EQ*C(I,J)
D(NP2,NP2)= D(NP2,NP2) - HLIU*C(I,J+1)
A(NP2,NP3)= A(NP2,NP3) - HVIB*EQB*C(I,J-1)

```

```
A(NP2,I)= - HVIB*EQB*V(J-1)
B(NP2,I)= HLI*(AL(J)+SL(J)) + HVI*(V(J)+SV(J))*EQ
D(NP2,I)= - HLIU*AL(J+1)
A(NP2,NP1)= A(NP2,NP1)-V(J-1)*C(I,J-1)*(HVIB*EQDT(I,TB)+EQB*(BHV(I
1)+2.0*CHV(I)*TB))
B(NP2,NP1)= B(NP2,NP1)+C(I,J)*((AL(J)+SL(J))*(BHL(I)+2.0*CHL(I)*T(
1J)))+(V(J)+SV(J))*(HVI*EQDT(I,T(J))+EQ*(BHV(I)+2.0*CHV(I)*T(J)))
51 D(NP2,NP1)= D(NP2,NP1) - AL(J+1)*C(I,J+1)*(BHL(I)+2.0*CHL(I)*TU)
G(NP2)= HF(J)
CALL BAND(J)
GO TO 30
52 B(NP2,NP2)= 1.0
G(NP3)= G(NP3) + V(J-1)
CALL BAND(J)
DO 64 J=1,NS
IF (ABS(F(C(NP2,J))-0.4*AL(J)) 64,64,63
63 C(NP2,J)= AL(J)*C(NP2,J)/ABS(F(C(NP2,J)))*0.4
64 AL(J)= AL(J) + C(NP2,J)
65 DO 67 J=1,NS
IF (ABS(F(C(NP1,J))-DTLIM) 67,67,66
66 C(NP1,J)= DTLIM*C(NP1,J)/ABS(F(C(NP1,J))
67 T(J)= T(J) + C(NP1,J)
GO TO 12
END
```

XBL 679-4907

```

SUBROUTINE OUTPUT
DIMENSION A(13,13),B(13,13),C(13,40),D(13,27),G(13),X(13,13),Y(13,
113),AK(10),BK(10),CK(10),DK(10),SL(40),SV(40),SUMX(40),T(40),AL(4
20),V(40)
COMMON A,B,C,D,G,X,Y,NP3,NS,AK,BK,CK,DK,KTYP,NC,JCOTYP,ITERAT,NP1,
1NP2,QC,QR,SL,SV,SUMX,T,AL,V
110 FORMAT (11H1,I4,11H ITERATIONS)
111 FORMAT (I3,5E15.6,3E14.6)
112 FORMAT (118HO J      T(J)          L(J)          V(J)
1X(1,J)      X(2,J)      X(3,J)      X(4,J)      X(5,J) )
113 FORMAT (119HO J      X(6,J)      X(7,J)      X(8,J)
1X(9,J)      X(10,J)     X(11,J)     X(12,J)     X(13,J) )
114 FORMAT (105HO J      X(14,J)     X(15,J)     X(16,J)
1X(17,J)     X(18,J)     X(19,J)     X(20,J) )
115 FORMAT (17HOCONDENSER LOAD =,E14.6,19H, REBOILER LOAD =,E14.6)
116 FORMAT (34HOTOP PRODUCT AMOUNTS BY COMPONENTS/E63.6,E15.6,3E14.6/
1(E18.6,4E15.6,3E14.6))
117 FORMAT (15HOBOTTOM PRODUCT/E63.6,E15.6,3E14.6/(E18.6,4E15.6,3E14.6
1))
118 FORMAT (20HOVAPOR DRAW ON STAGE,I4/E63.6,E15.6,3E14.6/(E18.6,4E
115.6,3E14.6))
119 FORMAT (21HOLIQUID DRAW ON STAGE,I4/E63.6,E15.6,3E14.6/(E18.6,4E
115.6,3E14.6))
PRINT 110, ITERAT
PRINT 112
IF (NC=5) 120,120,122
120 DO 121 J=1,NS
121 PRINT 111, J,T(J),AL(J),V(J),(C(I,J),I=1,NC)
GO TO 127
122 PRINT 111, (J,T(J),AL(J),V(J),(C(I,J),I=1,5),J=1,NS)
PRINT 113
IF (NC=13) 123,123,125
123 DO 124 J=1,NS
124 PRINT 111, J,(C(I,J),I=6,NC)
GO TO 127
125 PRINT 111,(J,(C(I,J),I=6,13),J=1,NS)
PRINT 114
DO 126 J=1,NS
126 PRINT 111, J,(C(I,J),I=14,NC)
127 DO 128 I=1,NC
128 SUMX(I)= C(I,1)*AL(1)
PRINT 117, (SUMX(I),I=1,NC)
DO 130 I=1,NC
EQ= EQUIL(I,T(NS))
IF (JCOTYP) 130,130,129
129 EQ= 1.0
130 SUMX(I)= C(I,NS)*EQ*V(NS)
PRINT 116, (SUMX(I),I=1,NC)
DO 137 J=1,NS
IF (SL(J)) 133,133,131
131 DO 132 I=1,NC
132 SUMX(I)= C(I,J)*SL(J)
PRINT 119, J,(SUMX(I),I=1,NC)
133 IF (SV(J)) 137,137,134
134 DO 136 I=1,NC
EQ= EQUIL(I,T(J))
IF (J=NS) 136,138,138
138 IF (JCOTYP) 136,136,135
135 EQ= 1.0

```



```
136 SUMX(I)= C(I,J)*EQ*SV(J)
    PRINT 118, J,(SUMX(I),I=1,NC)
137 CONTINUE
    PRINT 115, QC,QR
    RETURN
    END

    FUNCTION EQUIL(I,T)
    DIMENSION A(13,13),B(13,13),C(13,40),D(13,27),G(13),X(13,13),Y(13,
113),AK(10),BK(10),CK(10),DK(10)
    COMMON A,B,C,D,G,X,Y,NP3,NS,AK,BK,CK,DK,KTYP
    IF (KTYP-1) 1,1,2
1 EQUIL= EXPF(AK(I)/(T+DK(I)) + BK(I)+CK(I)*(T+DK(I)))
    RETURN
2 EQUIL= AK(I) + BK(I)*T + CK(I)*T**2 + DK(I)*T**3
    RETURN
    END

    FUNCTION EQDT(I,T)
    DIMENSION A(13,13),B(13,13),C(13,40),D(13,27),G(13),X(13,13),Y(13,
113),AK(10),BK(10),CK(10),DK(10)
    COMMON A,B,C,D,G,X,Y,NP3,NS,AK,BK,CK,DK,KTYP
    IF (KTYP-1) 1,1,2
1 EQDT= EXPF(AK(I)/(T+DK(I))+BK(I)+CK(I)*(T+DK(I)))*(CK(I)-AK(I)/(
1T+DK(I))**2)
    RETURN
2 EQDT= BK(I) + 2.0*CK(I)*T + 3.0*DK(I)*T**2
    RETURN
    END
```

```

41 IF (IN(1)) 47,47,42
42 K= IN(1)
   GO TO (43,44,45,46),K
C   FIXED BOTTOM PRODUCT COMPOSITION, SPECS(1)= X(I)
43 I= IN(3)
   B(NP2,I)= 1.0
   G(NP2)= SPECS(1)
   GO TO 48
C   FIXED BOTTOM PRODUCT COMPONENT AMOUNT, SPECS(1)= X(I)*AL(J)
44 I= IN(3)
   B(NP2,I)= AL(I)
   B(NP2,NP2)= C(I,1)
   G(NP2)= SPECS(1)
   GO TO 48
C   FIXED REBOILER TEMPERATURE, SPECS(1)= T(J)
45 B(NP2,NP1)= 1.0
   G(NP2)= SPECS(1) - T(1)
   GO TO 48
C   FIXED VAPOR FLOW FROM REBOILER, SPECS(1)= V(1)
46 B(NP2,NP3)= 1.0
   G(NP2)= SPECS(1) - V(1)
   GO TO 48
C   FIXED BOTTOM PRODUCT AMOUNT
47 B(NP2,NP2)= 1.0
48 G(NP3)= G(NP3) + AL(J+1)

52 IF (IN(2)) 61,61,53
53 K= IN(2)
   GO TO (54,56,59,60),K
C   FIXED TOP PRODUCT COMPOSITION, SPECS(2)= Y(I)
54 I= IN(4)
   B(NP2,I)= EQUIL(I,T(NS))
   B(NP2,NP1)= C(I,NS)*EQDT(I,T(NS))
   G(NP2)= SPECS(2)
   IF (JCOTYP) 62,62,55
55 B(NP2,I)= 1.0
   B(NP2,NP1)= 0.0
   GO TO 62
C   FIXED TOP PRODUCT COMPONENT AMOUNT, SPECS(2)= Y(I)*V(J)
56 I= IN(4)
   EQ= EQUIL(I,T(NS))
   B(NP2,NP1)= C(I,NS)*V(NS)*EQDT(I,T(NS))
   IF (JCOTYP) 58,58,57
57 EQ= 1.0
   B(NP2,NP1)= 0.0
58 B(NP2,NP3)= C(I,NS)*EQ
   B(NP2,I)= EQ*V(NS)
   G(NP2)= SPECS(2)
   GO TO 62
C   FIXED CONDENSER TEMPERATURE, SPECS(2)= T(J)
59 B(NP2,NP1)= 1.0
   G(NP2)= SPECS(2) - T(NS)
   GO TO 62
C   FIXED TOP PRODUCT AMOUNT
60 B(NP2,NP3)= 1.0
   GO TO 62
C   FIXED REFLUX
61 B(NP2,NP2)= 1.0
62 G(NP3)= G(NP3) + V(J-1)

```

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

[The page contains extremely faint, illegible text, likely bleed-through from the reverse side of the document. The text is arranged in approximately 25 horizontal lines across the page.]

