C&EE50

UNCLASSIFIED

AD 664 045

CHEMIST--THE RAND CHEMICAL EQUILIBRIUM PROGRAM

E. C. DeLand

The Rand Corporation
Santa Monica, California

December 1967

Norm Shapiro
Program Library Dept.

Processed for . . .

DEFENSE DOCUMENTATION CENTER
DEFENSE SUPPLY AGENCY

CLEARINGHOUSE
FOR FEDERAL SCIENTIFIC AND TECHNICAL INFORMATION

U. S. DEPARTMENT OF COMMERCE / NATIONAL BUREAU OF STANDARDS / INSTITUTE FOR APPLIED TECHNOLOGY

UNCLASSIFIED

MEMORANDUM
RM-5404-PR
DECEMBER 1967

AD664045

CHEMIST—THE RAND CHEMICAL EQUILIBRIUM PROGRAM

E. C. DeLand

DDC
RECEIVED
JAN 18 1968
C

PREPARED FOR:
UNITED STATES AIR FORCE PROJECT RAND

The RAND Corporation
SANTA MONICA • CALIFORNIA

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va. 22151

14

MEMORANDUM
RM-5404-PR
DECEMBER 1967

CHEMIST—THE RAND CHEMICAL EQUILIBRIUM PROGRAM

E. C. DeLand

This research is supported by the United States Air Force under Project RAND — Contract No. F44620-67-C-0045 — monitored by the Directorate of Operational Requirements and Development Plans, Deputy Chief of Staff, Research and Development, Hq USAF. RAND Memoranda are subject to critical review procedures at the research department and corporate levels. Views and conclusions expressed herein are nevertheless the primary responsibility of the author, and should not be interpreted as representing the official opinion or policy of the United States Air Force or of The RAND Corporation.

DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

The RAND Corporation
1700 MAIN ST. • SANTA MONICA • CALIFORNIA • 90406

PREFACE

This Memorandum reports in detail on the structure and use of CHEMIST, the RAND chemical equilibrium program, a computer program used to simulate complex chemical equilibria. This report answers the growing demand for a reference manual to accompany and document the program. It should be of interest both to those having similar computer programs and to those wishing to use CHEMIST for their problems.

The manual will be updated periodically as the CHEMIST program evolves. The present References and Selected Bibliography comprise as complete a listing of the literature as is possible at this writing. It would be appreciated if users acquainted with additional material would submit bibliographic information for incorporation into later editions.

PRECEDING
PAGE BLANK

SUMMARY

This Memorandum is essentially a manual for the use of CHEMIST, a computer program used to simulate complex chemical equilibria. CHEMIST has been used to make mathematical models of simple chemical systems (e.g., bicarbonate system in water solution), organic systems (e.g., the ionization of serum albumin), viable biological systems (e.g., blood chemistry, electrolyte and fluid distribution in the "whole body," function of the kidney), and inorganic systems at non-standard pressures and temperatures (e.g., planet atmospheres, rocket exhausts, graphite-carbon vapor system). Obviously, CHEMIST can meet the varied requirements of many different users, except that it is applicable only to the computation of chemical equilibria or "steady-states" and not directly to the study of chemical kinetics.

The program--written in natural language to facilitate use by professionals not trained in computer programming--has evolved over the years, and will continue to do so. Consequently, this manual will be modified periodically to keep abreast of changes. This edition of the program and manual will be designated as CHEMIST; subsequent editions will be designated by an appropriate Roman numeral (i.e., CHEMIST II).

Chapter I is a general introduction to the theory and the literature. Chapter II details the operational control of the program, and Chap. III shows elaborated examples of its use. Finally, Chap. IV documents the subroutines.

PRECEDING
PAGE BLANK

ACKNOWLEDGMENTS

The CHEMIST program has been at The RAND Corporation in various guises for many years, and it is impossible to credit adequately the many contributors to this manual. However, the origins of the program may be traced directly to the White, Johnson, and Dantzig paper of 1958.† It was originally programmed for the IBM 704 computer by Herschell E. Kanter and for an analog computer by E. C. DeLand. The first publication was "A Mathematical Model of the Human External Respiratory System," by G. B. Dantzig, et al.‡

While the present author happens to have put this manual together, there are many contributors to the modern form of CHEMIST. Mathematician N. Z. Shapiro has derived much of the theoretical foundations for the method of studying complex chemical systems; mathematician R. J. Clasen developed the essential characteristics of the present (digital) computer program. Physical chemist J. C. De Haven and the present author have primarily been concerned with applications. Programmer analysts Leola Cutler, Marion Shapley, and Rose Heirschfeldt have improved the code and written additional subroutines. An interesting note of appreciation is due E.A.H. Magnier, M.D., who wrote the subroutine GOALN8.

The author wishes to thank these people especially, but others as well, for their unstinting constructive criticism and unrelenting motivation.

---

†W. B. White, S. M. Johnson, and G. B. Dantzig, "Chemical Equilibrium in Complex Mixtures," J. Chem. Physics, Vol. 28, No. 5, May 1958, pp. 751-755.

‡Perspectives in Biology and Medicine, Vol. IV, No. 3, Spring 1961, pp. 324-376.

PRECEDING
PAGE BLANK

CONTENTS

PRECEDING PAGE BLANK

TABLES

Chapter I

INTRODUCTION

GENERAL REMARKS

Laboratory determination of constituent quantities of a chemical milieu at equilibrium is, for all but the simplest systems, an arduous and exacting process. Indeed, in inorganic chemistry the procedure may take days of highly skilled attention, while in organic chemistry a complete determination is frequently impossible. Yet, in principle, given the required data and conditions of the experiment, one should be able to calculate the concentrations of the equilibrium constituents by well-established procedures; and, using a computer, with relative ease.

There are three well-known methods for calculating equilibrium constituents: kinetic equations; mass-action equations; and by an indirect method of optimization of certain thermodynamic properties. Generally, the data required for computation in a thermodynamically closed system in a single phase are temperature, pressure, moles of each reactant or "component," the list of expected chemical reactions, and the list of either equilibrium constants or forward and backward reaction rate constants for each chemical reaction.

By the usual kinetic method, each chemical reaction is transformed into an equivalent set of ordinary differential equations for which the reaction rate constants are required. Using mass-action equations (the usual method of calculation in chemical laboratories) each chemical equation is usually transformed into a non-linear algebraic equation for which the equilibrium constants are required. In either

case, the sets of equations are solved simultaneously while maintaining stoichiometric conservation of mass. For kinetic equations, the time trajectory of the system to equilibrium is computed; for mass-action equations, only the final equilibrium state. The additional information by the kinetic method is obtained at the expense of requiring twice as many reaction constants in the data.

The third method--the method used by CHEMIST--again computes only the final steady equilibrium state using the equilibrium constants, but is more compatible to computer solution. Essentially, this method is based on a theorem of Sir Willard Gibbs [1] to the effect that the equilibrium composition will be such that the total thermodynamic free energy of the system is minimized (or the entropy maximized) under the conditions of the experiment. CHEMIST, using an iterative procedure from mathematical programming, determines that composition which minimizes the system's total free energy, subject to the constraints on the system. The data required are equivalent to those for the mass-action equation method.

Not all chemical systems having real, unique solutions may be solved with this program. First, if the concentration or the absolute amount of a species becomes very small during the iterative procedure, the numerical precision limitations and the round-off error of the computer preclude an accurate solution and the program executes an exit. Second, space limitations in computer memory dictate a size limitation for the system. Third, since CHEMIST is designed to simulate equilibrium systems, time-dependent and steady-state systems may be modeled only insofar as they can be approximated by equilibrium systems (e.g., see J. C. De Haven and N. Z. Shapiro [2]).

The References attached below (pp. 129-130) list: several applications of the program (to open and closed systems); several papers relating to definitions of aspects of chemical systems; analyses of membrane equilibria; discussions of the existence of solutions; and details of methods of solution. The last named topic--the method of solution using CHEMIST--is of principal concern here, and is treated in papers by R. J. Clasen, in particular Ref. 3. (This paper, not reviewed here, should help to clarify certain aspects of the present Memorandum).

This Memorandum will not show the justification of the theoretical methods used by CHEMIST. Instead, it emphasizes the useful, practical aspects. For the interested reader, the mathematical theory is discussed particularly in Gibbs [1]; White, Johnson, and Dantzig [4]; Dantzig, De Haven, et al. [5]; and in the several papers by Shapiro [6-14].

CHEMIST is designed to allow the researcher to formulate a problem and get results with only a minimum knowledge of the inner workings of the program. Communication with the program is in English, chemical, and FORTRAN languages. The instructions are wholly contained herein, along with a documentation of the FORTRAN program. For the researcher who wants results, who wants to set up and solve a problem, it is sufficient only to read Chaps. I and II, and to follow closely the examples of Chap. III. The most difficult part will be to find and properly enter the free-energy parameters (equilibrium constants); the next most difficult, to set up "conceptual sub-compartments." These, as well as other problems, are illustrated herein along with examples of the action of the program on specified data in various circumstances. With this useful function in mind, most of

The maximum number of non-zero entries in the matrix is 460. Thus, if a problem has 60 constraints and 169 species, the maximum average number of non-zero entries per column is roughly 2.8.

System Approximations

CHEMIST is a generalized computer program for computing either the equilibrium distribution of species in a thermodynamically closed, idealized chemical milieu or, under certain circumstances, the steady-state distribution of a specified open system. The system is assumed to consist of a finite number of homogeneous phases or compartments, each of which has a specified pressure and temperature. Thus, isolated systems, which may change pressure and temperature as a result of chemical reaction, are not included unless the final pressure and temperature are known. At present, automatic adjustment of the reaction constants for changes in pressure or temperature are not made internally since, generally, the pressure and temperature dependent functions are not known. Similarly, allowance is not made for changes in external fields such as electrical or gravitational.

After a period of time, a reversible closed system will reach an equilibrium distribution for which the conservation laws hold and the mass-action laws hold for each reaction considering the activities of each species under the specified conditions. In such idealized conditions, knowing the intrinsic reaction constants and the activity coefficients at the given temperature and pressure, it is in principle possible to compute exactly the equilibrium distribution of

the output species. In practice, approximations to the ideal systems must be made, approximations which vary with the circumstances. And, since CHEMIST is designed primarily for equilibrium systems, the simulation of open or steady-state thermodynamic systems again requires certain approximations. Under these circumstances, "effective" or "apparent" mass-action constants are substituted for the intrnsic equilibrium constants as required whenever the effective constants are known, as from empirical measurements of species gradients in a biological system. The stationary states of certain time-independent open systems thus may be approximately calculated as though they were equilibrium systems, effective constants being used to approximate the non-equilibrium reactions.

Similarly, empirical parameters are determined and used in CHEMIST whenever possible for complex systems (open or closed) in order to allow automatically for the possibly unknown activity of a species or for its unknown osmotic coefficient. For example, the need for empirical parameters becomes particularly clear in concentrated protein solutions, as in the interior of the red cell. Frequently, appropriate empirical parameters are not known, and either closed-system parameters must be used or the appropriate value is determined by iterative procedures. In the case of human blood, it is probable that the hydrogen ion activity in the red cell cannot be measured, but must be assigned an apparent constant found by adjusting the H^+ activity to give the measured (hemolized) pH.

Thus, while in principle idealized closed systems may be computed exactly from the moles of components, the chemical reactions, and the mass-action constants, in practice,

closed systems (to say nothing of open systems) may only be approximated owing to the necessary use of certain approximate constants. Nevertheless, since this difficulty is not inherent in the computer but in the chemical system, calculations may proceed with CHEMIST with the same confidence and the same restrictions and caveats as without the computer program. Potentially, however, complex hypotheses which take account of the non-idealized system can be incorporated into the program; e.g., the calculation of activity or osmotic coefficients.

Time-dependent systems may also be approximated under certain conditions using CHEMIST. For example, if one reaction is very slow with respect to most of the reactions in the system, the fast reactions may be assumed to be in equilibrium at all times with, however, inputs and losses from and to the slow reaction. The slow reaction, with known kinetic parameters thus serves as the forcing function for the equilibrium system which gradually changes with time.

Names

The chemical-constituents input to the system are always referred to as components; the output, or product constituents, as species. Obviously, under varying conditions (e.g., changing a reaction equilibrium constant or incrementing a component) the equilibrium or steady-state distribution of species will vary. The species are, thus, dependent variables to be determined subject to the conditions of the experiment. The names of the species are also the names of the corresponding columns of the model

matrix; the names of the components (also referred to as constraints) are also the names of the corresponding rows of the model matrix. These names may be chosen arbitrarily from the user's special vocabulary to suit a particular problem--with the following conditions:

1) In the program, the names of the constraints may have 12 characters (see Chap. II below); but only the first six are significant, and each name must be unique in the first six characters. The species names have only six characters, usually the name or acronym of the chemical compound represented. Species names may be the same as component names, and a species in one compartment may have the same name as a species in another compartment, but any two species in a single compartment must have different names.
2) Each compartment must also be named, and the name need not be distinct from that of any variable. A component, variable, or compartment may be named in any convenient way--except that the variable hydrogen ion should be named H^+ since in the pH calculation the value of a variable with this name in a given compartment is used.
3) It is good practice to left adjust each name within its field since a name is not the same unless it is punched identically in each occurrence. Also, it is usually desirable to give the same chemical substance in different compartments the same name since the output will be better organized.

Units and Terminology

CHEMIST uses concentrations in the mole fraction scale for internal computations. Thus, for a system consisting of K chemical phases or compartments and having x_j moles of species X_j, j=1,...,N, the concentration of species X_j in the kth phase or compartment is $\hat{x}_j = x_j/\bar{x}_k$, where $\bar{x} = \sum_{j \in k} x_j$.

Provision is also made for printing the output of the program in other systems of units, e.g., moles per liter of water. The internal vector X1 may be printed in the output in addition to the vector X (the mole numbers) and the vector $\hat{x}$ (the mole fractions of each species). The components of X1, say, moles per liter of water for each species, must be computed using FORTRAN in the MAIN routine (or a subroutine which the user may write). X1 will automatically be printed by the OUTPUT subroutine by setting the toggle IV(23) = 1.

If T is absolute temperature and R is the gas constant, then, in ideal solutions, the chemical potential of each component μ_j is related to its mole fraction by

$$\mu_j = \mu_j^o + RT \ln \hat{x}_j \qquad (1)$$

where μ_j^o is the Gibbs' free energy per mole of the pure substance defined at specified standard conditions. With these μ_j^o and with $\hat{x}$, in the mole fraction scale, for non-ideal solutions an activity coefficient may be defined for each component so that

$$\mu_j = \mu_j^o + RT \ln \gamma_j \hat{x}_j \qquad (2)$$

where γ_j is a function of the concentration of all components in the phase as well as temperature and pressure. In biological systems, the concentrations of components does not vary over extreme ranges, and frequently γ_j is assumed to be a variable function only of T and P.

By transposing terms, as in algebra, a general chemical reaction

$$a_{11}\ X_1 + \ldots + a_{1r}\ X_r \rightleftharpoons a_{21}\ X_1, + \ldots +, a_{2r}\ X_r$$

may more conveniently be written

$$\sum_{i=1}^{r} a_i\ X_i \rightleftharpoons 0 \tag{3}$$

where the stoichiometric coefficients, a_i, may be zero or negative. When the mole numbers are strictly positive, a condition for equilibrium for this reaction is [15]

$$\sum_{i} a_i\ \mu_i = 0\ . \tag{4}$$

That is, the concentrations $\hat{x}_j$ will change until the sum of the Gibbs' free energy per mole of each species times its respective stoichiometric coefficient is zero.

For γ constituents of Eq. (3), any one, say X_j, may be singled out as the product species by moving it to the right-hand side and dividing the entire chemical equation by the stoichiometric coefficient a_j. In the notation of CHEMIST, "solving" each equation for exactly one product species gives

$$\sum_{\substack{i=1 \\ i\neq j}}^{r} a_{ij}\, X_i \rightleftarrows X_j \qquad (5)$$

where the X_i are components, at least one of which is required to produce each product species, X_j. For example,

$$HCO_3^- \rightleftarrows 1\ H_2O + 1\ CO_2 - 1\ H^+ , \quad K_j$$

where HCO_3^- is produced from three components with an equilibrium constant K_j.

Mathematical Summary

For each equation, as Eq. (3), we have the (ideal) Gibbs' free-energy function

$$F(x) = \sum_{i=1}^{r} x_i\, \mu_i = \sum_{i=1}^{r} x_i(\mu_i^o + RT \ln \hat{x}_i) \qquad (6)$$

where x is the mole-number vector $(x_1,\ldots,x_\gamma)$. For a single reaction at equilibrium, F(x) is minimized over all feasible values of x subject to any constraints on the system [15]. The values of x_i producing the minimum for F may be found by differentiating Eq. (6) and setting

$$\left(\frac{\partial F}{\partial x_i}\right)_{T,P} = 0 , \quad \text{for all } i . \qquad (7)$$

More generally, for any number of reactions and species in an ideal system at constant temperature and pressure,

F(x) is the sum of the Gibbs' free energy over all possible species in the output milieu:

$$F(x) = \sum_{j=1}^{N} x_j (\mu_j^o + RT \ln \hat{x}_j) \ , \tag{8}$$

and the necessary and sufficient condition for equilibrium is that F(x) be minimized subject to the constraints $x_j \geq 0$ for all j and the conservation of mass equations. For N equations of the form Eq. (3), where M components form N species, the conservation of mass equations may be written

$$\sum_{j=1}^{N} a_{ij} x_j = b_i \ , \quad i=1,\ldots,M \tag{9}$$

where b_i moles of each component are input to the system and, again, a_{ij} are the stoichiometric coefficients. Similarly, a conservation of charge equation may be written

$$\sum_{j=1}^{N} \nu_j x_j = 0 \tag{10}$$

where ν_j is the charge of each species and x_j are the mole numbers. It can be shown [6] that the minimization of F(x) over the range of the vector x and subject to the constraint Eqs. (9) with all $x_j > 0$ is equivalent to the existence of M Lagrange multipliers $\Pi = (\Pi_1,\ldots,\Pi_M)$, the chemical potentials μ_i for component i at equilibrium, which satisfy

$$\sum_{i=1}^{M} \Pi_i \, a_{ij} = \mu_j^o + RT \ln \hat{x}_j \ , \quad j=1,\ldots,N \qquad (11) \ ,$$

a result particularly useful for the computer program. In CHEMIST, iterative procedure is used [3] to find values of x and Π satisfying Eqs. (11). However, it is convenient to divide Eqs. (11) by RT so that μ_j^o/RT has the usual definition, $-\ln K_j$, for each reaction.

In CHEMIST notation, $c_j = \Delta F^o/RT = -\ln K_j$, i.e., from Eqs. (1) and (4), for the j<u>th</u> reaction of equilibrium

$$\sum_i (a_i \, \mu_i^o + RT \ln \hat{x}_i^{a_i}) = 0 \ ,$$

or

$$\sum_i a_i \, \mu_i^o = -RT \sum_i \ln \hat{x}_i^{a_i} = -RT \ln \prod_{i=1}^{\ell} \hat{x}_i^{a_i}$$

or

$$\Delta F^o = -RT \ln K_j$$

and

$$c_j = \frac{\Delta F^o}{RT} = -\ln K_j \ , \qquad (12)$$

where $K_j = \prod \hat{x}_i^{a_i}$ is the mass-action constant on the mole-fraction scale for the reaction, and ΔF^o is the usual notation [16] for the increment in Gibbs' free energy for the reaction.

For non-ideal solutions (as in most biological problems), we have

$$K_A = \prod_i (\gamma_i \hat{x}_i)^{a_i}$$

where K_A is the apparent constant for species in the milieu having activity coefficients γ_i. Usually, for calculations in non-ideal systems, using CHEMIST, K_A, the apparent constant, will be used in preference to K, the ideal solution constant. In this case, the free-energy parameter c_j in CHEMIST corresponds to an apparent constant,

$$c_j = -\ln_e K_A = -\ln K - \ln \prod_i \gamma_i^{a_i} \qquad (13)$$

where K_A is the apparent constant in mole fraction units.

Frequently, c_j has no such straightforward definition but is instead an empirical constant derived within the model to satisfy a given constraint; e.g., the Na^+ gradient across "active" membranes where the thermodynamic function is not known.

Equilibrium Constant

Note, of course, that K in Eq. (12) is obtained in the mole-fraction scale and is not numerically equal to the equilibrium constant for the same reaction on the molality or molarity scales. Consider a simple example: the chemical equation in dilute aqueous solution

$$RA \rightleftharpoons R + A$$

gives rise to the mass-action equation

$$\frac{(R)\ (A)}{(RA)} = K_d$$

where parenthesis indicate concentration on the volume scale, moles per liter, and K_d is the dissociation constant.

If we define K' on the mole fraction scale, however,

$$K' = \frac{(R)/ALITER \cdot (A)/ALITER}{(RA)/ALITER} = \frac{(R)\ (A)}{(RA)\ ALITER}$$

where ALITER = moles of water per liter at 1 atm pressure and temperature T (ALITER = 55.139673 at 37°C), and the parameter c from Eq. (13) is

$$c = -\ln K'$$

$$= -\ln \left(\frac{(R)\ (A)}{(RA)\ ALITER} \right) = -\ln K_d + \ln ALITER \ .$$

More generally, for the ith reaction

$$\sum_j a_j X_j \rightleftharpoons 0$$

at equilibrium, we have

$$K_j = \prod_j (x_j)^{a_j} \tag{14}$$

and

$$K_j' = \prod_j \hat{x}_j^{a_j} (ALITER)^{\Sigma a_j} \tag{15}$$

where, again, some of the a_j are negative, and

$$c_j \equiv -\ln K'_j \quad . \tag{16}$$

Also, particularly in the case of ionization reactions, with

$$pK_j = -\log_{10} K_j$$

then,

$$c_j = 2.30259\ pK_j + \ln \text{ALITER} \quad . \tag{17}$$

Finally, in each model of a chemical system using CHEMIST, certain of the c_j are set, logically, to zero. Practically speaking, this arises from the fact that if N equations are to be solved simultaneously, N equilibrium constants are required, except that for each algebraic constraint on the system (e.g., a conservation or mass equation) the number of equilibrium constants can be reduced by one. In the simultaneous system, an algebraic constraint on the mole numbers and an equilibrium constant are equivalent information. Thus, in a system having, say, eleven constraints (see the "Soda-Pop" example, Chap. III below), eleven of the c_j may be set arbitrarily--but the remaining c_j must be determined relative to the arbitrary values so set. It is most convenient to set the arbitrary c_j at zero. Also, with one c_j for each output species, it is convenient to set those c_j to zero for which the corresponding output species is produced from exactly one input

component. The examples in Chap. III below illustrate this practical matter; for theoretical justifications see Shapiro and Shapley [6] and Ref. 17.

Chapter II

PROGRAM DESCRIPTION

THE PROGRAM DECK

From the user's view, the "program" deck consists essentially of three parts: the FORTRAN deck, the Data deck, and a sequence of Control cards.

The FORTRAN deck consists of a MAIN routine and a large number of subroutines. All of the subroutines are compiled into an object deck of machine-language instructions (described in Chap. IV below). The user normally need not be familiar with the details of the subroutines. However, each time on the machine he will be concerned with the MAIN routine--described briefly in this chapter and in more detail in Chap. III.

By the Data deck is meant an ordered collection of data cards, described below, which together define a specific (unique) model of a chemical or biological subsystem. The Control cards consist of an ordered sequence of instructions inserted in the data deck by the user; in essence, these are a sequence of macroinstructions to the computer for manipulating the data, and consist of two kinds: Data Controls, which set or define numerical data; and Verb Controls, which are instructions to compute with or transform the data in some way.

In summary, for a typical machine pass the user has a MAIN routine, the subroutine object deck, and a specific data deck which also contains Control instruction cards.

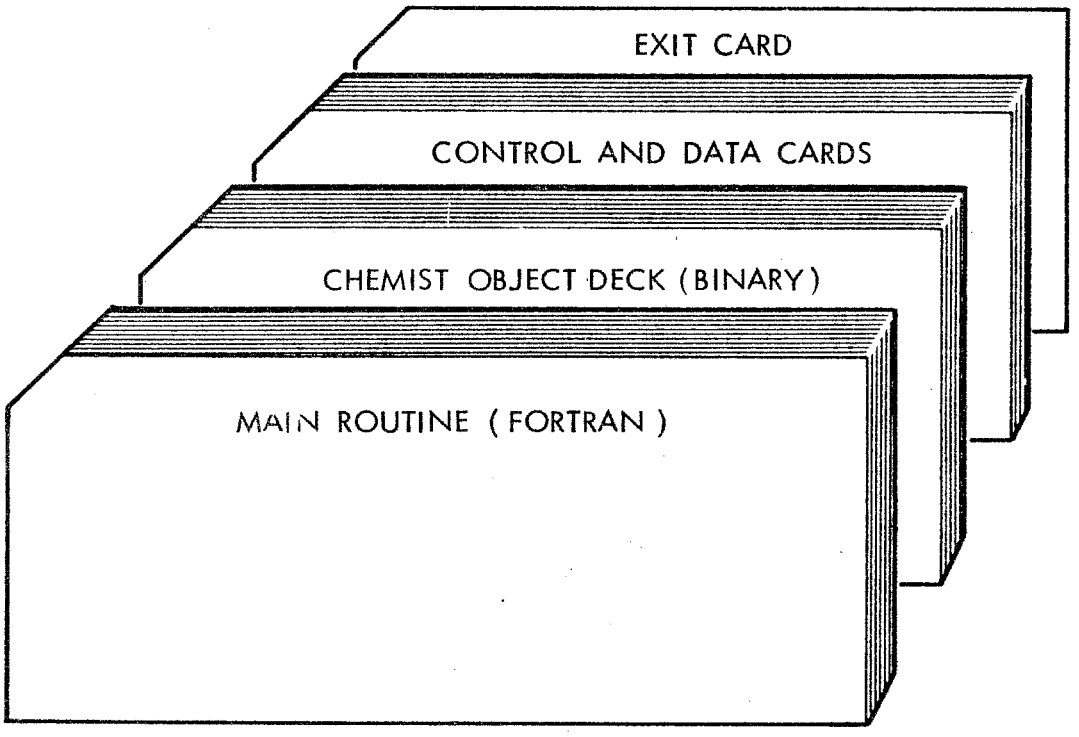
THE MAIN ROUTINE

The MAIN routine is a FORTRAN program having nominal control of the machine pass. Control usually passes from the MAIN routine to subsequent Control cards; but, should the user choose, control of the machine pass can be retained in the MAIN routine or returned to the MAIN routine at a later time for special purposes. If the user does not exercise this choice, the MAIN routine is very short and serves only to introduce the START subroutine, passing control to subsequent Control cards. In this event, the MAIN routine contains (besides the COMMON, EQUIVALENCE, and INTEGER cards) only the FORTRAN instructions to CALL START (a subroutine containing such nominal constants as the moles per liter of water at 37°C) and CALL INPUT (a subroutine passing control to the first subsequent Control card in the data deck). Of course, as in any FORTRAN program, the last card of the MAIN routine is an END card.

If the user does choose to return control to the MAIN routine from the Control cards (see RETURN, p. 48 below), he may insert any correct list of FORTRAN instructions for manipulating the data just after the last CALL INPUT statement. (Since such a FORTRAN list may be quite varied, detailed discussion and examples will be deferred until Chap. III below.) For now, the MAIN routine is the first deck in the object program (the deck of cards presented to the computer) and consists of the following cards:

```
0 $IBFTC NZSC    NODECK
  C                        4-7-67
1       COMMON AIJ(460), IRCW(460), JCOL(460)
  C
2       COMMON  /INPT/KA(12),KB(12),BBB(60,5),PH(25),T(20),BMULT(5)
  C
3       COMMON /SLVE/IV(30),TOL(20),NR(60,2),B(60),PIE(75),V1(75),V2(75),
       1         V3(75),V4(75),X(170),XPF(170),KN(170),C(170),X1(170),
       2         X2(170),X3(170),XBAR(25),NAN(25,2),KL(26), R(75,75),
       3         JCOMP(170),FE,FE2,ERMB,XEPB,NEMB,ERPA,XEMA,NEMA
  C
4       EQUIVALENCE (IV(1),M),(IV(2),MEND),(IV(3),NCOMP),(IV(4),N,NTOT),
       1         (IV(5),NIT),(IV(6),NOT),(IV(7),PF),(IV(8),ITER),
       2         (IV(9),ITMAX),(IV(10),IERROR),(IV(11),LASTCP),(IV(12),KE),
       3         (IV(13),MAXM),(IV(14),MAXP),(IV(15),MAXN),(IV(16),MAXMD),
       4         (IV(17),END),(IV(18),BLANK),(IV(19),H2O),(IV(20),HPLUS),
       5         (IV(21),NCYCLE),(IV(22),NBSTAR),(IV(23),KPF),
       6         (IV(24),NAIJ),(IV(25),MAXAIJ),(IV(26),IOPT)
5       EQUIVALENCE (TOL(3),XMIN),(TOL(4),XSTART),(TOL(5),BARMIN),
       1           (TOL(11),ALITER),(TOL(12),RT)
  C
6       INTEGER PF
7       INTEGER END,BLANK,H2O,HPLUS
  C
10      CALL START
11      CALL INPUT
44      CALL EXIT
45      END
```

The MAIN routine is followed in the object program by the subroutines, the end of which is marked by the $ENTRY or $DATA card to indicate that all subsequent cards are Control cards and data cards. Infrequently, it may be necessary to recompile the binary subroutine deck because of modifications; but for the more usual pass, we have the sequence: MAIN routine, binary subroutines, CONTROL and DATA cards, EXIT, as follows:



CONTROL CARDS AND DATA ARRAYS

This program may be thought of as a single pass interpretive program.† That is, the program is supplied with a set of Control cards, each of which is a macroinstruction which the computer executes as soon as the card is read. After the instruction has been executed, the next Control card is read in sequence, until the Control cards terminate with an EXIT. The computer "interprets" each macroinstruction, thus determining the operations it should perform including, finally, reading the next macroinstruction. Once read and executed, a Control card is never re-read,

†The following discussion is based upon unpublished documentation accompanying R. J. Clasen's program produced at RAND.

control passing to the next card. Subsequent cards in the sequence are thus independent and may be organized at the discretion of the user, subject only to natural logical ordering (e.g., one would not call for the instruction SOLVE before the data are read in). Thus, an interpretive program may be different for each pass on the computer according to the dictates of the particular problem.

Immediately following the CHEMIST subroutines in the object deck presented to the computer are the Control and data cards, which together constitute the array of the problem to be solved in the computer pass. Each Control card is a macroinstruction to the computer to treat current data in a particular way. Thus, the Control cards and the data cards are intermixed in a logical order.

The Control cards are of two types, <u>data</u> Controls and <u>verb</u> Controls. A data Control card is a macroinstruction to the computer to interpret and store the numerical data card or cards immediately following. A verb Control card, being more general, is a macroinstruction to treat data already in computer memory. Thus, a verb Control card is always followed by another Control card, but a data Control card is always followed by data. In the following list of available Control cards, only the first six characters of each name are significant; a Control card consists of these six characters punched in columns 1-6, followed by any other characters or blanks through column 72. In some instances, however, the seventh column is used as a subscript (indicated by N in Table I) and columns 8-12 must be blank.

Table I

CONTROL CARDS

I. Data Control Cards

	Control Card Name	FORTRAN Equivalent[a]		
1.	ROWS	CALL ROWS (0) (followed by data)		
2.	MATRIX	CALL MATRIX (0) (followed by data) MEND = M + NCOMP		
3.	VECTORX	CALL VECTOR (0) (followed by data)		
4.	SCALEC	CALL SCALEC (6H....., VALUE)[b]		
5.	ADDB N	NBSTAR = N CALL ROWS (2) (followed by data)	or:	BBB(I,N) = BBB(I,N) + AA
6.	ALTERBN	NBSTAR = N CALL ROWS (1) (followed by data)	or:	BBB(I,N) = AA
7.	B N	NBSTAR = N CALL ROWS (3) (followed by data)	or:	BBB(I,N) = 0.0, ALL I BBB(I,N) = AA(I), ALL I
8.	ALTERA	CALL MATRIX (-1) (followed by data)	or:	A(I,J) = AA
9.	ALTERC	CALL MATRIX (1) (followed by data)	or:	C(J) = AA and A(I,J) = AI
10.	ADDC	CALL MATRIX (2)	or:	C(J) = C(J) + AA and A(I,J) = A(I,J) + AA

II. Special Data Controls

1.	CYCLE	NCYCLE = K
2.	LIMIT	ITMAX = K
3.	LITER	ALITER = AA
4.	MULTIPLIERS	BMULT(I) = AA(I), I = 1,5
5.	RT	RT = AA
6.	TOLERANCES	TOL(I) = AA(I), I = 1,6

Table I--Continued

III. Verb Controls

	Control Card Name	FORTRAN Equivalent
1.	(------)COMMENT	---
2.	TITLE	(See Section F)
3.	SOLVE	MEND = M + NCOMP CALL SOLVE
4.	OUTPUT	CALL OUTPUT
5.	RETURN	---
6.	DELETE	CALL DELETE(1)
7.	PRINTROWS	CALL ROWS(-2)
8.	PUNCHROWS	CALL ROWS(-3)
9.	PRINTMATRIX	CALL PUNCHM(0)
10.	PUNCHMATRIX	CALL PUNCHM(1)
11.	PRINTTABLEAU	CALL PRINTT
12.	PUNCHX	CALL VECTOR(1)
13.	PRINTPIE	CALL PRINTP
14.	SIMPLEX	CALL LP (MON) IF (MON.NE.0) GO TO "ERROR"
15.	MESSAGES	PF = 0
16.	ALLMESSAGES	PF = 1
17.	NOMESSAGES	PF = -1
18.	GOTO AA	---
19.	IFGOTOAA	IF (IOPT.NE.1) GO TO AA
20.	SYMBOLAA	---
21.	RELAXBN	NBSTAR = N CALL ROWS (4)
22.	JACOB	CALL JACOBS(0,0,1)
23.	MINIJACOB	CALL JACOBS(0,2,1)
24.	EJECT	CALL PAGE
25.	CLEAR	CALL START
26.	EXIT	CALL EXIT
27.	END	---

---

[a]While the Control Card is inserted in the data deck with the appropriate data cards immediately following, the FORTRAN equivalent cards are inserted in the (FORTRAN) MAIN routine with the data cards still in the data deck at the appropriate place to be read.

[b]While the Control Card SCALEC may be followed by any number of data cards, the FORTRAN statement CALL SCALEC (6H.....,V) must be used once for each compartment to be scaled.

The Control cards of Table I are listed with a FORTRAN equivalent--i.e., if the user chooses to maintain control of the program in the MAIN routine (whose statements are in FORTRAN) rather than relinquish control to the Control cards, he may obtain the equivalent action of the Control card by using the listed FORTRAN statements.

Note that to relinquish control, the user will insert the FORTRAN statement CALL INPUT in the MAIN routine. This statement transfers control to the next unused Control card in the data deck. To regain control in the MAIN routine, the user will insert the Control card RETURN at the appropriate place in the data deck. Control is thus transferred to the next FORTRAN statement after the last used CALL INPUT statement in the MAIN routine.

To facilitate the following discussion of the Control cards, a list of the names of the program variables and an explanation is given in Table II.

In addition to the Control cards in Table I, the program will recognize an "END" card even though it does not appear at the end of a set of data cards. In this case, the "END" card causes no action.

Any Control card that does not have one of the above words in the first six columns, and the first six columns are <u>not</u> blank is a Title card--which will cause the page to be restored and the punches in the card to be printed as the title. If the first six columns are blank, no action other than printing the card will be taken. Care should be taken not to mispunch a Control card so that it will be treated as a Title card. Two Title cards in succession will cause the program to EXIT.

Table II

INTERNAL VARIABLES

Name	Dimension	Meaning
AIJ	460	Coefficient of matrix entry
ALITER	1	Moles per liter H2O at Temperature T
B	60	Total values of constraint equations
BBB	60 x 5	Components of value of constraint equations
BLANK	1	Contains 6 blank characters
BMULT	5	Arbitrary component multipliers
C	170	Free energy parameters
END	1	Contains characters END followed by 3 blanks
ERMA	1	RMS equilibrium error (mass action)
ERMB	1	RMS mass balance error
FE	1	Value of objective function
FE2	1	Value of objective function times RT
H2O	1	Contains characters H2O followed by 3 blanks
HPLUS	1	Contains characters H+ followed by 4 blanks
IERROR	1	Flag for termination in SOLVE, normal = 1
IOPT	1	Flag for optimal solution, optimal = 1
IROW	460	Row number for matrix entry
ITER	1	Iteration number
ITMAX	1	Maximum number of iterations allowed
IV	30	Constants, see Equivalence in MAIN
JCOL	460	Column number for matrix entry
JCOMP	170	Contains compartment number for each species
KA	12	Temporary storage for incoming BCD
KB	12	Problem title storage
KE	1	Flag for singular matrix
KL	26	List of first variables in each compartment
KN	170	Names of output species
KPF	1	Flag for an extra line of output
LASTCP	1	Number of compartment where XBAR is too small
M	1	Number of constraints
MAXAIJ	1	Maximum number of matrix entries allowed
MAXM	1	Maximum number of constraints
MAXMD	1	Maximum size for M+NCOMP
MAXN	1	Maximum number of columns
MAXP	1	Maximum number of compartments
MEND	1	Number of simultaneous equations = M + NCOMP

Table II--Continued

Name	Dimension	Meaning
N	1	Number of output species, same as NTOT
NAIJ	1	Number of entries in matrix array
NAM	25 x 2	Compartment names
NBSTAR	1	Subscript for selected component of B
NCOMP	1	Number of compartments
NCYCLE	1	Number of compartments (or columns) printed per page
NEMA	1	Column number of maximum equivalent error
NEMB	1	Row number of maximum mass balance error
NIT	1	Number FORTRAN logical unit (nominally 5), input
NOT	1	Number FORTRAN logical unit (nominally 6), output
NTOT	1	Number of unknown variables or output species
NR	60 x 2	Row (constraint) names
PF	1	Flag for message print
PH	25	Computed pH in each compartment
PIE	75	Lagrange multipliers
R	75 x 75	Matrix for linear equations
RT	1	Product of gas constant and temperature
T	20	Temporary storage
TOL	20	Computing decision tolerances
V1	75	Scratch vector
V2	75	Scratch vector
V3	75	Scratch vector
V4	75	Scratch vector
X	170	Unknowns, variables, output species
XBAR	25	Sum of x in compartment
XEMA	1	Maximum equilibrium error
XEMB	1	Maximum mass balance error
XMF	170	Mole fractions of x in compartment
X1	170	Scratch vector
X2	170	Scratch vector
X3	170	Scratch vector

There are no general restrictions on ordering control and data cards. The only major restrictions concern the fact that certain names must be defined before certain other quantities can be input. All input numbers in arrays (matrix or vectors) are identified with alphanumeric names which have no relation to the position of the entry in the array.

Finally, Table III is a listing of the data deck from a sample problem, a simple system called Soda Pop. This problem is discussed below in detail (p. 56 ff.), but the listing exemplifies the use of Control cards; in the subsequent discussion, Soda Pop will be used to illustrate the action of each card.

Here we note that the first card is a title card and the last card is an "EXIT" card. The "EXIT" card terminates the run on the computer. However, the program is not restricted to solving only one problem per pass. As many problems as desired may be stacked and solved on the same run, with a "CLEAR" control card separating each problem and the "EXIT" card appearing at the end of the final problem.

DATA CONTROL CARDS

Each of the following data Control cards is immediately followed in the deck by an array of data whose format is described below. The number of data cards following each Control card varies with the problem; therefore, each data array must be terminated with an END Control card.

The action of the Control card is equivalent to the FORTRAN statements listed in each case; the user, therefore,

Table III

DATA DECK FOR EXPERIMENTAL SODA-POP DECK

```
EXPERIMENTAL SODA POP DECK      5-1-67
ROWS
O2              0.              2.09900E-01 1.31500E-01 0.          0.
CO2             0.              3.00000E-04 5.26300E-02 0.          0.
N2              0.              7.89800E-01 7.54000E-01 0.          0.
H2O             55.13967        0.          6.10600E-02 0.          0.
H+              1.0      -03    0.          0.          0.          0.
CL-             140.     -03    0.          0.          0.          0.
NA+             130.     -03    0.          0.          0.          0.
K+              20.      -03    0.          0.          0.          0.
GLUCOSE         10.0      -3    0.          0.          0.          0.
LACTIC-         10.0      -3    0.          0.          0.          0.
MISC-           1.0       -3    0.          0.          0.          0.
END
MATRIX
GAS PHASE
        O2      -10.93999994    1.0 O2
        CO2      -7.74074000    1.0 CO2
        N2      -11.51999998    1.0 N2
        H2O      2.79           1.0 H2O
LIQUID PHASE
        O2       0.             1.0 O2
        CO2      0.             1.0 CO2
        N2       0.             1.0 N2
        H+       0.             1.0 H+
        OH-      39.39          1.0 H2O      -1.0H+
        H2O      0.             1.0 H2O
        CL-      0.             1.0 CL-
        NA+      0.             1.0 NA+
        K+       0.             1.0 K+
        GLUCOS   0.             1.0 GLUCOS
        LACTIC   0.             1.0 LACTIC
        HCO3-    18.0556        1.0 CO2      1.0 H2O     -1.0 H+
        H2CO3    6.566          1.0 CO2      1.0 H2O
        CO3=     45.6616        1.0 CO2      1.0 H2O     -2.0 H+
        MISC     -20.128        1.0 MISC-    1.0 H+
        MISC-    0.             1.0 MISC-
END
MULTIPLIERS
 1.             0.              1000.       0.          0.          0.
VECTORX      EXPERIMENTAL SODA POP DECK      5-1-67
GAS PHASE    O2      1.31500E 02CO2      5.26287E 01N2       7.54000E 02
GAS PHASE    H2O     6.10230E 01
LIQUID PHASEO2       1.29516E-04CO2      1.27070E-03N2       4.15794E-04
LIQUID PHASEH+       3.33229E-05OH-      7.18394E-10H2O      5.51766E 01
LIQUID PHASECL-      1.40000E-01NA+      1.30000E-01K+       2.00000E-02
LIQUID PHASEGLUCOS 1.00000E-02LACTIC 1.00000E-02HCO3-  3.03114E-05
LIQUID PHASEH2CO3  1.77831E-06CO3=     5.17534E-11MISC     9.96989E-04
LIQUID PHASEMISC-  3.01078E-06
END
SOLVE
OUTPUT
EXIT
```

has a choice of data control by the Control cards or by direct modification in the MAIN routine.

1. ROWS

a) Each data card following the ROWS Control card gives the name and the value of a constraint equation. Examples of constraints on the system are the mass conservation equations, charge conservation equations, and subgroup accounting equations. The name of the row (constraint) is arbitrary (but must be unique in the first six letters); the total value of the row (constraint) is the value of the variable B(I). Allowance is made for five possible sources for components; B(I) is the sum of all five times the appropriate multiplier as follows:

b) B(I), the value of the constraint, is computed as

$$B(I) = \sum_{J=1}^{J=5} BBB(I,J) * BMULT(J) \ .$$

c) Each data card is punched with the following information:

Columns	Data in I^{th} data card
1-12	Name of row I.
13-24	BBB(I,1) Floating point numbers.
25-36	BBB(I,2)
37-48	BBB(I,3)
49-60	BBB(I,4)
61-72	BBB(I,5)

d) Row names must be unique in the first 6 characters, although 12 characters are allowed for the word.

e) The FORTRAN equivalent consists of specifying the BBB(I,J) as required followed by the statement CALL ROWS(-1) which evaluates B(I).

f) Example: The immediate result of the Control card ROWS is that the input data is reproduced on the output printer as follows:

ROWS						
1	O2	0.	2.0990000E-01	1.3150000E-01	0.	0.
2	CO2	0.	3.0000000E-04	5.2630000E-02	0.	0.
3	N2	0.	7.8980000E-01	7.5400000E-01	0.	0.
4	H2O	5.5139670E 01	0.	6.1060000E-02	0.	0.
5	H+	1.0000000E-03	0.	0.	0.	0.
6	CL-	1.4000000E-01	0.	0.	0.	0.
7	NA+	1.3000000E-01	0.	0.	0.	0.
8	K+	2.0000000E-02	0.	0.	0.	0.
9	GLUCOSE	1.0000000E-02	0.	0.	0.	0.
10	LACTIC-	1.0000000E-02	0.	0.	0.	0.
11	MISC-	1.0000000E-03	0.	0.	0.	0.

2. MATRIX

a) The data cards following this Control card are of two kinds: 1) Compartment name cards (e.g., PLASMA or RED CELLS), which must be punched in the first 12 columns (unique in the first six); and 2) column or species cards containing the name of an output species, stoichiometric coefficients of the constraint equations (e.g., of the chemical equation generating that species or of the charge conservation equation), and the free-energy parameter for the reaction generating that species.

b) The format for cards of Type 2 is:

Columns	Meaning
1- 6	Blank
7-12	Column Name, X(J), Output Species
13-24	Free-energy constant, c(J), floating point
25-30	Matrix coefficient, $a(I_1,J)$, floating point
31-36	Name of Row I_1 (which the above coefficient is in)
37-42	Matrix coefficient, $a(I_2,J)$, floating point
43-48	Name of Row I_2
49-54	Matrix coefficient, $a(I_3,J)$, floating point
55-60	Name of Row I_3
61-66	Matrix coefficient, $a(I_4,J)$, floating point
67-72	Name of Row I_4

c) If any row name is blank, the corresponding matrix entry will be ignored. The row names used are the first six characters of the row name as entered by the ROWS Control card. If a row name has not been defined by the ROWS Control card, the entry will be skipped and a message saying that a row is undefined will be printed. If more than four matrix entries are needed for a matrix column, two or more matrix column data cards may be placed next to each other, each with the same name in columns 7-12. When there are two or more matrix column cards in the same compartment with the same name, the value of the free-energy constant is obtained from the first of these cards.

After the data for all the matrix columns in one compartment have been given, another compartment name may be assigned. Then, following this compartment name card, the matrix-column data cards for the coefficients in this compartment are given as above.

After all of the data cards, an END Control card must be used.

d) Example: The immediate result of the MATRIX Control card is to reproduce the input data on the output sheet as follows (the END card is omitted):

```
MATRIX

   GAS PHASE
 1       O2        -10.940000    1.000 O2
 2       CO2        -7.740740    1.000 CO2
 3       N2        -11.520000    1.000 N2
 4       H2O         2.790000    1.000 H2O

   LIQUID PHASE
 5       O2          0.000000    1.000 O2
 6       CO2         0.000000    1.000 CO2
 7       N2          0.000000    1.000 N2
 8       H+          0.000000    1.000 H+
 9       OH-        39.390000    1.000 H2O       -1.000 H+
10       H2O         0.000000    1.000 H2O
11       CL-         0.000000    1.000 CL-
12       NA+         0.000000    1.000 NA+
13       K+          0.000000    1.000 K+
14       GLUCOS      0.000000    1.000 GLUCOS
15       LACTIC      0.000000    1.000 LACTIC
16       HCO3-      18.055600    1.000 CO2         1.000 H2O   -1.000 H+
17       H2CO3       6.566000    1.000 CO2         1.000 H2O
18       CO3=       45.661600    1.000 CO2         1.000 H2O   -2.000 H+
19       MISC      -20.128000    1.000 MISC-       1.000 H+
20       MISC-       0.000000    1.000 MISC-
```

3. VECTORX

a) The data cards contain initial estimates for the values of the vector x, up to three values per card. The format is:

Column	
1-12	Compartment name
13-18	1st column name
19-30	1st estimated value
31-36	2nd column name
37-48	2nd estimated value
49-54	3rd column name
55-66	3rd estimated value

b) If any compartment name or column name has not previously been input by MATRIX data, an error condition is set up and a message is given. In this event, VECTORX is ignored and the SOLVE subroutine obtains an initial estimate of the vector x from SIMPLEX. Computation continues. A good initial guess saves computation time.

c) The FORTRAN equivalent is to read the names of the vector x in any format.

d) Example (see Table III, p. 30).

4. SCALEC

a) Scales up (or down) the size of a complete compartment in a chemical system; e.g., double the size of the plasma compartment, or halve the red cell compartment. If the Kth compartment is to be incremented, and J is a species in the Kth compartment, then the action of SCALEC is

$$X(J) = X(J) + SCALE(K) * X(J)$$

$$B(I) = B(I) + SCALE(K) * \sum_{J=1}^{N} A(I,J) * X(J)$$

and

$$BBB(I,1) = BBB(I,1) + SCALE(K) * \left(\sum_{J=1}^{N} A(I,J) * X(J) \right) \Big/ BMULT(1)$$

where SCALE(K) is a number input.

b) The data cards following SCALEC have the format:

Columns 1-12 compartment name
Columns 13-24 SCALE(K), with decimal point

c) The FORTRAN equivalent is CALL SCALEC (6H....., SCALE(K)), where the first six letters of the compartment name are inserted after the Hollerith symbol.

d) Note that since the action of SCALEC is to algebraically add an amount to an existing compartment, the result is (1 + SCALE(K)) times the existing compartment. Of course, SCALE(K) may be negative.

e) Example: The Control and data cards

```
SCALEC
LIQUID PHASE      0.50
```

give the following new ROWS (cf. ROWS, pp. 31-32 above).

```
SCALEC
** COMPARTMENT 'LIQUID PHASE' HAS BEEN SCALED BY    5.0000000 E-01. **

PRINTROWS

ROW NAME                    B                    B1                   B2                   B3
                                          MULT.= 1.0000000E 00  -0.          1.0000000E 03

  1  O2              1.3150006E 02      6.4757950E-05      2.0990000E-01      1.3150000E-01
  2  CO2             5.2630651E 01      6.5139239E-04      3.0000000E-04      5.2630000E-02
  3  N2              7.5400021E 02      2.0789692E-04      7.8980000E-01      7.5400000E-01
  4  H2O             1.4378798E 02      8.2727981E 01      0.                 6.1060000E-02
  5  H+              1.5000001E-03      1.5000001E-03      0.                 0.
  6  CL-             2.1000000E-01      2.1000000E-01      0.                 0.
  7  NA+             1.9500000E-01      1.9500000E-01      0.                 0.
  8  K+              3.0000000E-02      3.0000000E-02      0.                 0.
  9  GLUCOSE         1.5000000E-02      1.5000000E-02      0.                 0.
 10  LACTIC-         1.5000000E-02      1.5000000E-02      0.                 0.
 11  MISC-           1.5000001E-03      1.5000001E-03      0.                 0.
END   OF ROWS IN STORAGE
```

* * *

The following three Control cards are used to change the values of BBB(I,J), i.e., they adjust the amounts of input components or values of constraint equations (see ROWS, pp. 31-32). The total value of the Ith row is therby altered for the next solution. The format of the data cards following the Control card is:

Columns 1-12 The (unique in first 6 cols.) row name.
Columns 13-24 AA, a floating point number.

The Controls B and ALTERB will define new row names if the name has not previously been used, ADDB will not. B first zeros the Nth B column and then does ALTERBN.

5. ADDB J

a) Adds the read value AA to the previous BBB(I,J).

b) Example: Use of the Control cards ADDB..3 and PRINTROWS results in (cf. ROWS, pp. 31-32):

```
ADDB  3
  CO2              5.2630000E-03
  END    (NEW ROWS ARE *-END)

PRINTROWS

ROW NAME                    B                    B1                   B2                   B3
                                      MULT.= 1.0000000E 00  -0.                  1.0000000E 03

  1  O2              1.3149997E 02      -3.2378972E-05     2.0990000E-01     1.3150000E-01
  2  CO2             5.7892673E 01      -3.2569616E-04     3.0000000E-04     5.7892999E-02
  3  N2              7.5399990E 02      -1.0394845E-04     7.8980000E-01     7.5400000E-01
  4  H2O             1.0240551E 02       4.1345515E 01     0.                6.1060000E-02
  5  H+              7.5000013E-04       7.5000013E-04     0.                0.
  6  CL-             1.0500000E-01       1.0500000E-01     0.                0.
  7  NA+             9.7499998E-02       9.7499998E-02     0.                0.
  8  K+              1.4999999E-02       1.4999999E-02     0.                0.
  9  GLUCOSE         7.4999992E-03       7.4999992E-03     0.                0.
 10  LACTIC-         7.4999992E-03       7.4999992E-03     0.                0.
 11  MISC-           7.5000012E-04       7.5000012E-04     0.                0.
END   OF ROWS IN STORAGE
```

6. ALTERBJ

a) Alters current value of BBB(I,J) to a new value.

b) Example (cf. ROWS, pp. 31-32):

```
ALTERB1
  NA+              1.4300000E-01
  END    (NEW ROWS ARE *-END)

PRINTROWS

ROW NAME                    B                    B1                   B2                   B3
                                      MULT.= 1.0000000E 00  -0.                  1.0000000E 03

  1  O2              1.3149997E 02      -3.2378972E-05     2.0990000E-01     1.3150000E-01
  2  CO2             5.7892673E 01      -3.2569616E-04     3.0000000E-04     5.7892999E-02
  3  N2              7.5399990E 02      -1.0394845E-04     7.8980000E-01     7.5400000E-01
  4  H2O             1.0240551E 02       4.1345515E 01     0.                6.1060000E-02
  5  H+              7.5000013E-04       7.5000013E-04     0.                0.
  6  CL-             1.0500000E-01       1.0500000E-01     0.                0.
  7  NA+             1.4300000E-01       1.4300000E-01     0.                0.
  8  K+              1.4999999E-02       1.4999999E-02     0.                0.
  9  GLUCOSE         7.4999992E-03       7.4999992E-03     0.                0.
 10  LACTIC-         7.4999992E-03       7.4999992E-03     0.                0.
 11  MISC-           7.5000012E-04       7.5000012E-04     0.                0.
END   OF ROWS IN STORAGE
```

7. B J

a) Zeros out the Jth B vector, and then reads new values into Jth column of B.

b) Example (cf. ROWS, pp. 31-32):

```
B     1
  H2O              5.5000000E 01
  CL-              1.4500000E-01
  NA+              1.3500000E-01
  END    (NEW ROWS ARE *-ED)

PRINTROWS

ROW NAME                    B                    B1                   B2                   B3
                                          MULT.= 1.0000000E 00  -0.                  1.0000000E 03

  1  O2              1.3150000E 02        0.                   2.0990000E-01        1.3150000E-01
  2  CO2             5.7892999E 01        0.                   3.0000000E-04        5.7892999E-02
  3  N2              7.5400000E 02        0.                   7.8980000E-01        7.5400000E-01
  4  H2O             1.1606000E 02        5.5000000E 01        0.                   6.1060000E-02
  5  H+             -0.                   0.                   0.                   0.
  6  CL-             1.4500000E-01        1.4500000E-01        0.                   0.
  7  NA+             1.3500000E-01        1.3500000E-01        0.                   0.
  8  K+             -0.                   0.                   0.                   0.
  9  GLUCOSE        -0.                   0.                   0.                   0.
 10  LACTIC-        -0.                   0.                   0.                   0.
 11  MISC-          -0.                   0.                   0.                   0.
END   OF ROWS IN STORAGE
```

8. ALTERA

a) Used to change a stoichiometric coefficient in the matrix. The data card format is the same as described for "MATRIX." The free-energy parameter (i.e., cols. 13-24) is ignored on the data cards. If a compartment or column name is read which was not input by the previous "MATRIX," the problem will be cut off.

b) In this example, the coefficient of H^+ is changed from 1 to 2 (cf. MATRIX, pp. 32-34 above):

```
ALTERA
LIQUID PHASE          -0.000000   -0.000        -0.000        -0.000        -0.000
      MISC           -21.000000    2.000H+      -0.000        -0.000        -0.000
END                   -0.000000   -0.000        -0.000        -0.000        -0.000

PROBLEM HAS    11 ROWS,    20 COLUMNS,    2 COMPARTMENTS,     27 NON ZERO MATRIX ENTRIES.

PRINTMATRIX

MATRIX IN STORAGE

   GAS PHASE
 1       O2            -10.940000          1.000 O2
 2       CO2            -7.740740          1.000 CO2
 3       N2            -11.520000          1.000 N2
 4       H2O             2.790000          1.000 H2O

   LIQUID PHASE
 5       O2              0.000000          1.000 O2
 6       CO2             0.000000          1.000 CO2
 7       N2              0.000000          1.000 N2
 8       H+              0.000000          1.000 H+
 9       OH-            39.390000          1.000 H2O      -1.000 H+
10       H2O             0.000000          1.000 H2O
11       CL-             0.000000          1.000 CL-
12       NA+             0.000000          1.000 NA+
13       K+              0.000000          1.000 K+
14       GLUCOS          0.000000          1.000 GLUCOS
15       LACTIC          0.000000          1.000 LACTIC
16       HCO3-          18.055600          1.000 CO2      1.000 H2O      -1.000 H+
17       H2CO3           6.566000          1.000 CO2      1.000 H2O
18       CO3=           45.661600          1.000 CO2      1.000 H2O      -2.000 H+
19       MISC          -20.128000          1.000 MISC-    2.000 H+
20       MISC-           0.000000          1.000 MISC-
END   OF MATRIX IN STORAGE
```

9. ALTERC

a) Same as ALTERA, except that the free-energy parameters, c_j values, are not ignored. The program uses the last appropriate c_j value read in the ALTERC data.

b) This example alters the HCO_3^-, c_j, and the H^+ coefficient (cf. ALTERA above):

```
ALTERC
LIQUID PHASE          -0.000000   -0.000        -0.000        -0.000        -0.000
      HCO3-           16.250040    1.000H+      -0.000        -0.000        -0.000
END                   -0.000000   -0.000        -0.000        -0.000        -0.000

PROBLEM HAS    11 ROWS,    20 COLUMNS,    2 COMPARTMENTS,     27 NON ZERO MATRIX ENTRIES.

PRINTMATRIX

MATRIX IN STORAGE

   GAS PHASE
 1       O2            -10.940000          1.000 O2
 2       CO2            -7.740740          1.000 CO2
 3       N2            -11.520000          1.000 N2
 4       H2O             2.790000          1.000 H2O

   LIQUID PHASE
 5       O2              0.000000          1.000 O2
 6       CO2             0.000000          1.000 CO2
 7       N2              0.000000          1.000 N2
 8       H+              0.000000          1.000 H+
 9       OH-            39.390000          1.000 H2O      -1.000 H+
10       H2O             0.000000          1.000 H2O
11       CL-             0.000000          1.000 CL-
12       NA+             0.000000          1.000 NA+
13       K+              0.000000          1.000 K+
14       GLUCOS          0.000000          1.000 GLUCOS
15       LACTIC          0.000000          1.000 LACTIC
16       HCO3-          16.250040          1.000 CO2      1.000 H2O      1.000 H+
17       H2CO3           6.566000          1.000 CO2      1.000 H2O
18       CO3=           45.661600          1.000 CO2      1.000 H2O     -2.000 H+
19       MISC          -20.128000          1.000 MISC-    2.000 H+
20       MISC-           0.000000          1.000 MISC-
END   OF MATRIX IN STORAGE
```

10. ADDC

a) Using this Control card, the format for the data cards is the same as the format for the data cards of "MATRIX." The numerical entries are added appropriately to both the C(J) and A(I,J) values.

b) Example (cf. ALTERC above):

```
ADDC
LIQUID PHASE          -0.000000   -0.000      -0.000      -0.000      -0.000
      HCO3-            1.805560   -2.000H+    -0.000      -0.000      -0.000
END                   -0.000000   -0.000      -0.000      -0.000      -0.000

PROBLEM HAS    11 ROWS,    20 COLUMNS,    2 COMPARTMENTS,   27 NON ZERO MATRIX ENTRIES.

PRINTMATRIX

MATRIX IN STORAGE

   GAS PHASE
 1       O2          -10.940000      1.000 O2
 2       CO2          -7.740740      1.000 CO2
 3       N2          -11.520000      1.000 N2
 4       H2O           2.790000      1.000 H2O

   LIQUID PHASE
 5       O2            0.000000      1.000 O2
 6       CO2           0.000000      1.000 CO2
 7       N2            0.000000      1.000 N2
 8       H+            0.000000      1.000 H+
 9       OH-          39.390000      1.000 H2O      -1.000 H+
10       H2O           0.000000      1.000 H2O
11       CL-           0.000000      1.000 CL-
12       NA+           0.000000      1.000 NA+
13       K+            0.000000      1.000 K+
14       GLUCOS        0.000000      1.000 GLUCOS
15       LACTIC        0.000000      1.000 LACTIC
16       HCO3-        18.055600      1.000 CO2      1.000 H2O      -1.000 H+
17       H2CO3         6.566000      1.000 CO2      1.000 H2O
18       CO3=         45.661600      1.000 CO2      1.000 H2O      -2.000 H+
19       MISC        -20.128000      1.000 MISC-    2.000 H+
20       MISC-         0.000000      1.000 MISC-
END   OF MATRIX IN STORAGE
```

SPECIAL DATA CONTROL CARDS

In routine calculation with this program, there are several parameters and physical constants required for subsidiary computation, setting tolerances, and organizing output. These parameters each have a nominal value set by either the subroutine START or the Control card CLEAR. The nominal values are those most frequently used (37°C, 1.0 atm pressure), but they may be altered for special cases in two ways: by FORTRAN redefinition in the MAIN routine, or by use of the Special Data Control Cards. Each of the following data Control cards is followed immediately by one and only one data card punched with values of the parameters to be used. If the data Control card is not used, the parameter takes its nominal value. Since there is but one data card with each Control, an END card is not required with these data lists.

1. CYCLE

   a) FORTRAN equivalent: NCYCLE = AA (integer, no decimal point).

   b) The number of compartments to be printed per page of output is punched in columns 1-4 (right justified, no decimal point), nominally 8.

2. LIMIT

   a) FORTRAN equivalent: ITMAX = AA (an integer).

   b) The maximum number of iterations per solution is punched in columns 1-4 (right justified, no decimal point) of the data card, nominally 40.

3. LITER

   a) FORTRAN equivalent: ALITER = AA.

   b) The value of moles per liter of water at $T^{o}C$ used in conversion of scales from molar to mole fraction, and in the pH calculation.

   c) The value, punched in columns 1-12 of the data card, must have a decimal point. Nominal value is 55.139673 for $37^{o}C$, 1.0 atm pressure.

4. MULTIPLIERS

   a) FORTRAN equivalent:
   ```
   BMULT(N) = AA
   CALL ROWS(-1).
   ```

   b) This Control card sets the multipliers BMULT(J) for computing the values of the constraint equations (see ROWS, pp. 31-32 above), and evaluates:

$$B(I) = \sum_{J=1}^{5} BBB(I,J) * BMULT(J) \ .$$

c) In each run, at least one BMULT(J) must be non-zero. BMULT(1) is set to 1.0 by subroutine START and by CLEAR; but if a MULTIPLIERS Control card is used, BMULT(1) is reset to the punched value (a blank field would be zero).

d) The data format on the data card is simply 12 columns per number in sequence; each number punched must have a decimal point.

e) RELAXB and SCALEC will not work properly if BMULT(1) = 0.0.

f) Example: The following shows the ROWS data in memory before and after use of the MULTIPLIERS Control card.

```
PRINTROWS

RCW NAME                    B                    B1                   B2                   B3
                                        MULT.= 1.0CCOOCOE CO    0.                   1.0000000E 03

  1  02              1.31500000E 02       0.                   2.0990000E-01        1.3150000E-01
  2  C02             5.2630000E 01        C.                   3.0000000E-04        5.2630000E-02
  3  N2              7.5400CC0E 02        C.                   7.8980000E-01        7.5400000E-01
  4  H20             1.1619967E 02        5.5139670E 01        0.                   6.1060000E-02
  5  H+              1.0000000E-03        1.0CCOCOOE-03        0.                   0.
  6  CL-             1.40C00C0E-01        1.4CC00C0E-01        0.                   0.
  7  NA+             1.30000C0E-01        1.30C00000E-01       0.                   0.
  8  K+              2.0000000E-02        2.0000000E-02        0.                   0.
  9  GLUCOSE         1.0000000E-02        1.00C0CC0E-02        0.                   0.
 10  LACTIC-         1.0000000E-02        1.0000000E-02        0.                   0.
 11  MISC-           1.0000000E-03        1.0CC00000E-03       0.                   0.
END   OF ROWS IN STORAGE

MULTIPLIERS
 MUL(1)= 1.0000E 00  MUL(2)= 1.0000E 03  MUL(3)= 1.0000E 03  MUL(4)=-0.          MUL(5)=

PRINTROWS

RCW NAME                    B                    B1                   B2                   B3
                                        MULT.= 1.0000000E 00    1.0000000E 03        1.0000000E 03

  1  02              3.4140000E 02        0.                   2.0990000E-01        1.3150000E-01
  2  C02             5.293C000E 01        0.                   3.0000000E-04        5.2630000E-02
  3  N2              1.5438000E 03        C.                   7.8980000E-01        7.5400000E-01
  4  H20             1.1619967E 02        5.5139670E 01        0.                   6.1060000E-02
  5  H+              1.CC00CC0E-03        1.CC000C0E-03        0.                   0.
  6  CL-             1.4C000000E-01       1.40C00000E-01       0.                   0.
  7  NA+             1.30000000E-01       1.30000000E-01       0.                   0.
  8  K+              2.000CCC0E-02        2.CCCOC00E-02        0.                   0.
  9  GLUCOSE         1.00000000E-02       1.00000C0E-02        0.                   0.
 10  LACTIC-         1.00000000E-02       1.00C00000E-02       0.                   0.
 11  MISC-           1.00000C0E-03        1.0C000C0E-03        0.                   0.
END   OF ROWS IN STORAGE
```

5. RT

a) FORTRAN equivalent: RT = AA (floating point number.

b) R is the universal gas constant; T the absolute temperature, nominally 616.27403.

c) This constant for computing the free energy of the system is used by OUTPUT and PRINTPIE; and, of course, may be used in the MAIN routine.

d) Punched in columns 1-12 of the data card, with decimal point.

6. TOLERANCES

a) FORTRAN equivalent: TOL(N) = AA.

b) TOL(1) through TOL(5) are tolerances used by the subroutine SOLVE (as explained in Ref. 3).

VERB CONTROL CARDS

The verb Control cards are macroinstructions for treating data already in computer memory. Each such instruction is punched in the first 12 columns of a card. The first six columns are unique for each instruction. Frequently, a symbol or a number is punched in columns 7-12, beginning always in column 7; the symbol or number is then interpreted as an argument or subscript for the instruction as required. Whereas data Control cards are always followed by data cards, verb Control cards are complete in themselves. At the conclusion of execution, control is transferred to the next Control card or to the MAIN routine as appropriate.

1. TITLE (any six letters not blank, and not a Control card)

Reproduces the BCD information in columns 1-72 on the output sheet. The word TITLE is superfluous, any card which is not blank in the first six columns or another Control card will be treated as though it were a TITLE card.

Two TITLE cards may not be used in succession. If two are required, they may be separated by an END card, which does nothing, or by a COMMENT Control card.

The last TITLE card read is used for the heading of each output sheet.

2. .....(COMMENT).

If the first six columns of a Control card are blank, the only result is that the BCD information in columns 7-72 is reproduced on the output sheet.

3. SOLVE

Causes the problem to be solved to completion following the method of Clasen [3]. No initial, feasible VECTORX is necessary. Of course, the problem as it exists in the computer may not have a feasible solution, and in these cases messages are printed appropriately (see Examples, Chap. III below). The result of SOLVE for a feasible problem is the following list of messages both with and without the use of VECTORX:

```
SOLVE

SIMPLEX  0.    1 ITERATIONS. MAX MIN ELEMENT= 5.0000000 E-04. CONDITION   0
SIMPLEX  1.    2 ITERATIONS  FR ENG=-1.1109346 E 04
SIMPLEX  2.    0 ITERATIONS  FR ENG=-1.1109346 E 04
ITERATION    1  CHANGE IN FREE ENERGY=-6.1035156 E-04  STEP SIZE= 2.6146968 E-01 AV THETA= 4.54114E 00
ITERATION    2  CHANGE IN FREE ENERGY=-6.1035156 E-04  STEP SIZE= 1.8970017 E-01 AV THETA= 3.94945E 00
ITERATION    3  CHANGE IN FREE ENERGY=-6.1035156 E-03  STEP SIZE= 7.3694358 E-01 AV THETA= 3.18247E 00
ITERATION    4  CHANGE IN FREE ENERGY=-3.6254883 E-02  STEP SIZE= 1.0000000 E 00 AV THETA= 2.34344E 00
ITERATION    5  CHANGE IN FREE ENERGY=-2.8210450 E-01  STEP SIZE= 1.0000000 E 00 AV THETA= 1.68408E 00
ITERATION    6  CHANGE IN FREE ENERGY=-1.6680908 E 00  STEP SIZE= 1.0000000 E 00 AV THETA= 1.16728E 00
ITERATION    7  CHANGE IN FREE ENERGY=-6.6837158 E 00  STEP SIZE= 1.0000000 E 00 AV THETA= 7.55397E-01
ITERATION    8  CHANGE IN FREE ENERGY=-1.6467774 E 01  STEP SIZE= 1.0000000 E 00 AV THETA= 4.42403E-01
ITERATION    9  CHANGE IN FREE ENERGY=-2.1269043 E 01  STEP SIZE= 1.0000000 E 00 AV THETA= 2.57832E-01
ITERATION   10  CHANGE IN FREE ENERGY=-1.1234253 E 01  STEP SIZE= 1.0000000 E 00 AV THETA= 1.32562E-01
ITERATION   11  CHANGE IN FREE ENERGY=-1.9671387 E 00  STEP SIZE= 1.0000000 E 00 AV THETA= 2.15730E-02
ITERATION   12  CHANGE IN FREE ENERGY=-2.3681641 E-02  STEP SIZE= 1.0000000 E 00 AV THETA= 3.36555E-04
ITERATION   13 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   14 MAX CHANGE IN PIE= 4.6876813 E-04 MAX ROW ERROR=-2.4306297 E-02
ITERATION   15 MAX CHANGE IN PIE= 2.6914882 E-07 MAX ROW ERROR=-7.6293945 E-06

OUTPUT



SOLVE
PROJECTION  1  SIZE        0.00.  SCALE        1.00
ITERATION   1 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   2 MAX CHANGE IN PIE= 9.8790495 E-06 MAX ROW ERROR=-7.6293945 E-05

      NOTE FASTER SOLVE WITH VALID VECTORX
```

4. OUTPUT

Causes the current values for the species in moles and mole-fractions to be printed along with appropriate messages. The number of compartments per page of output may be set by NCYCLE. A third line of output--say, the moles per liter of water for each species--may be printed by RETURNing to the MAIN routine, computing the new values of the vector x, storing the result in X1, and setting IV(23)=1. If IV(23)=1, the vector X1 will be printed as a third line of output (see Chap. III below). The subroutine OUTPUT also calls subroutines (see Chap. IV below) which compute $\bar{x}$ for every compartment and pH for the compartments containing a species named (exactly) H+.... .

If the current solution is infeasible or non-optimal (IERROR $\neq$ 1 or IOPT = 0), appropriate messages are printed. (See discussion in Chap. III below.)

Table IV shows an example of normal output for Soda Pop.

5. RETURN

Transfers control to the MAIN routine. Thus, after RETURN, the next instruction executed by the program is: the first valid FORTRAN statement after the last used CALL INPUT statement in the MAIN routine (see Chap. III below).

6. DELETE

Removes the last row of the matrix while adding the row times the Π value for this row to the free-energy-parameter vector:

$$C(J) = C(J) + A(I,J) * \Pi(I) \quad , \quad \text{for all } J \ ,$$

where I is the number of the row deleted. That is, the entire last constraint equation is deleted, but the c_j values for any columns (species) thus affected are incremented by precisely the free-energy equivalent of that constraint. Therefore, SOLVE used before and after DELETE will give identical solutions. (See Chap. III below.) This Control card, along with PRINTMATRIX, can be used to determine an unknown c_j value. (See Example I--Soda Pop, Chap. III below.)

7. PRINTROWS

Prints the current data of the input components, i.e., the current names of the components and their corresponding B(I) and BBB(I,J) values. (For an example, see MULTIPLIERS, pp. 43-44 above.)

Table IV

NORMAL SOLUTION FOR EXPERIMENTAL SODA-POP DECK

EXPERIMENTAL SODA POP DECK 5-1-67

RMS MASS BALANCE ERROR= 4.726E-08 MAX. ERROR= 9.060E-08 ON ROW CO2
RMS EQUILIBRIUM ERROR= 1.163E-07 MAX. ERROR= 2.616E-07 IN CO3= OF LIQUID PHASE

OPTIMAL SOLUTION
OBJECTIVE= -1.1168583 E 04 RT * OBJECTIVE= -6.8829077 E 06

		GAS PHASE	LIQUID PHASE
X-BAR		9.99151E 02	5.54895E 01
PH		0.	4.47707E 00
O2	MOLES	1.31500E 02	1.29516E-04
	MFRAC	1.31612E-01	2.33406E-06
CO2	MOLES	5.26287E 01	1.27070E-03
	MFRAC	5.26734E-02	2.28998E-05
N2	MOLES	7.54000E 02	4.15794E-04
	MFRAC	7.54640E-01	7.49320E-06
H2O	MOLES	6.10230E 01	5.51766E 01
	MFRAC	6.10749E-02	9.94361E-01
H+	MOLES	-0.	3.33229E-05
	MFRAC	-0.	6.00527E-07
OH-	MOLES	-0.	7.18394E-10
	MFRAC	-0.	1.29465E-11
CL-	MOLES	-0.	1.40000E-01
	MFRAC	-0.	2.52300E-03
NA+	MOLES	-0.	1.30000E-01
	MFRAC	-0.	2.34279E-03
K+	MOLES	-0.	2.00000E-02
	MFRAC	-0.	3.60429E-04
GLUCOS	MOLES	-0.	1.00000E-02
	MFRAC	-0.	1.80214E-04
LACTIC	MOLES	-0.	1.00000E-02
	MFRAC	-0.	1.80214E-04
HCO3-	MOLES	-0.	3.03114E-05
	MFRAC	-0.	5.46254E-07
H2CO3	MOLES	-0.	1.77831E-06
	MFRAC	-0.	3.20477E-08
CO3=	MOLES	-0.	5.17534E-11
	MFRAC	-0.	9.32671E-13
MISC	MOLES	-0.	9.96989E-04
	MFRAC	-0.	1.79672E-05
MISC-	MOLES	-0.	3.01078E-06
	MFRAC	-0.	5.42585E-08

8. PUNCHROWS

Punches out current values of rows and multipliers in the ROWS and MULTIPLIERS format.

9. PRINTMATRIX

Prints the matrix and free-energy parameters in the input format of MATRIX. (See examples under ALTERA and ALTERC, pp. 40-41 above.)

10. PUNCHMATRIX

Punches out the matrix currently in storage.

11. PRINTTABLEAU

Prints the matrix in tableau form; e.g., for Soda Pop:

MATRIX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	0	0
3	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	1	1	0	0	0	0	0	1	1	1	0	0
5	0	0	0	0	0	0	0	1	-1	0	0	0	0	0	0	-1	0	-2	1	0
6	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

The rows and columns of the matrix correspond to constraints and species, respectively, in the model.

12. PUNCHX

Punches out present values of the vector x in accordance with format for VECTORX. These punchouts may then be used as the input VECTORX in a subsequent run.

13. PRINTPIE

Prints out the current values of PIE (as explained in Ref. 3):

	PIE	PIE*RT	B
O2	-12.96790016	-7991.780090	1.3150000E 02
CO2	-10.68438447	-6584.508667	5.2630000E 01
N2	-11.80151427	-7272.966736	7.5400000E 02
H2O	-0.00565448	-3.484709	1.1619967E 02
H+	-14.32545722	-8828.407227	1.0000000E-03
CL-	-5.98230606	-3686.739868	1.4000000E-01
NA+	-6.05641407	-3732.410706	1.3000000E-01
K+	-7.92821622	-4885.953735	2.0000000E-02
GLUCOSE	-8.62136340	-5313.122375	1.0000000E-02
LACTIC-	-8.62136340	-5313.122375	1.0000000E-02
MISC-	-16.72950649	-10309.960327	1.0000000E-03
PRINTPIE			

14. SIMPLEAA

Provides a preliminary solution for the problem by using a linear programming algorithm. It provides the model with initial guesses (initial, but non-optimal, feasible solutions) which are used with the SOLVE Control card. SOLVE calls SIMPLE if it is required. SIMPLE also tests the problem for feasibility. This Control card has an alphanumeric symbol--punched in columns 7-12--which may be any of 6 alphanumeric characters, including blanks. If the problem is feasible, this symbol is ignored. If it is infeasible, the program executes a GOTO instruction on this symbol (described below, p. 52). If the symbol in columns 7-12 is X-----, control skips to the next CLEAR or EXIT card when the problem is infeasible.

15. MESSAGES

Prints a one-line message for each iteration when solving.

16. ALLMESSAGES

Prints all possible messages.

17. NOMESSAGES

Suppresses messages on inputting and solving.

18. GOTO AA

Has a six-character BCD word punched in columns 7-12 which causes the computer to space forward along the Control cards until it reaches a SYMBOL Control card with the same punches in columns 7-12. If the punches in columns 7-12 in this card are X-----, the program will stop spacing if it reaches a Control card bearing CLEAR or EXIT first. An example of this Control card is: GOTO--ROSE.

19. IFGOTOAA

Has a symbol punched in columns 7-12. If the solution to the latest problem solved was optimal, this Control card is ignored. If the solution was not optimal, a GOTO is executed on the symbol in columns 7-12.

20. SYMBOL

Has a six-character alphanumeric word punched in columns 7-12. This is a dummy Control card used by the SIMPLE, GOTO, and IFGOTO Control cards. An example of such a Control card is: SYMBOLROSE.

21. RELAXBN

Where N = 2, 3, 4, or 5, this Control card replaces BBB(I,1) by: BBB(I,1) + (BMULT(N)/BMULT (1)) * BBB(I,N), for all I, and then sets BMULT(N) = 0.0. BBB(I,N) is not changed. Example (cf. ROWS, pp. 31-32 above):

```
RELAX83

PRINTROWS

ROW NAME                    B                  B1                 B2                 B3
                                        MULT.= 1.0000000E 00  -0.                0.

  1  02              1.3150000E 02      1.3150000E 02      2.0990000E-01      1.3150000E-01
  2  CO2             5.7892999E 01      5.7892999E 01      3.0000000E-04      5.7892999E-02
  3  N2              7.5400000E 02      7.5400000E 02      7.8980000E-01      7.5400000E-01
  4  H2O             1.1606000E 02      1.1606000E 02      0.                 6.1060000E-02
  5  H+              1.0000000E 03      1.0000000E 03      0.                 0.
  6  CL-             1.4500000E-01      1.4500000E-01      0.                 0.
  7  NA+             1.3500000E-01      1.3500000E-01      0.                 0.
  8  K+          -0.                    0.                 0.                 0.
  9  GLUCOSE     -0.                    0.                 0.                 0.
 10  LACTIC-     -0.                    0.                 0.                 0.
 11  MISC-       -0.                    0.                 0.                 0.
END   OF ROWS IN STORAGE

CLEAR
```

22. JACOB

Prints an array of partial derivatives of the output mole numbers with respect to the input components. In the example (Table V), if an input component (shown as column headings) were incremented positively by one mole, the output species (shown down the left side) response would be the corresponding number in the array in moles. Note that the first two entries in each compartment are $\bar{x}$ and pH, whose responses are also shown. These partial derivatives are computed, of course, from the current values of x in storage. Table V is the result of the JACOB Control card applied to the example Soda Pop.

23. MINIJACOB

Prints the partial derivatives of the total number of moles in a compartment, $\bar{x}$, with respect to the moles of input components, and the partial of pH with respect to the input components. Example:

Table V

PARTIAL DERIVATIVES COMPUTED FROM THE SUBROUTINE JACOB FOR THE MODEL "SODA-POP"

JACOB

B JACOBIAN, CYCLE 1

	O2	CO2	N2	H2O	H+	CL-	NA+	K+
GAS PHASE	1.06463E 00	1.06457E 00	1.06463E 00	6.52342E-03	4.34637E-01	-1.16370E 00	-1.16370E 00	-1.16370E 00
PH								
O2	9.99999E-01	2.88950E-07	2.88950E-07	-2.31734E-06	-1.15867E-06	-5.20110E-06	-5.20110E-06	-5.20110E-06
CO2	2.89184E-06	9.99978E-01	2.89184E-06	-2.33423E-05	4.54852E-01	-5.20248E-05	-5.20248E-05	-5.20248E-05
N2	9.43604E-07	9.43604E-07	1.00000E 00	-7.43911E-06	-3.71681E-06	-1.66996E-05	-1.66996E-05	-1.66996E-05
H2O	6.46224E-02	6.45932E-02	6.46229E-02	6.55651E-03	-2.02101E-02	-1.16362E 00	-1.16362E 00	-1.16362E 00
LIQUID PHASE	-6.46255E-02	-6.45729E-02	-6.46265E-02	9.93477E-01	5.20305E-01	2.16370E 00	2.16370E 00	2.16370E 00
PH	1.88925E-04	-3.56454E-03	1.88923E-04	3.32754E-04	-6.51733E 03	4.07222E-03	4.07222E-03	4.07222E-03
O2	6.96069E-07	-2.88713E-07	-2.88845E-07	2.31799E-06	1.15808E-06	5.20104E-06	5.20104E-06	5.20104E-06
CO2	-2.83380E-06	2.13114E-05	-2.83383E-06	2.27415E-05	2.23443E-05	5.10268E-05	5.10268E-05	5.10268E-05
N2	-9.27292E-07	-9.26876E-07	-3.75849E-07	7.44160E-06	3.71788E-06	1.66973E-05	1.66973E-05	1.66973E-05
H+	-5.33054E-08	2.34726E-07	-5.33058E-08	5.71078E-07	5.00068E-01	9.86901E-07	9.86901E-07	9.86901E-07
OH-	-5.28864E-13	-6.73732E-12	-5.28874E-13	1.34849E-11	-1.07807E-05	2.18863E-11	2.18863E-11	2.18863E-11
H2O	-6.46223E-02	-6.45935E-02	-6.46229E-02	9.93443E-01	4.75095E-01	1.16362E 00	1.16362E 00	1.16362E 00
CL-	0.	0.	0.	-0.		1.00000E 00	-0.	-0.
NA+	0.	0.	0.	-0.	-0.	-0.	1.CCCOOE 00	-0.
K+	0.	0.	0.	-0.	-0.	-0.	-0.	1.00000E 00
GLUCOS	0.	0.	0.	-0.	-0.	-0.	-0.	-0.
LACTIC	0.	0.	0.	-0.	-0.	-0.	-0.	-0.
HCO3-	-5.46105E-08	2.59369E-07	-5.46109E-08	5.68762E-07	-4.54873E-01	9.58729E-07	9.58729E-07	9.58729E-07
H2CO3	-3.97749E-09	2.98125E-08	-3.97751E-09	3.20057E-08	2.99078E-08	3.95723E-08	3.95723E-08	3.95723E-08
CO3=	-7.07280E-14	1.80709E-14	-7.07290E-14	1.01075E-12	-1.55329E-06	2.12220E-12	2.12220E-12	2.12220E-12
MISC	-1.30577E-09	2.46367E-08	-1.30577E-09	-2.29988E-09	4.50455E-02	-2.81462E-08	-2.81462E-08	-2.81462E-08
MISC-	1.30579E-09	-2.46370E-08	1.30578E-09	2.29989E-09	-4.50458E-02	2.81460E-08	2.81460E-08	2.81460E-08

B JACOBIAN, CYCLE 2

	GLUCOSE	LACTIC-	MISC-
GAS PHASE	-1.16370E 00	-1.16370E 00	-1.59702E 00
PH			
O2	-5.20110E-06	-5.20110E-06	-4.04530E-06
CO2	-5.20248E-05	-5.20248E-05	-4.53535E-01
N2	-1.66996E-05	-1.66996E-05	-1.30020E-05
H2O	-1.16362E 00	-1.16362E 00	-1.14347E 00
LIQUID PHASE	2.16370E 00	2.16370E 00	1.64496E 00
PH	4.07222E-03	4.07222E-03	6.49771E 03
O2	5.20104E-06	5.20104E-06	4.04644E-06
CO2	5.10268E-05	5.10268E-05	2.87498E-05
N2	1.66973E-05	1.66973E-05	1.29906E-05
H+	9.86901E-07	9.86901E-07	-4.98561E-01
OH-	2.18863E-11	2.18863E-11	1.07483E-05
H2O	1.16362E 00	1.16362E 00	6.89956E-01
CL-	-0.	-0.	3.25963E-11
NA+	-0.	-0.	3.02680E-11
K+	-0.	-0.	4.65661E-12
GLUCOS	1.CCCOOE 00	-0.	2.32831E-12
LACTIC	-0.	1.CCOOOE 00	2.32831E-12
HCO3-	9.58729E-07	9.58729E-07	4.53504E-01
H2CO3	3.95723E-08	3.95723E-08	9.75450E-09
CO3=	2.12220E-12	2.12220E-12	1.54862E-06
MISC	-2.81462E-08	-2.81462E-08	9.52079E-01
MISC-	2.81460E-08	2.81460E-08	4.79209E-02

```
PINIJACOB
 P JACOBIAN, CYCLE  1
                O2            CO2           N2            H2O           H+            CL-           NA+           K+
GAS PHASE      1.06463E 00  1.06457E 00  1.06463E 00  6.52342E-03  4.34637E-01 -1.16370E 00 -1.16370E 00 -1.16370E 00
 PH
LIQUID PHASE -6.46255E-02 -6.45729E-02 -6.46265E-02  9.93477E-01  5.20305E-01  2.16370E 00  2.16370E 00  2.16370E 00
 PH            1.88925E-04 -3.56454E-03  1.88923E-04  3.32754E-04 -6.51733E 03  4.07222E-03  4.07222E-03  4.07222E-03
 P JACOBIAN, CYCLE  2
                GLUCOSE       LACTIC-       MISC-
GAS PHASE      -1.16370E 00 -1.16370E 00 -1.59702E 00
 PH
LIQUID PHASE  2.16370E 00  2.16370E 00  1.64496E 00
 PH           4.07222E-03  4.07222E-03  6.49771E 03
```

24. EJECT

Causes the page to be ejected and a new heading to be printed.

25. CLEAR

Zeros out all common storage; it may be used to start a new problem and to erase all information produced by a previous problem. It also sets all parameters to their nominal values (see Special Data Control cards, p. 42 ff. above).

26. EXIT

Terminates the job.

27. END

Does nothing.

Chapter III

EXAMPLES

The following examples have been chosen to illustrate certain principles in the operation of CHEMIST. Because CHEMIST is a single-pass interpretive program and may vary with each pass on the machine, it would not be possible to anticipate all aspects which may, in the construction of subsequent models, cause difficulty or misunderstanding. We can, however, show the solutions to major problems already encountered. It should be noted, too, that the program is continuously evolving. CHEMIST was designed to solve a particular class of problems; but as needs arise, the program grows and the class broadens--that is to say, new problems will inevitably arise.

Although this Memorandum is not intended to elucidate in any detail the theoretical basis for the program, occasionally in the following examples reference to the mathematical bases will be required to justify certain operations. At these junctures, results or more fundamental research are quoted and the proofs referenced.

EXAMPLE I--SODA POP

The first example is a simple system, but far-reaching enough to illustrate most of the basic operations. Soda Pop is a two-phase system consisting of a gas phase, a liquid phase having a carbon dioxide system plus glucose, the sodium and potassium salts of lactic acid, and a miscellaneous anion.

Table VI is a listing of the complete data deck for the Soda-Pop model. Each line of print in the table is equivalent to one card in the deck. The Control cards in this deck are, in order:

```
ROWS
END
MATRIX
END
MULTIPLIERS
VECTORX
END
SOLVE
OUTPUT
EXIT
```

All other cards are data cards. Each Control card is processed in sequence; the first is ROWS. A ROWS card is always followed by data; in this case, the components of the Soda-Pop model.

The names of the components are listed on the left, and mole numbers for each component in the five columns to the right. To determine the total moles of each component input for this model, each column of mole numbers is multiplied by its respective multiplier (after the MULTIPLIERS Control card) and the results are summed. Evidently, the first column of mole numbers represents one liter of water (at 37°C) plus one millimole of H^+, plus 140 millimoles of Na^+, etc.; i.e., one liter of solution of ionic strength 0.151 moles per liter.

The second column of mole numbers represents dry fresh air--the sum of the mole numbers is 1.0: 20.99 percent is O_2, 0.03 percent is CO_2, and the rest N_2. However, in this model, the second multiplier (MULT(2)) is zero; so no fresh air is to be used. The third column of

Table VI

LISTING OF THE COMPLETE DATA DECK

FOR THE SODA-POP MODEL

MATRIX

GAS PHASE						
1	O2	-10.940000	1.000 O2			
2	CO2	-7.740740	1.000 CO2			
3	N2	-11.520000	1.000 N2			
4	H2O	2.790000	1.000 H2O			
LIQUID PHASE						
5	O2	0.000000	1.000 O2			
6	CO2	0.000000	1.000 CO2			
7	N2	0.000000	1.000 N2			
8	H+	0.000000	1.000 H+			
9	OH-	39.390000	1.000 H2O	-1.000 H+		
10	H2O	0.000000	1.000 H2O			
11	CL-	0.000000	1.000 CL-			
12	NA+	0.000000	1.000 NA+			
13	K+	0.000000	1.000 K+			
14	GLUCOS	0.000000	1.000 GLUCOS			
15	LACTIC	0.000000	1.000 LACTIC			
16	HCO3-	18.055600	1.000 CO2	1.000 H2O	-1.000 H+	
17	H2CO3	6.566000	1.000 CO2	1.000 H2O		
18	CO3=	45.661600	1.000 CO2	1.000 H2O	-2.000 H+	
19	MISC	-20.128000	1.000 MISC-	1.000 H+		
20	MISC-	0.000000	1.000 MISC-			

mole numbers is evidently saturated gas (37°C, one atmosphere); and the mole fraction of O_2 in this gas is 0.1315, or 0.1315 x 760 = 100 mm Hg. The pCO_2 is 40 mm Hg, i.e., this gas mixture is that found in the alveoli of normal, resting human males.

The MULT(3) is 1000., so that this model will use 1000 moles of the gas mixture to equilibrate with one liter (MULT(1) = 1.0) of solution.

After the Control card MATRIX, the data cards show two compartments or phases, gas and liquid, and a list of species expected in each compartment. The second column (in the data cards and in the output) contains the respective free-energy parameters for each species. The subsequent data columns under MATRIX show the lists of components (and stoichiometric coefficients) out of which each species is formed, essentially a list of chemical equations.

The free-energy parameters are computed following Eq. (14) (p. 16 above). Consider

$$OH^- \rightleftarrows 1.0\ H_2O - 1.0\ H^+ , \quad 39.39 .$$

We have, by convention, the mass action equation (37°C)

$$(H^+)(OH^-) = 2.3775 \times 10^{-14} .$$

Converting to mole-fraction scale by dividing each concentration by moles per liter of solvent (water), 55.13967, and taking the natural log, we get,

$$\ln \frac{(H^+)(OH^-)}{(ALITER)^2} = -39.39 .$$

Again, for bicarbonate the reaction in the model is

$$HCO_3^- \rightleftarrows CO_2 + H_2O - H^+ , \quad 18.0556 .$$

This c_j is obtained as follows: From the mass-action equation, with $(H_2O) = 1.0$ by convention,

$$\frac{(HCO_3^-)(H^+)}{(H_2O)(CO_2)} = 1 \times 10^{-6.01} ,$$

and

$$\ln \frac{10^{-6.01}}{ALITER} = -18.0556 .$$

Here, the concentration of water is already expressed, by convention, in mole-fraction scale. Anticipating a subsequent example, a model of blood, we use the apparent first ionization constant for carbonic acid in plasma, pK = 6.01 at 37°C. In the theoretical case of pure water, De Haven [10] shows a model of the ideal carbonate system using the data from Edsall and Wyman.† De Haven's paper also derives the free-energy parameters for CHEMIST in more detail.

Conversely, we can compute the pK of the miscellaneous anion MISC- from the implied mass-action equation shown in the matrix. Evidently, we have

$$\ln \frac{[MISC^-][H^+]}{[MISC]} = -20.1280 ,$$

---

†See Ref. 18, Chap. 10.

where square brackets indicate concentration on the mole fraction scale, or

$$\frac{[\text{MISC-}][H^+]}{[\text{MISC}]} = \frac{(\text{MISC-})(H^+)}{(\text{MISC})(\text{ALITER})} = \exp(-20.1280) \ .$$

Multiplying by ALITER = 55.1397 gives

$$\frac{(\text{MISC-})(H^+)}{(\text{MISC})} = \exp(-16.11813) = 10^{-7.0} \ ,$$

or the pK = 7.0 for this reaction at 37°C.

The solubility coefficient for a gas in a liquid phase <u>at one atmosphere</u> is usually defined as:

$$a = \text{ml gas at STP/ml liquid at } T^\circ C \ ,$$

where STP is 0°C and one atmosphere. However, for this mathematical model, we need a on the mole-fraction scale [17].

$$a' = a \frac{\text{ml}}{\text{ml}} \times \frac{\text{moles of gas}}{\text{ml at STP}} \Big/ \frac{\text{moles } H_2O}{\text{ml at } T^\circ C} \ .$$

For 37°C

$$a' = a \times \frac{1}{22{,}400} \Big/ 0.05513967$$

$$= a \times 8.0963 \times 10^{-4} \ .$$

For example, for O_2 in pure water at 37°C, a = 0.02386 and a' = 1.9318×10^{-5}. For O_2 in plasma, a = 0.0214 and a' = 1.7326×10^{-5} in the mole-fraction scale.

In the model, for the chemical "reaction"

$$O_{2gas} \rightleftarrows O_{2liquid}$$

at 37°C in simulated plasma, we have,

$$\frac{[O_2]_{gas}}{[O_2]_{liquid}} = 1.7326 \times 10^{-5} = \exp(c_1) \ ;$$

so

$$c_1 = \ln 1.7326 \times 10^{-5} = -10.940 \ .$$

Similar computations are made for the other gases.

The data cards listed after the Control card VECTORX (which may also have the title of the deck punched in the remaining columns) are an initial guess for the solution supplied either by the programmer or by the PUNCHX Control card from a previous pass of the same data deck. The purpose of the initial guess is to save computation time, but it is not required. If the data cards of VECTORX are not supplied with the data deck, or if any of the names given in the data cards of VECTORX do not compare precisely with those in MATRIX (compartment names as well as species names), or if the initial guess is a very poor one, the SOLVE subroutine: 1) obtains initial starting value by PROJECTION into the feasible solution space [3]; 2) solves a linear programming problem (obtained by dropping the log terms out of the free-energy function) using the SIMPLEX subroutine; and 3) iterates the full non-linear problem by two different methods to obtain the optimal solution. If the VECTORX supplied is a feasible solution, or if a valid

solution exists in storage (from a previous SOLVE), SOLVE first calls PROJECTION, which is used to obtain an initial solution when the solution expected next differs from the previous in only relatively minor ways [3].

The messages shown with the Control card SOLVE (p. 46 above), were printed out for the Soda-Pop model after VECTORX and the related data cards were omitted. These messages are printed by the SOLVE subroutine. The SOLVE subroutine also sets the flag IERROR = 1 if the problem is feasible and the solution has converged; IERROR $\neq$ 1 if otherwise. If the problem is not feasible, or if an optimal solution (within the TOLERANCES) has not been found by the program, other messages will appear. For example, if the matrix inversion subroutine finds a singular matrix, this is reported. If the SIMPLEX subroutine finds an infeasible linear problem, the combination of rows giving a dependent set is listed. If ITMAX is exceeded, that message is given and for the next pass ITMAX should be increased depending upon the list of messages.

Examining the list of messages with the Control card SOLVE (p. 46 above), one may note that both the free-energy function and the maximum mass-conservation error are decreasing steadily. These values may be oscillatory or stationary at values outside the TOLERANCES [3]. In such cases, the solution is non-optimal and the problem being solved must be reviewed.

The result of the OUTPUT Control card is shown in Table XII. First, the last read TITLE Control card is reproduced. Next, two numerical error messages are printed. These are almost self-explanatory. The errors in the conservation of mass (and other constraint) equations are

Table VII

NORMAL OUTPUT OF EXPERIMENTAL SODA-POP DECK

```
EXPERIMENTAL SODA POP DECK      5-1-67


RMS MASS BALANCE ERROR= 4.726E-08  MAX. ERROR= 9.060E-08 ON ROW CO2
RMS  EQUILIBRIUM ERROR= 1.163E-07  MAX. ERROR= 2.616E-07 IN CO3=   OF LIQUID PHASE

OPTIMAL SOLUTION
OBJECTIVE=  -1.1168583 E 04  RT * OBJECTIVE=  -6.8829077 E 06

                GAS PHASE     LIQUID PHASE

X-BAR           9.99151E 02   5.54895E 01

PH              0.            4.47707E 00

O2      MOLES   1.31500E 02   1.29516E-04
        MFRAC   1.31612E-01   2.33406E-06

CO2     MOLES   5.26287E 01   1.27070E-03
        MFRAC   5.26734E-02   2.28998E-05

N2      MOLES   7.54000E 02   4.15794E-04
        MFRAC   7.54640E-01   7.49320E-06

H2O     MOLES   6.10230E 01   5.51766E 01
        MFRAC   6.10749E-02   9.94361E-01

H+      MOLES  -0.            3.33229E-05
        MFRAC  -0.            6.00527E-07

OH-     MOLES  -0.            7.18394E-10
        MFRAC  -0.            1.29465E-11

CL-     MOLES  -0.            1.40000E-01
        MFRAC  -0.            2.52300E-03

NA+     MOLES  -0.            1.30000E-01
        MFRAC  -0.            2.34279E-03

K+      MOLES  -0.            2.00000E-02
        MFRAC  -0.            3.60429E-04

GLUCOS  MOLES  -0.            1.00000E-02
        MFRAC  -0.            1.80214E-04

LACTIC  MOLES  -0.            1.00000E-02
        MFRAC  -0.            1.80214E-04

HCO3-   MOLES  -0.            3.03114E-05
        MFRAC  -0.            5.46254E-07

H2CO3   MOLES  -0.            1.77831E-06
        MFRAC  -0.            3.20477E-08

CO3=    MOLES  -0.            5.17534E-11
        MFRAC  -0.            9.32671E-13

MISC    MOLES  -0.            9.96989E-04
        MFRAC  -0.            1.79672E-05

MISC-   MOLES  -0.            3.01078E-06
        MFRAC  -0.            5.42585E-08
```

averaged and the RMS error (one-sigma) is computed and printed, along with the constraint having maximum absolute error. The RMS error for all mass-action equations in the model is then computed and printed, along with the maximum absolute error for any species.

Next, the OUTPUT subroutine tests the flag IOPT (set by the subroutine ARITH) and prints the appropriate message (in this case, "Optimal Solution"). If IOPT = 0, "Not Optimal Solution" is printed along with the values of θ_j [2] for each species. The θ_j are, roughly, a measure of the degree to which a constraint is violated--by scanning the θ_j printed out, the analyst can see where the trouble is likely to lie: for an optimal solution, the θ_j are greater than -1.0; the more negative the θ_j less than -1.0, the more difficulty the program is experiencing in satisfying the constraint.

Next, the current value of the objective function is printed by the OUTPUT subroutine. The objective function is

$$G(x) = \sum_{J=1}^{N} x_j (c_j + \ln \hat{x}_j) \ ,$$

where the concentration, $\hat{x}_j$ of X_j is computed within the appropriate compartment. RT times G(x) is the Gibbs' free-energy function, but its interpretation in this context is precarious because of the definitions of c_j. Frequently, the c_j are not the standard free-energy parameters for matching empirical data (see p. 7 above; also, Ref. 17).

Finally, the species are listed by compartment. Usually, as in this case, moles and mole fractions of

each species are printed. (In the next example, we show the result of returning to the MAIN routine to compute moles per liter of water as well.) The first line printed is XBAR (i.e., $\bar{x}$), the sum of the moles of all species in the compartment. The second is the pH of appropriate compartments, i.e., those (and only those) compartments containing a species named precisely H^+ (left adjusted). The list of species follows. The same substance in more than one compartment (e.g., H_2O, CO_2) can be given the same name to improve the appearance of the output, as was done here.

In this simple problem, it is easy to check the conservation of mass equations; e.g., for CO_2:

gas phase	52.6287 moles
liquid phase	0.0012707
HCO_3^-	0.0000303
H_2CO_3	---
$CO_3^=$	---
	52.6300010 moles ,

which is the mole fraction of CO_2 in the gas phase times 1000 moles of gas (BMULT(3) = 1000.).

It is also simple to check the mass-action equations, for example:

$$\frac{[HCO_3^-][H^+]}{[CO_2][H_2O]} = \frac{5.462 \times 10^{-7} \times 6.005 \times 10^{-7}}{2.29 \times 10^{-5} \times 0.9944}$$

$$= 1.4406 \times 10^{-8} ,$$

and

$$\ln 1.4406 \times 10^{-8} = -18.0556 ,$$

as shown in species HCO_3^- in the MATRIX (Table VI).

EXAMPLE II--MAIN ROUTINE AND DELETE

The MAIN routine used for Soda Pop until now is that shown above (p. 21). We will now alter the MAIN routine in order to calculate, in addition to the moles and mole-fractions of the vector x, the moles per liter of water for each species. Also, we will print this result as a third line of output in a subsequent CALL OUTPUT command from the MAIN routine. Table VIII is a listing of the modified MAIN to perform those functions.

At any time subsequent to the SOLVE Control card, the vector x in storage contains current values for the distribution of species. By using the RETURN Control card, control is transferred to the MAIN routine and x is available for computation. We assume RETURN has been used and control is transferred to statement number 12 in Table VIII, the next valid FORTRAN instruction after the last used CALL INPUT statement. First, the vector X1 in storage is cleared since, during SOLVE, that vector is used for temporary storage. Next, we search through each compartment to find the number of species having the name H_2O. The vector KL(K) contains the numbers of the first species in each compartment; therefore, NH_2O is the number of the species H_2O in compartment K.

In statement 27, the moles of x_j in compartment K are converted to moles of x_j per liter of H_2O in compartment K

Table VIII

EXAMPLE OF MAIN ROUTINE FOR COMPUTING THIRD LINE OF OUTPUT

```
6472,SODA,H3790,8,50,100,P                    FORTRAN SOURCE LIST                    07/24/67      PAGE  3
      ISN       SOURCE STATEMENT

        0 $IBFTC NZSC     NODECK
          C                        4-7-67
        1       COMMON AIJ(460), IROW(460), JCOL(460)                                  NZSC0060
          C                                                                            NZSC0070
        2       COMMON  /INPT/KA(12),KB(12),BBB(60,5),PH(25),T(20),BMULT(5)            NZSC0080
          C                                                                            NZSC0090
        3       COMMON /SLVE/IV(30),TOL(20),NR(60,2),B(60),PIE(75),V1(75),V2(75),      NZSC0100
               1        V3(75),V4(75),X(170),XMF(170),XN(170),C(170),X1(170),         NZSC011C
               2        X2(170),X3(170),XBAR(25),NAM(25,2),KL(26), R(75,75),          NZSC0120
               3        JCOMP(170),FE,FE2,ERMB,XEMB,NEMB,ERMA,XEMA,NEMA
          C                                                                            NZSC0140
        4       EQUIVALENCE (IV(1),M),(IV(2),MEMO),(IV(3),NCOMP),(IV(4),N,NTOT),       NZSC0150
               1        (IV(5),NIT),(IV(6),MOT),(IV(7),PF),(IV(8),ITER),              NZSC0160
               2        (IV(9),ITMAX),(IV(10),IERROR),(IV(11),LASTCP),(IV(12),KE),    NZSC0170
               3        (IV(13),MAXM),(IV(14),MAXP),(IV(15),MAXN),(IV(16),MAXMO),     NZSC0180
               4        (IV(17),END),(IV(18),BLANK),(IV(19),H2O),(IV(20),MPLUS),      NZSC0190
               5        (IV(21),NCYCLE),(IV(22),NBSTAR),(IV(23),KPF),                 NZSC0200
               6        (IV(24),NAIJ),(IV(25),MAXAIJ),(IV(26),IOPT)
        5       EQUIVALENCE (TOL(3),XMIN),(TOL(4),XSTART),(TOL(5),BARMIN),            NZSC0220
               1            (TOL(11),ALITER),(TOL(12),RT)                             NZSC0230
          C                                                                            NZSC0240
        6       INTEGER PF                                                           NZSC0250
        7       INTEGER END,BLANK,H2O,MPLUS                                          NZSC0260
          C                                                                            NZSC0270
       10       CALL START
       11       CALL INPUT
          C        COMPUTE MOLES/LITER H2O FOR THIRD LINE OF OUTPUT
       12       DO 1 J=1,N
       13     1  X1(J)=0.
       15       DO 10 K=1,NCOMP
       16        KTA=KL(K)
       17        KTB=KL(K+1)-1
       20        DO 2 J=KTA,KTB
       21         IF (KN(J).EQ.H2O) NH2O=J
       24     2 CONTINUE
       26        DO 3 J=KTA,KTB
       27     3   X1(J) = X(J)*ALITER / X(NH2O)
       31    10 CONTINUE
          C              SET FLAG TO PRINT THIRD LINE
       33       IV(23)=1
       34       CALL OUTPUT
       35       IV(23)=0
       36       CALL INPUT
          C
          C
       37       NBSTAR=5
       40       VALUE=PH(2)+.2
       41       CALL PHSLV(6HLIQUID,VALUE)
       42       CALL OUTPUT
       43       CALL INPUT
       44       CALL EXIT
       45       END
```

(ALITER is the moles of water per liter of water at 37°C). The result is stored in the vector X1. Finally, we set IV(23) = 1 and CALL OUTPUT. With IV(23) = 1, the vector X1 will be printed as a third line of output as shown in Table IX.

As another example, the Soda-Pop model will be used to show how a chemical reaction with an unknown equilibrium constant can be incorporated into the model. If data equivalent to K, the mass-action constant, are known--for example, in the mass-action equation

$$\frac{[MISC^-][H^+]}{[MISC]} = K$$

if K is unknown but the concentrations on the left are known--then K can be computed and the reaction incorporated in the usual way. Frequently, however, the equivalent data are not known until an equilibrium for the total milieu can be computed. For example, in the mass-action equation above $[H^+]$ may be unknown <u>because</u> K is unknown. If, however, either [MISC] or [MISC-] are known, K can still be computed by constraining, say, [MISC] to be the known amount and then solving all of the equations of the milieu simultaneously. In the context of the above problem, this is accomplished in one pass as follows:

First, we add a constraint with ALTERB and ALTERA:

```
ALTERB1
* MISC**          5.0000000E-04
  END    (NEW ROWS ARE *-ED)

ALTERA
LIQUID PHASE        -0.000000   -0.000        -0.000        -0.000        -0.000
      MISC         -20.128000    1.000MISC-    1.000H+       1.000MISC**  -0.000
END                 -0.000000   -0.000        -0.000        -0.000        -0.000
```

Table IX

EXPERIMENTAL SODA-POP WITH THIRD LINE OF OUTPUT

```
EXPERIMENTAL SODA POP DECK      5-1-67

RMS MASS BALANCE ERROR= 5.704E-08  MAX. ERROR= 9.966E-08 ON RON CO2
RMS  EQUILIBRIUM ERROR= 1.132E-07  MAX. ERROR= 2.596E-07 IN OH-     OF LIQUID PHASE

OPTIMAL SOLUTION
OBJECTIVE=  -1.1168003 E 04  RT * OBJECTIVE=  -6.8825503 E 06

                GAS PHASE     LIQUID PHASE

X-BAR           9.99151E 02   4.86171E 01

PH              0.            4.47707E 00

O2      MOLES   1.31500E 02   9.71869E-05
        MFRAC   1.31612E-01   2.33406E-06
       X1       1.18822E 02   1.29429E-04

CO2     MOLES   5.26287E 01   9.53021E-04
        MFRAC   5.26734E-02   2.28998E-05
       X1       4.75546E 01   1.26986E-03

N2      MOLES   7.54000E 02   3.11845E-04
        MFRAC   7.54640E-01   7.49320E-06
       X1       6.81305E 02   4.15914E-04

H2O     MOLES   6.10230E 01   4.13824E 01
        MFRAC   6.10749E-02   9.94361E-01
       X1       5.51397E 01   5.51397E 01

H+      MOLES  -0.            2.49922E-05
        MFRAC  -0.            6.00526E-07
       X1      -0.            3.33007E-05

OH-     MOLES  -0.            5.38795E-10
        MFRAC  -0.            1.29465E-11
       X1      -0.            7.17912E-10

CL-     MOLES  -0.            1.05000E-01
        MFRAC  -0.            2.52300E-03
       X1      -0.            1.39906E-01

NA+     MOLES  -0.            9.75000E-02
        MFRAC  -0.            2.34279E-03
       X1      -0.            1.29913E-01

K+      MOLES  -0.            1.50000E-02
        MFRAC  -0.            3.60429E-04
       X1      -0.            1.99866E-02

GLUCOS  MOLES  -0.            7.50000E-03
        MFRAC  -0.            1.80214E-04
       X1      -0.            9.99331E-03

LACTIC  MOLES  -0.            7.50000E-03
        MFRAC  -0.            1.80214E-04
       X1      -0.            9.99331E-03

HCO3-   MOLES  -0.            2.27335E-05
        MFRAC  -0.            5.46254E-07
       X1      -0.            3.02911E-05

H2CO3   MOLES  -0.            1.33379E-06
        MFRAC  -0.            3.20477E-08
       X1      -0.            1.77712E-06

CO3=    MOLES  -0.            3.89150E-11
        MFRAC  -0.            9.32670E-13
       X1      -0.            5.17187E-11

MISC    MOLES  -0.            7.47742E-04
        MFRAC  -0.            1.79672E-05
       X1      -0.            9.96322E-04

MISC-   MOLES  -0.            2.25800E-06
        MFRAC  -0.            5.42585E-08
       X1      -0.            3.00876E-06

GOTO  OK

SYMBOLOK
```

The ALTERB1 data Control card adds a constraint (row) to the matrix (cf. ROWS, pp. 31-32 above):

```
PRINTROWS
ROW NAME              B                  B1                 B2                 B3                 B4                 B5
                                  MULT.= 1.00000000E 00     0.                 1.00000000E 03     0.                 0.

  1  O2          1.31500000E 02          0.                 2.09900000E-01     1.31500000E-01     0.                 0.
  2  CO2         5.26300000E 01          0.                 3.00000000E-04     5.26300000E-02     0.                 0.
  3  N2          7.54000000E 02          0.                 7.89800000E-01     7.54000000E-01     0.                 0.
  4  H2O         1.16199967E 02          5.91396702E 01     0.                 6.10600000E-02     0.                 0.
  5  H+          1.00000000E-03          1.00000000E-03     0.                 0.                 0.                 1.00000000E 00
  6  CL-         1.40000000E-01          1.40000000E-01     0.                 0.                 0.                 1.00000000E 00
  7  NA+         1.30000000E-01          1.30000000E-01     0.                 0.                 0.                 0.
  8  K+          2.00000000E-02          2.00000000E-02     0.                 0.                 0.                 0.
  9  GLUCOSE     1.00000000E-02          1.00000000E-02     0.                 0.                 0.                 0.
 10  LACTIC-     1.00000000E-02          1.00000000E-02     0.                 0.                 0.                 0.
 11  MISC-       1.00000000E-03          1.00000000E-03     0.                 0.                 0.                 0.
 12  MISC**      5.00000000E-04          5.00000000E-04    -0.                -0.                -0.                -0.
END   OF ROWS IN STORAGE
```

And the ALTERA modifies the MISC column (cf. MATRIX, pp. 32-34 above):

```
PRINTMATRIX

MATRIX IN STORAGE

   GAS PHASE
  1       O2        -10.940000     1.000 O2
  2       CO2        -7.740740     1.000 CO2
  3       N2        -11.520000     1.000 N2
  4       H2O         2.790000     1.000 H2O

   LIQUID PHASE
  5       O2          0.000000     1.000 O2
  6       CO2         0.000000     1.000 CO2
  7       N2          0.000000     1.000 N2
  8       H+          0.000000     1.000 H+
  9       OH-        39.390000     1.000 H2O       -1.000 H+
 10       H2O         0.000000     1.000 H2O
 11       CL-         0.000000     1.000 CL-
 12       NA+         0.000000     1.000 NA+
 13       K+          0.000000     1.000 K+
 14       GLUCOS      0.000000     1.000 GLUCOS
 15       LACTIC      0.000000     1.000 LACTIC
 16       HCO3-      18.055600     1.000 CO2       1.000 H2O   -1.000 H+
 17       H2CO3       6.586000     1.000 CO2       1.000 H2O
 18       CO3=       45.661600     1.000 CO2       1.000 H2O   -2.000 H+
 19       MISC      -20.128000     1.000 MISC-     1.000 H+     1.000 MISC**
 20       MISC-       0.000000     1.000 MISC-
END    OF MATRIX IN STORAGE
```

That is, the species MISC is now constrained to be 5.0 $\times 10^{-4}$ moles in the final equilibrium since one of its components is MISC**. MISC** is a component of the model and must show up in the output--the only place it can go is into the species MISC. We now SOLVE, call OUTPUT:

```
EXPERIMENTAL SODA POP DECK      5-1-67

RMS MASS BALANCE ERROR= 9.750E-08  MAX. ERROR= 2.328E-07 ON ROW MISC
RMS  EQUILIBRIUM ERROR= 1.781E-07  MAX. ERROR= 5.024E-07 IN CO3=   OF LIQUID PHASE

OPTIMAL SOLUTION
OBJECTIVE=  -1.1168580 E 04  RT * OBJECTIVE=  -6.8829057 E 06

                GAS PHASE     LIQUID PHASE

X-BAR            9.99151E 02  5.54904E 01

PH               0.           3.29910E 00

O2      MOLES    1.31500E 02  1.29518E-04
        MFRAC    1.31612E-01  2.33406E-06

CO2     MOLES    5.26287E 01  1.27072E-03
        MFRAC    5.26735E-02  2.28998E-05

N2      MOLES    7.54000E 02  4.15801E-04
        MFRAC    7.54640E-01  7.49321E-06

H2O     MOLES    6.10225E 01  5.51771E 01
        MFRAC    6.10744E-02  9.94354E-01

H+      MOLES -0.             5.02012E-04
        MFRAC -0.             9.04682E-06

OH-     MOLES -0.             4.76874E-11
        MFRAC -0.             8.59380E-13

CL-     MOLES -0.             1.40000E-01
        MFRAC -0.             2.52296E-03

NA+     MOLES -0.             1.30000E-01
        MFRAC -0.             2.34275E-03

K+      MOLES -0.             2.00000E-02
        MFRAC -0.             3.60422E-04

GLUCOS  MOLES -0.             1.00000E-02
        MFRAC -0.             1.80211E-04

LACTIC  MOLES -0.             1.00000E-02
        MFRAC -0.             1.80211E-04

HCO3-   MOLES -0.             2.01209E-06
        MFRAC -0.             3.62601E-08

H2CO3   MOLES -0.             1.77833E-06
        MFRAC -0.             3.20475E-08

CO3=    MOLES -0.             2.28043E-13
        MFRAC -0.             4.10960E-15

MISC    MOLES -0.             5.00000E-04
        MFRAC -0.             9.01056E-06

MISC-   MOLES -0.             5.00000E-04
        MFRAC -0.             9.01056E-06
```

(Note that MISC = 5 x 10^{-4} moles) and then DELETE:

```
DELETE
** ROW **MISC **          ** HAS BEEN DELETED **
```

The DELETE operation (see DELETE, p. 48 above) removes the last row of the matrix (the one just added) and replaces that constraint with

$$C(J) = C(J) + A(M,J)* \Pi(M) \quad , \quad \text{for all } J$$

where M is the number of the row removed, i.e., the constraint is replaced by an equivalent change in the thermodynamic parameters of each species X_j affected. Finally, we PRINTMATRIX to obtain the new C(J) for the MISC reaction:

```
PRINTMATRIX

MATRIX IN STORAGE

   GAS PHASE
 1       O2          -10.940000      1.000 O2
 2       CO2          -7.740740      1.000 CO2
 3       N2          -11.520000      1.000 N2
 4       H2O           2.790000      1.000 H2O

   LIQUID PHASE
 5       O2            0.000000      1.000 O2
 6       CO2           0.000000      1.000 CO2
 7       N2            0.000000      1.000 N2
 8       H+            0.000000      1.000 H+
 9       OH-          39.390000      1.000 H2O      -1.000 H+
10       H2O           0.000000      1.000 H2O
11       CL-           0.000000      1.000 CL-
12       NA+           0.000000      1.000 NA+
13       K+            0.000000      1.000 K+
14       GLUCOS        0.000000      1.000 GLUCOS
15       LACTIC        0.000000      1.000 LACTIC
16       HCO3-        18.055600      1.000 CO2      1.000 H2O      -1.000 H+
17       H2CO3         6.566000      1.000 CO2      1.000 H2O
18       CO3=         45.661600      1.000 CO2      1.000 H2O      -2.000 H+
19       MISC        -11.613097      1.000 MISC-    1.000 H+
20       MISC-         0.000000      1.000 MISC-
END   OF MATRIX IN STORAGE
```

In the MISC row, C(J) = -11.613097; or, exp(-11.6131) = $10^{-3.05}$ converting to pK, pK = +3.05 (cf. 7.0, p. 61 above). If we now SOLVE again, the result will be identical to that with the constraint; the equilibrium constant K which produces 5×10^{-4} moles of MISC per liter of solution has been found.

A final example using the Soda-Pop model is the following method for "goaling" the output. This general method allows one to obtain in a single pass a desired value of a (dependent) variable in the output by adjusting the value of a (independent) variable in the input. Thus, one might adjust the HCl input to give a desired pH, or the amount of hemoglobin to give a desired hematocrit. Frequently, in matching laboratory data, a desired (dependent) variable has been measured but the (independent) variable causes or producing the result has not. Such a case requires an adjustment, in the model construction, of the independent variable to the appropriate level by watching the variation of the dependent variables.

The procedure uses Newton's method of iteration, and is subject to the theoretical limitations of that method as well as the numerical limitation of the computer. It consists, essentially, of: SOLVEing to give the initial value of the dependent variable and the increment required to give the desired result; computing the partial derivative of the dependent with respect to the independent variable; and solving the equation

$$\Delta I \left(\frac{\partial D}{\partial I} \right) = \Delta D \qquad (22)$$

for ΔI, the increment in the independent variable necessary to give ΔD, the desired increment in the dependent variable. Usually, this computation must be repeated three or four times in a loop in order to yield the desired accuracy since the partial derivative is not a constant and changes with the changing values of the independent variables.

Table X is a listing of the routine called PHSLV, which adjusts the pH of a given compartment to a particular value by varying the H^+ ion plus the Cl^- ion in the input. H^+ and Cl^- are selected by entering a 1.0 in BBB(5,5) and in BBB(6,5) of the Soda-Pop model (see ROWS, pp. 31-32). NBSTAR is then set at 5 in the MAIN routine to indicate that BMULT(5) will be varied, thus varying the input of HCl. In the PHSLV routine, first the compartment is found by comparing NAME in the call statement with the stored names of the compartments. Next, the species H^+ and H_2O are found and the pH computed. DIFF is the increment in pH required.

A second solution is obtained after incrementing BMULT(5), and hence the HCl input a small amount. The $\partial pH/\partial HCl$ is then computed and

$$\Delta HCl \frac{\partial pH}{\partial HCl} = DIFF$$

is solved for ΔHCl. BMULT(5) is incremented by this amount and the loop is repeated. Eventually, if the procedure converges, the error in pH is less than the criterion. Table XI is the printed result of the FORTRAN statement CALL PHSLV (LIQUID, 3.5) in the MAIN routine (see bottom of Table VIII, p. 68 above).

Table X

SUBROUTINE pHSOLVE FOR GOALING pH

```
6472,SDCA,F37SC,8,50,1CC,P                    FORTRAN SOURCE LIST                    07/24/67      PAGE  5
      ISN       SOURCE STATEMENT

       C $IBFTC PHSLV   NODECK                                                                    PHSLCC2C
       1        SUBROUTINE PHSLV(NAME,VALLE)                                                     PHSLCC3C
         C                        4-7-67                                                         PHSLCC5C
         C             CODE MODIFIED FOR PACKED MATRIX STORAGE                                   PHSLC060
         C             A(I,J) STORED AS THREE ENTRIES                                            PHSLCC7C
         C                IROW(K) = I                                                            PHSLCC8C
         C                JCOL(K) = J                                                            PHSLCC9C
         C                AIJ(K) = A(I,J)                                                        PHSLC1CC
         C                                                                                       PHSLC15C
       2        COMMON AIJ(460), IROW(460), JCOL(460)                                           PHSLC16C
         C                                                                                       PHSLC17C
       3        COMMON  /INPT/KA(12),KB(12),BBB(60,5),PH(25),T(20),BMULT(5)                     PHSLC18C
         C                                                                                       PHSLC19C
       4        COMMON /SLVE/IV(30),TCL(20),NR(60,2),B(60),PIE(75),V1(75),V2(75),               PHSLC2CC
              1         V3(75),V4(75),X(170),XMF(170),XN(170),C(170),X1(170),                  PHSLC21C
              2         X2(170),X3(170),XBAR(25),NAM(25,2),KL(26), R(75,75),                   PHSLC22C
              3         JCOMP(170),FE,FE2,ERMB,XERB,NERB,ERMA,XERA,NERA                        PHSLC23C
         C                                                                                       PHSLC24C
       5        EQUIVALENCE (IV(1),M),(IV(2),MEND),(IV(3),NCOMP),(IV(4),N,NTOT),               PHSLC25C
              1         (IV(5),NIT),(IV(6),NCT),(IV(7),MF),(IV(8),ITER),                       PHSLC26C
              2         (IV(9),ITMAX),(IV(10),IERROR),(IV(11),LASTCP),(IV(12),KE),             PHSLC27C
              3         (IV(13),MAXM),(IV(14),MAXP),(IV(15),MAXN),(IV(16),MAXMD),              PHSLC28C
              4         (IV(17),END),(IV(18),BLANK),(IV(19),H2C),(IV(20),HPLUS),               PHSLC29C
              5         (IV(21),NCYCLE),(IV(22),NBSTAR),(IV(23),KPF),                          PHSLC3CC
              6         (IV(24),NAIJ),(IV(25),MAXAIJ),(IV(26),IOPT)                            PHSLC31C
       6        EQUIVALENCE (TCL(3),XMIN),(TOL(4),XSTART),(TOL(5),BARMIN),                     PHSLC32C
              1            (TCL(11),ALITER),(TCL(12),RT)                                       PHSLC33C
         C                                                                                       PHSLC34C
       7        INTEGER MF                                                                      PHSLC35C
      10        INTEGER END,BLANK,H2C,HPLUS                                                     PHSLC36C
         C                                                                                       PHSLC38C
      11        RLOG10=  0.434294                                                               PHSLC39C
         C FIND COMPARTMENT                                                                    PHSLC4CC
      12        DO 1  K =1,NCOMP                                                                PHSLC41C
      13          IF (NAME.EQ.NAM(K,1))  GO TO 2                                                PHSLC42C
      16      1 CONTINUE                                                                        PHSLC43C
         C FIND NAMES                                                                          PHSLC44C
      20        WRITE (NOT,99) NAME                                                             PHSLC45C
      21     99 FORMAT(16H NO COMPARTMENT ,A6)                                                  PHSLC46C
      22        CALL EXIT                                                                       PHSL046C
      23      2 MTA = KL(K)                                                                     PHSLC47C
      24        MTB = KL(K+1) - 1                                                               PHSLC48C
      25        NH2O =0                                                                         PHSLC49C
      26        NHPLUS = 0                                                                      PHSLC5CC
      27        DO 3 J = MTA,MTB                                                                PHSLC51C
      30          IF (XN(J).EQ.H2C) NH2C = J                                                    PHSLC52C
      33          IF (XN(J).EQ.HPLUS) NHPLUS = J                                                PHSLC53C
      36      3 CONTINUE                                                                        PHSLC54C
      40        IF (NH2O.NE.0.AND.NHPLUS.NE.0)  GO TO 4                                         PHSLC55C
      43        WRITE (NOT,98)                                                                  PHSLC56C
      44     98 FORMAT(15H PH NOT DEFINED)                                                      PHSLC57C
      45        CALL EXIT                                                                       PHSLC58C
      46      4 DO  10  IIII =1,5                                                               PHSL059C
      47          CALL ROWS(-1)                                                                 PHSLC6CC
      50          CALL SOLVE                                                                    PHSLC61C
      51          IF (IERROR.NE.1) RETURN                                                       PHSLC62C
      54          PHNOW =  - ALOG(X(NHPLUS)*ALITER/X(NH2O))*RLOG10                              PHSLC63C
      55          DIFF = VALLE - PHNOW                                                          PHSLC64C
      56          IF ( ABS(DIFF) .LT.1.E-3  )      RETURN                                       PHSLC65C
      61          CALL RCALC                                                                    PHSLC85C
      62          CALL MATINV(R,MEND,PIE,0,V2,V3,V4,KE)                                         PHSLC86C
      63          IF (KE.NE.0) RETURN                                                           PHSLC87C
      66          RATE = 0.0                                                                    PHSLC88C
      67        RATE = PART(5,-2,C,-1)  + PART(6,-2,0,-1)                                       PHSLC95C
      70          IF (RATE.EQ.C.)RETURN                                                         PHSLC97C
      73          BMULT(NBSTAR) = BMULT(NBSTAR) + DIFF/ RATE                                    PHSLC98C
      74     10 CONTINUE                                                                       PHSLC99C
      76        WRITE(NOT,91) DIFF                                                              PHSL1CCC
      77     91 FORMAT(16H ERROR IN PH IS F15.6)
     100        RETURN
     101        END
```

Table XI

PRINTED OUTPUT AFTER CALLING PHSLV(LIQUID, 3.5) FOLLOWED BY THE CONTROL CARDS "OUTPUT" AND "PRINTR"

```
RETURN
PROJECTION  1  SIZE        0.00,  SCALE        1.00
ITERATION   1 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   2 MAX CHANGE IN PIE= 6.1063559 E-07 MAX ROW ERROR= 9.9182129 E-05
PROJECTION  1  SIZE        0.46,  SCALE        1.00
ITERATION   1  CHANGE IN FREE ENERGY=-2.6855469 E-03  STEP SIZE= 1.0000000 E 00 AV THETA= 8.60213E-02
ITERATION   2  CHANGE IN FREE ENERGY=-8.5149219 E-04  STEP SIZE= 1.0000000 E 00 AV THETA= 8.59049E-02
ITERATION   3 POSITIVE TDA, GO TO METHOD 2
ITERATION   4 MAX CHANGE IN PIE= 6.0806593 E-07 MAX ROW ERROR=-2.1362305 E-04
PROJECTION  1  SIZE        0.18,  SCALE        1.00
ITERATION   1 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   2 MAX CHANGE IN PIE= 7.4749751 E-07 MAX ROW ERROR= 1.2969971 E-04
PROJECTION  1  SIZE        0.00,  SCALE        1.00
ITERATION   1 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   2 MAX CHANGE IN PIE= 7.9667743 E-07 MAX ROW ERROR= 1.3732910 E-04


EXPERIMENTAL SODA POP DECK      5-1-67


RMS MASS BALANCE ERROR= 7.567E-08  MAX. ERROR= 1.164E-07 ON ROW LACTIC-
RMS  EQUILIBRIUM ERROR= 1.627E-07  MAX. ERROR= 5.037E-07 IN CO3=   OF LIQUID PHASE

OPTIMAL SOLUTION
OBJECTIVE=  -1.1108570 E 04  RT * OBJECTIVE=  -6.8828998 E 06

                GAS PHASE     LIQUID PHASE

X-BAR           9.99151E 02   5.54894E 01

PH              0.            3.49863E 00

O2      MOLES   1.31500E 02   1.29516E-04
        MFRAC   1.31612E-01   2.33406E-06

CO2     MOLES   5.26287E 01   1.27069E-03
        MFRAC   5.26734E-02   2.28998E-05

N2      MOLES   7.54000E 02   4.15793E-04
        MFRAC   7.54640E-01   7.49320E-06

H2O     MOLES   6.10231E 01   5.51766E 01
        MFRAC   6.10749E-02   9.94362E-01

H+      MOLES -0.             3.17089E-04
        MFRAC -0.             5.71441E-06

OH-     MOLES -0.             7.54959E-11
        MFRAC -0.             1.36055E-12

CL-     MOLES -0.             1.39701E-01
        MFRAC -0.             2.51762E-03

NA+     MOLES -0.             1.30000E-01
        MFRAC -0.             2.34279E-03

K+      MOLES -0.             2.00000E-02
        MFRAC -0.             3.60429E-04

GLUCOS  MOLES -0.             1.00000E-02
        MFRAC -0.             1.80215E-04

LACTIC  MOLES -0.             1.00000E-02
        MFRAC -0.             1.80215E-04

HCO3-   MOLES -0.             3.18542E-06
        MFRAC -0.             5.74059E-08

H2CO3   MOLES -0.             1.77831E-06
        MFRAC -0.             3.20478E-08

CO3=    MOLES -0.             5.71559E-13
        MFRAC -0.             1.03003E-14

MISC    MOLES -0.             3.87123E-04
        MFRAC -0.             6.97652E-06

MISC-   MOLES -0.             6.12877E-04
        MFRAC -0.             1.10449E-05

PRINTROWS

ROW NAME                  B                  B1                 B2                 B3                 B4                 B5
                                    MULT.= 1.0000000E 00   0.            1.0000000E 03   0.            -2.9897303E-04

  1  O2           1.3150000E 02   0.            2.0990000E-01   1.3150000E-01   0.            0.
  2  CO2          5.2630000E 01   0.            3.0000000E-04   5.2630000E-02   0.            0.
  3  N2           7.5400000E 02   0.            7.8980000E-01   7.5400000E-01   0.            0.
  4  H2O          1.1619967E 02   5.5139670E 01  0.            6.1060000E-02   0.            0.
  5  H+           7.0102697E-04   1.0000000E-03  0.            0.              0.            1.0000000E 00
  6  CL-          1.3970103E-01   1.4000000E-01  0.            0.              0.            1.0000000E 00
  7  NA+          1.3000000E-01   1.3000000E-01  0.            0.              0.            0.
  8  K+           2.0000000E-02   2.0000000E-02  0.            0.              0.            0.
  9  GLUCOSE      1.0000000E-02   1.0000000E-02  0.            0.              0.            0.
 10  LACTIC-      1.0000000E-02   1.0000000E-02  0.            0.              0.            0.
 11  MISC-        1.0000000E-03   1.0000000E-03  0.            0.              0.            0.
END   OF ROWS IN STORAGE
```

This procedure can be completely generalized; several dependent variables can be satisfied simultaneously by varying corresponding, appropriate independent variables. In this case, Eq. (22) should be read as a matrix equation, which is solved by calling MATINV appropriately. The partial derivatives are found by using PART, a subroutine normally called by JACOBS for computing the partial derivatives listed by that routine. The theory is exactly the same as in the simpler case above; difficulties arise only in the logical sorting and incrementing of the variables appropriately, and in computing partial derivatives with respect to compounds (e.g., HCl) instead of just components. Such partials are computed as the sum of the partials of each component taken separately. Also, it is possible to compute the partial derivatives of certain output variables with respect to either the thermodynamic parameters (c(j)) or the stoichiometric coefficients of the matrix of constraints. This generalization of the goal routines has been accomplished in the thesis of Magnier [19], where the subroutine is called GOALN8 (also see JACOB, p. 106 ff., and PART, pp. 121-122 below).

EXAMPLE III--BSA SOLUTION

The following example (slightly more complex than Soda Pop) deals with the ionization of protein, serum albumin. Table XII is a listing of the model to be used; it contains one liter of H_2O at 25°C, 1.0 millimole of (bovine) serum albumin, and 0.15 mole of NaCl. (An excess of H^+ ion is discussed below.)

Table XII

DATA DECK FOR EXPERIMENTAL MODEL OF SERUM ALBUMIN SOLUTION

```
BSA SOLUTION - ONE LITER H2O    25 DEG. C.

ROWS
  1  H+              5.5469963E 01   -0.          -0.          -0.          -0.
  2  OH-             5.5343600E 01   -0.          -0.          -0.          -0.
  3  NA+             1.5000000E-01   -0.          -0.          -0.          -0.
  4  CL-             1.5000000E-01   -0.          -0.          -0.          -0.
  5  PROTEIN         1.0000000E-03   -0.          -0.          -0.          -0.
  6  ACARB  SITES    0.              -0.          -0.          -0.          -0.
  7  BCARB  SITES    0.              -0.          -0.          -0.          -0.
  8  IMID   SITES    0.              -0.          -0.          -0.          -0.
  9  AMINO  SITES    0.              -0.          -0.          -0.          -0.
 10  EAMINO SITES    0.              -0.          -0.          -0.          -0.
 11  PHENOL SITES    0.              -0.          -0.          -0.          -0.
 12  GUANID SITES    0.              -0.          -0.          -0.          -0.

MATRIX

   PRCSOL
  1      H+              0.000000        1.000 H+
  2      OH-             0.000000        1.000 OH-
  3      H2O           -40.254170        1.000 H+        1.000 OH-
  4      NA+             0.000000        1.000 NA+
  5      CL-             0.000000        1.000 CL-
  6      PROTN           0.000000        1.000 PROTEI -105.100 BCARB   -17.000 IMID    -60.500 EAMINO
  6      PROTN           0.000000      -20.200 PHENOL  -23.400 GUANID   -1.060 ACARB    -1.060 AMINO

   PROSITES
  7      ACOOH         -12.648270        1.000 H+        1.000 ACARB
  8      ACOO-           0.000000       -0.000           1.000 ACARB
  9      BCOOH         -13.269970        1.000 H+        1.000 BCARB
 10      BCOO-           0.000000       -0.000           1.000 BCARB
 11      IMID+         -19.901430        1.000 H+        1.000 IMID
 12      IMID            0.000000       -0.000           1.000 IMID
 13      AMIN+         -21.858630        1.000 H+        1.000 AMINO
 14      AMIN            0.000000       -0.000           1.000 AMINO
 15      EAMIN+        -26.578940        1.000 H+        1.000 EAMINO
 16      EAMIN           0.000000       -0.000           1.000 EAMINO
 17      PHENO         -27.845370        1.000 H+        1.000 PHENOL
 18      PHENO-          0.000000       -0.000           1.000 PHENOL
 19      GUAN+         -31.644650        1.000 H+        1.000 GUANID
 20      GUAN            0.000000       -0.000           1.000 GUANID

PROBLEM HAS    12 ROWS,    20 COLUMNS,     2 COMPARTMENTS,      35 NON ZERO MATRIX ENTRIES.

MULTIPLIERS
 MUL(1)= 1.00000E 00  MUL(2)=-0.          MUL(3)=-0.          MUL(4)=-0.          MUL(5)=-0.
```

This example is described in detail and the titration curves are computed in DeLand [12]; here, we show only the essential details. The example illustrates the division of the protein into subclasses of homogeneous binding sites for H^+ ion. Each class of sites ionizes at an assigned pK (which may be determined during the course of the computation using, say, Linderstrom-Lang theory). There are five principal classes; the first contains 105 carboxyl sites, the second 17 imidazole sites, and so on--226 in all, as listed in the species PROTN in Table XII above. H^+ ion is to react with the protein in solution and the fraction of sites of each class having a proton bound depends upon the pK of the class and the pH. In principle, then, one can compute successive steps along the titration curve merely by adding HCl or NaOH and calling SOLVE and OUTPUT. In the output, one can read the pH and the moles of each class ionized; i.e., in equilibrium at that pH.

In practice, it is slightly more complicated because there are 2^{226} possible species in all combinations of ionized and un-ionized sites to be computed--obviously impossible. Instead, since each class of sites is homogeneous, each is treated as a monobasic acid of concentration N times the concentration of the protein, where N is the number of sites in each class; and the ionization of this acid is computed separately at its assigned pK--i.e., the protein ionization is treated as though the solution contained monobasic acids corresponding exactly to the assumed protein ionization sites and each acting independently of the other. Under these assumptions (that each class is homogeneous and the classes--indeed, the sites--within a class, are independent and do not effect

the ionization of each other), Shapiro [13] has shown that the number of moles of ionized monobasic acid will equal the number of moles of ionized protein sites under the same conditions in the original protein.

To effect this computation, it is necessary to transfer the monobasic acids to a separate conceptual compartment in order not to change the effective ionic strength of the protein solution. Shapiro [8] describes the mathematical method and proves the equivalence of the new, compartmented problem with the old or protein solution problem. In Table XII (p. 79) there are five constraints (mass conservation equations) which transfer the ionization sites to a separate conceptual compartment, called PROSITES, for ionization. This may be seen more easily from the PRINTTABLEAU instruction:

MATRIX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	1	0	1	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
2	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	-1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	**	0	0	1	1	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	**	0	0	0	0	1	1	0	0	0	0	0	0	0	0
9	0	0	0	0	0	-1	0	0	0	0	0	0	1	1	0	0	0	0	0	0
10	0	0	0	0	0	**	0	0	0	0	0	0	0	0	1	1	0	0	0	0
11	0	0	0	0	0	**	0	0	0	0	0	0	0	0	0	0	1	1	0	0
12	0	0	0	0	0	**	0	0	0	0	0	0	0	0	0	0	0	0	1	1

PROTEIN SOLUTION ◄---------► PROSITES IONIZATION

Because the tableau rounds the matrix entries to the tens-and-units decimal places only, the full number of the full detached stoichiometric coefficient is not reproduced; however, by reading the listing (Table XII) simultaneously, we form the following equations between the compartments PROSITES and PROSOLUTION:

$$\begin{aligned} -105.1\ x_6 + 1.0\ x_7 + 1.0\ x_8 &= 0 \\ -17.0\ x_6 + 1.0\ x_9 + 1.0\ x_{10} &= 0 \\ \vdots \qquad\qquad\qquad\qquad & \vdots \\ -23.4\ x_6 + 1.0\ x_{15} + 1.0\ x_{16} &= 0 \end{aligned} \tag{23}$$

And within the compartment PROSITE, the beta-carboxyl sites, for example, are divided into two species: the ionized and unionized with the pK (from Eq. 17, p. 17 above):

$$pK_7 = \frac{-12.760624 + \ln \text{ALITER}}{2.30259}$$

$$= 3.76 \ .$$

Using beta carboxyl as an example, Eqs. (23) require that the sum of the moles of the species x(7) plus x(8) equal 105.1 times the moles of protein; i.e., there are 105.1 (empirically effective number, molecular weight 69,000) beta-carboxyl sites per molecule of BSA and, in this example, 105.1 moles of an equivalent monobasic acid ionizing.

With this model, we SOLVE, OUTPUT, and then RETURN control to the MAIN routine. Table XIII is a list of the MAIN routine in which we search for the isoionic point for this protein solution. Table XIV is the result of this calculation. Note that since the net charge on the components going into the model must be zero, the charge of the input component protein must be $-0.1264/1.0 \times 10^{-3}$ per mole, since there is a surplus of H^+ ion of 0.1264 moles.

Table XIII

MAIN ROUTINE FOR COMPUTING ISOIONIC POINT OF PROTEIN SOLUTION

```
    C
    C ISOIONIC POINT CALCULATION                                    NZS20220
    C
 47      1 DH=.001
 50        SUM1=  -X(8)-X(10)+X(11)+X(13)+X(15)-X(18)+X(19)
 51        DO 200 I=1,10
 52        BBB(1,1)=BBB(1,1)+DH
 53        CALL ROWS(-1)
 54        CALL SOLVE
 55        IF(IERROR.NE.1) CALL EXIT
 60        SUM2=  -X(8)-X(10)+X(11)+X(13)+X(15)-X(18)+X(19)
 61        IF(ABS(SUM2).LT..CCC01) GO TO 201
 64        DH= ( (SUM2-0.) / (SUM1-SUM2) ) *DH
 65        SUM1=SUM2
 66    200 CONTINUE
 70    201 CALL OUTPUT
 71        WRITE(6,300) NR(1,1),NR(1,2),BBB(1,1)
 72    300 FORMAT(1X,2A6,3H = 1PE20.8)
 73        WRITE(6,301) KN(11),X(11),KN(8),X(8),KN(13),X(13),KN(10),X(10),
          X   KN(15),X(15),KN(18),X(18),KN(19),X(19)
 74    301 FORMAT(1H0,16X,1H+,18X,1H-/(6X,A6,1PE13.5,1X,A6,1PE13.5/))
    C
    C REPEAT, USING REDUCED VOLUME OF SOLVENT (1.0 LITER OF SOLUTION)
 75        CALL INPUT
 76        GO TO 1
 77      7 CALL EXIT
100        END
```

Table XIV

COMPUTED DISTRIBUTION OF SPECIES FOR:

a) The isoionic point using intrinsic pK_i

X-BAR		5.56446E 01	2.28320E-01
PH		5.35324E 00	-0.
H+	MOLES	4.44594E-06	-0.
	MFRAC	7.98988E-08	-0.
OH-	MOLES	2.28225E-09	-0.
	MFRAC	4.10148E-11	-0.
H2O	MOLES	5.53436E 01	-0.
	MFRAC	9.94591E-01	-0.
NA+	MOLES	1.50000E-01	-0.
	MFRAC	2.69568E-03	-0.
CL-	MOLES	1.50000E-01	-0.
	MFRAC	2.69568E-03	-0.
PROTN	MOLES	1.00001E-03	-0.
	MFRAC	1.79714E-05	-0.
ACOOH	MOLES	-0.	2.57189E-05
	MFRAC	-0.	1.12644E-04
ACOO-	MOLES	-0.	1.03428E-03
	MFRAC	-0.	4.52996E-03
BCOOH	MOLES	-0.	4.65115E-03
	MFRAC	-0.	2.03712E-02
BCOO-	MOLES	-0.	1.00449E-01
	MFRAC	-0.	4.39948E-01
IMIC+	MOLES	-0.	1.65294E-02
	MFRAC	-0.	7.23958E-02
IMID	MOLES	-0.	4.70583E-04
	MFRAC	-0.	2.06107E-03
AMIN+	MOLES	-0.	1.05575E-03
	MFRAC	-0.	4.62401E-03
AMIN	MOLES	-0.	4.24561E-06
	MFRAC	-0.	1.85950E-05
EAMIN+	MOLES	-0.	6.04978E-02
	MFRAC	-0.	2.64969E-01
EAMIN	MOLES	-0.	2.16827E-06
	MFRAC	-0.	9.49661E-06
PHENO	MOLES	-0.	2.01998E-02
	MFRAC	-0.	8.84714E-02
PHENO-	MOLES	-0.	2.04046E-07
	MFRAC	-0.	8.93660E-07
GUAN+	MOLES	-0.	2.34000E-02
	MFRAC	-0.	1.02488E-01
GUAN	MOLES	-0.	1.67332E-09
	MFRAC	-0.	7.32885E-09

b) The isoionic point using reduced volume of solvent (1.0 liter of solution)

X-BAR		4.75478E 01	2.28320E-01
PH		5.35276E 00	-0.
H+	MOLES	3.79968E-06	-0.
	MFRAC	7.99128E-08	-0.
OH-	MOLES	1.94802E-09	-0.
	MFRAC	4.09696E-11	-0.
H2O	MOLES	4.72468E 01	-0.
	MFRAC	9.93669E-01	-0.
NA+	MOLES	1.50000E-01	-0.
	MFRAC	3.15472E-03	-0.
CL-	MOLES	1.50000E-01	-0.
	MFRAC	3.15472E-03	-0.
PROTN	MOLES	9.99995E-04	-0.
	MFRAC	2.10313E-05	-0.
ACOOH	MOLES	-0.	2.57233E-05
	MFRAC	-0.	1.12663E-04
ACOO-	MOLES	-0.	1.03428E-03
	MFRAC	-0.	4.52994E-03
BCOOH	MOLES	-0.	4.65193E-03
	MFRAC	-0.	2.03746E-02
BCOO-	MOLES	-0.	1.00448E-01
	MFRAC	-0.	4.39944E-01
IMIC+	MOLES	-0.	1.65295E-02
	MFRAC	-0.	7.23962E-02
IMID	MOLES	-0.	4.70503E-04
	MFRAC	-0.	2.06072E-03
AMIN+	MOLES	-0.	1.05576E-03
	MFRAC	-0.	4.62401E-03
AMIN	MOLES	-0.	4.24487E-06
	MFRAC	-0.	1.85918E-05
EAMIN+	MOLES	-0.	6.04979E-02
	MFRAC	-0.	2.64970E-01
EAMIN	MOLES	-0.	2.16789E-06
	MFRAC	-0.	9.49495E-06
PHENO	MOLES	-0.	2.01998E-02
	MFRAC	-0.	8.84714E-02
PHENO-	MOLES	-0.	2.04005E-07
	MFRAC	-0.	8.93503E-07
GUAN+	MOLES	-0.	2.34000E-02
	MFRAC	-0.	1.02488E-01
GUAN	MOLES	-0.	1.67303E-09
	MFRAC	-0.	7.32757E-09

This exact amount of H^+ ion is required to be attached in order to make the specified protein neutral in this solution. This is determined as follows: starting the computation with an arbitrary amount of H^+ ion, the problem is solved (equilibrium computed) and the net charge on the protein, considering the sign of each site, is algebraically added. If this charge is not zero, H^+ ion (as a component) is added or subtracted in an iterative loop until the protein net charge is satisfactorily small--in this case, less than 0.00001. At this point, the results are written in the output.

Evidently, because of the H^+ ion reaction with H_2O, the pH of the solution is also changing during the iteration, and the isoionic pH is that pH at which the loop finally indicates the protein to have zero net charge. Also, the isoionic pH, and the H^+ ion required to attain that pH, will be different with a different assignment of the pK_i. The H^+ ion surplus in this case is just that required to make the input components to an isoionic solution neutral. An equivalent statement is that the protein entered the model as sodium protinate and the neutral molecule HCl was added until the isoionic point was reached.

EXAMPLE IV--HUMAN BLOOD

Table XV begins with a list of components and continues through an entire printed output for an elementary model of the respiratory chemistry of human blood. This model is taken from DeLand [20], where it is described in detail; earlier models of the blood are described in

Table XV

DATA DECK AND OUTPUT FOR COMPLETE COMPUTER RUN OF ELEMENTARY BLOOD MODEL

```
ROWS
  1  O2              6.8300000E-03    2.0990000E-01    1.3150000E-01   -0.             -0.
  2  CO2             2.3490000E-02    3.0000000E-04    5.2600000E-02   -0.             -0.
  3  N2              4.3700000E-04    7.8980000E-01    7.5400000E-01   -0.             -0.
  4  H+              4.6500000E 01    -0.              6.1099999E-02   -0.             -0.
  5  OH-             4.6520200E 01    -0.              6.1099999E-02   -0.             -0.
  6  CL-             8.0050000E-02    -0.              -0.             -0.             -0.
  7  NA+             8.4820000E-02    -0.              -0.             -0.             -0.
  8  K+              4.5050000E-02    -0.              -0.             -0.             -0.
  9  CA++            1.6050000E-03    -0.              -0.             -0.             -0.
 10  MG++            1.6125000E-03    -0.              -0.             -0.             -0.
 11  SO4=            5.2500000E-04    -0.              -0.             -0.             -0.
 12  HPO4=           1.4300000E-03    -0.              -0.             -0.             -0.
 13  UREA            3.1420000E-03    -0.              -0.             -0.             -0.
 14  GLUCOSE         3.6660000E-03    -0.              -0.             -0.             -0.
 15  LACTIC-         2.0330000E-03    -0.              -0.             -0.             -0.
 16  NH4+            8.6900000E-04    -0.              -0.             -0.             -0.
 17  MISCPLASMA      8.7060000E-04    -0.              -0.             -0.             -0.
 18  MISCREDCELL     3.7505000E-03    -0.              -0.             -0.             -0.
 19  HB4             9.0899999E-03    -0.              -0.             -0.             -0.
 20  *PLASMA        -0.               -0.              -0.             -0.             -0.
 21  NA*             7.6441400E-02    -0.              -0.             -0.             -0.
 22  K*              2.2963500E-03    -0.              -0.             -0.             -0.
 23  CA*             1.3796000E-03    -0.              -0.             -0.             -0.
 24  MG*             4.6681800E-04    -0.              -0.             -0.             -0.

MATRIX

   AIR OUT
  1       O2            -10.940000          1.000 O2
  2       CO2            -7.690000          1.000 CO2
  3       N2            -11.520000          1.000 N2
  4       H2O           -36.600000          1.000 H+        1.000 OH-

   PLASMA
  5       O2              0.000000          1.000 O2
  6       CO2             0.000000          1.000 CO2
  7       N2              0.000000          1.000 N2
  8       H+              0.000000          1.000 H+        1.000 *PLASM
  9       OH-             0.000000          1.000 OH-      -1.000 *PLASM
 10       CL-             0.000000          1.000 CL-      -1.000 *PLASM
 11       NA+             0.000000          1.000 NA+       1.000 *PLASM
 11       NA+            -0.000000          1.000 NA*
 12       K+              0.000000          1.000 K+        1.000 *PLASM
 12       K+             -0.000000          1.000 K*
 13       CA++            0.000000          1.000 CA++      2.000 *PLASM
 13       CA++           -0.000000          1.000 CA*
 14       MG++            0.000000          1.000 MG++      2.000 *PLASM
 14       MG++           -0.000000          1.000 MG*
 15       SO4=            0.000000          1.000 SO4=     -2.000 *PLASM
 16       HPO4=           0.000000          1.000 HPO4=    -2.000 *PLASM
 17       UREA            0.000000          1.000 UREA
 18       GLUCOS          0.000000          1.000 GLUCOS
 19       LACTIC          0.000000          1.000 LACTIC   -1.000 *PLASM
 20       NH4+            0.000000          1.000 NH4+      1.000 *PLASM
 21       HCO3-         -21.350000          1.000 CO2       1.000 OH-      -1.000 *PLASM
 22       H2CO3         -32.840000          1.000 CO2       1.000 H+        1.000 OH-
 23       CO3=            6.260000          1.000 CO2      -1.000 H+        1.000 OH-      -2.000 *PLASM
 24       H2O           -39.390000          1.000 H+        1.000 OH-
 25       MISCPR          0.000000          1.000 MISCPL  -10.000 *PLASM

   RED CELLS
 26       O2             -0.000000          1.000 O2
 27       CO2             0.000000          1.000 CO2
 28       N2             -0.000000          1.000 N2
 29       H+              0.000000          1.000 H+
 30       OH-             0.000000          1.000 OH-
 31       CL-             0.000000          1.000 CL-
 32       NA+             2.193399          1.000 NA+
 33       K+             -2.941575          1.000 K+
 34       CA++            2.251790          1.000 CA++
 35       MG++           -0.457703          1.000 MG++
 36       SO4=           -2.000000          1.000 SO4=
 37       HPO4=          -2.000000          1.000 HPO4=
 38       UREA            0.000000          1.000 UREA
 39       GLUCOS          0.000000          1.000 GLUCOS
 40       LACTIC          0.000000          1.000 LACTIC
 41       NH4+            0.000000          1.000 NH4+
 42       HCO3-         -21.490000          1.000 CO2       1.000 OH-
 43       H2CO3         -32.840000          1.000 CO2       1.000 H+        1.000 OH-
 44       CO3=            6.120000          1.000 CO2      -1.000 H+        1.000 OH-
 45       H2O           -39.390000          1.000 H+        1.000 OH-
 46       MISCPR          0.000000          1.000 MISCRE
 47       HB4             0.000000          1.000 HB4
 48       HB4O2         -16.230000          1.000 O2        1.000 HB4

PROBLEM HAS    24 ROWS,    48 COLUMNS,    3 COMPARTMENTS,    80 NON ZERO MATRIX ENTRIES.
```

Table XV--Continued

```
MULTIPLIERS
 MUL(1)= 1.0000E 00  MUL(2)=-0.          MUL(3)= 1.0000E 03  MUL(4)=-0.          MUL(5)=-0.

VECTORX
 AIR OUT       O2        1.314980E 02   CO2       5.260220E 01   N2        7.540000E 02
 AIR OUT       H2O       6.102650E 01              -0.                     -0.
 PLASMA        O2        6.742080E-05   CO2       6.955620E-04   N2        2.164490E-04
 PLASMA        H+        2.108430E-08   OH-       3.076910E-07   CL-       5.742560E-02
 PLASMA        NA+       7.644130E-02   K+        2.296330E-03   CA++      1.379590E-03
 PLASMA        MG++      4.668090E-04   SO4=      1.844830E-04   HPO4=     5.024960E-04
 PLASMA        UREA      1.937870E-03   GLUCOS    2.261050E-03   LACTIC    1.458420E-03
 PLASMA        NH4+      4.388880E-04   HCO3-     1.386630E-02   H2CO3     9.892060E-07
 PLASMA        CO3=      1.940030E-05   H2O       2.872480E 01   MISCPR    8.706000E-04
 RED CELLS     O2        4.189320E-05   CO2       4.322010E-04   N2        1.344950E-04
 RED CELLS     H+        2.066270E-08   OH-       1.212240E-07   CL-       2.262440E-02
 RED CELLS     NA+       8.378670E-03   K+        4.275370E-02   CA++      2.254050E-04
 RED CELLS     MG++      1.145690E-03   SO4=      3.405170E-04   HPO4=     9.275040E-04
 RED CELLS     UREA      1.204130E-03   GLUCOS    1.404950E-03   LACTIC    5.745840E-04
 RED CELLS     NH4+      4.301120E-04   HCO3-     6.283980E-03   H2CO3     6.146620E-07
 RED CELLS     CO3=      5.574480E-06   H2O       1.784870E 01   MISCPR    3.750500E-03
 RED CELLS     HB4       3.353680E-04   HB4O2     8.754630E-03              -0.
END                     -0.                     -0.                       -0.

SOLVE
PROJECTION  1  SIZE        0.00,  SCALE        1.00
ITERATION   1 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   2 MAX CHANGE IN PIE= 1.7160532 E-02 MAX ROW ERROR= 5.4931641 E-04
ITERATION   3 MAX CHANGE IN PIE= 5.9323630 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   4 MAX CHANGE IN PIE= 2.4503556 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   5 MAX CHANGE IN PIE= 1.3516563 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   6 MAX CHANGE IN PIE= 2.4093508 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   7 MAX CHANGE IN PIE= 1.2332967 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   8 MAX CHANGE IN PIE= 1.3591567 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   9 MAX CHANGE IN PIE= 2.4347612 E-05 MAX ROW ERROR=-5.3405762 E-05

OUTPUT

RMS MASS BALANCE ERROR= 9.006E-08  MAX. ERROR= 2.127E-07 ON ROW H+
RMS  EQUILIBRIUM ERROR= 2.058E-07  MAX. ERROR= 3.385E-07 IN CO3=   OF  PLASMA

OPTIMAL SOLUTION
OBJECTIVE=  -1.5404338 E 04  RT * OBJECTIVE=  -9.4932936 E 06

                  AIR OUT       PLASMA        RED CELLS

X-BAR             9.99127E 02   2.88853E 01   1.79485E 01

PH                0.            7.39233E 00   7.19445E 00

O2        MOLES   1.31498E 02   6.74207E-05   4.18934E-05
          MFRAC   1.31613E-01   2.33409E-06   2.33409E-06

CO2       MOLES   5.26022E 01   6.95560E-04   4.32202E-04
          MFRAC   5.26482E-02   2.40801E-05   2.40801E-05

N2        MOLES   7.54000E 02   2.16449E-04   1.34495E-04
          MFRAC   7.54659E-01   7.49339E-06   7.49339E-06

H2O       MOLES   6.10265E 01   2.87247E 01   1.78488E 01
          MFRAC   6.10799E-02   9.94442E-01   9.94443E-01

H+        MOLES  -0.            2.10837E-08   2.06621E-08
          MFRAC  -0.            7.29913E-10   1.15119E-09

OH-       MOLES  -0.            3.07698E-07   1.21228E-07
          MFRAC  -0.            1.06524E-08   6.75419E-09

CL-       MOLES  -0.            5.74254E-02   2.26246E-02
          MFRAC  -0.            1.98805E-03   1.26053E-03

NA+       MOLES  -0.            7.64414E-02   8.37860E-03
          MFRAC  -0.            2.64638E-03   4.66813E-04

K+        MOLES  -0.            2.29635E-03   4.27537E-02
          MFRAC  -0.            7.94990E-05   2.38202E-03

CA++      MOLES  -0.            1.37960E-03   2.25400E-04
          MFRAC  -0.            4.77614E-05   1.25581E-05

MG++      MOLES  -0.            4.66818E-04   1.14568E-03
          MFRAC  -0.            1.61611E-05   6.38316E-05

SO4=      MOLES  -0.            1.84481E-04   3.40519E-04
          MFRAC  -0.            6.38667E-06   1.89720E-05

HPO4=     MOLES  -0.            5.02490E-04   9.27510E-04
          MFRAC  -0.            1.73961E-05   5.16761E-05

UREA      MOLES  -0.            1.93786E-03   1.20414E-03
          MFRAC  -0.            6.70883E-05   6.70883E-05
```

Table XV--Continued

```
GLUCOS  MOLES -0.          2.26105E-03  1.40495E-03
        MFRAC -0.          7.82768E-05  7.82768E-05

LACTIC  MOLES -0.          1.45841E-03  5.74589E-04
        MFRAC -0.          5.04898E-05  3.20131E-05

NH4+    MOLES -0.          4.38888E-04  4.30112E-04
        MFRAC -0.          1.51942E-05  2.39636E-05

HCO3-   MOLES -0.          1.38666E-02  6.28419E-03
        MFRAC -0.          4.80059E-04  3.50123E-04

H2CO3   MOLES -0.          9.89204E-07  6.14664E-07
        MFRAC -0.          3.42460E-08  3.42460E-08

CO3=    MOLES -0.          1.94012E-05  5.57484E-06
        MFRAC -0.          6.71665E-07  3.10601E-07

MISCPR  MOLES -0.          8.70600E-04  3.75050E-03
        MFRAC -0.          3.01399E-05  2.08959E-04

HB4     MOLES -0.         -0.           3.35368E-04
        MFRAC -0.         -0.           1.86850E-05

HB4O2   MOLES -0.         -0.           8.75463E-03
        MFRAC -0.         -0.           4.87763E-04

PRINTPIE

                          PIE              PIE*RT           B

O2                     -12.96789014     -7991.773926    1.3150683E 02
CO2                    -10.63412392     -6553.534424    5.2623489E 01
N2                     -11.80148911     -7272.951233    7.5400043E 02
H+                     -20.58247089    -12684.442261    1.0760000E 02
OH-                    -18.81310201    -11594.026123    1.0762020E 02
CL-                     -6.67622477     -4114.383911    8.0050000E-02
NA+                     -5.47618318     -3374.829468    8.4820000E-02
K+                      -8.98138273     -5534.992920    4.5050000E-02
CA++                    -9.03335118     -5567.019714    1.6050000E-03
MG++                   -10.11696613     -6234.823486    1.6125000E-03
SO4=                   -12.87254632     -7933.015991    5.2500000E-04
HPO4=                  -11.87051487     -7315.490051    1.4300000E-03
UREA                    -9.60950029     -5922.085449    3.1420000E-03
GLUCOSE                 -9.45525861     -5827.030334    3.6660000E-03
LACTIC-                -10.34936368     -6378.044067    2.0330000E-03
NH4+                   -10.63897252     -6556.522461    8.6900000E-04
MISCPLASMA             -14.96590781     -9223.100342    8.7060000E-04
MISCREDCELL             -8.47337353     -5221.920044    3.7505000E-03
HB4                    -10.88778973     -6709.862061    9.0899999E-03
*PLASMA                 -0.45562486      -280.789764    0.
NA*                     -0.00275413        -1.697296    7.6441400E-02
K*                      -0.00275797        -1.699668    2.2963500E-03
CA*                     -0.00469211        -2.891626    1.3796000E-03
MG*                     -0.00468655        -2.888201    4.6681800E-04

DELETE
** ROW **MG*          ** HAS BEEN DELETED **

DELETE
** ROW **CA*          ** HAS BEEN DELETED **

DELETE
** ROW **K*           ** HAS BEEN DELETED **

DELETE
** ROW **NA*          ** HAS BEEN DELETED **

SOLVE
PROJECTION  1  SIZE        0.00,  SCALE        1.00
ITERATION   1 AV THETA LESS THAN TOL(1), GO TO METHOD 2
ITERATION   2 MAX CHANGE IN PIE= 5.7120262 E-04 MAX ROW ERROR= 4.5967102 E-04
ITERATION   3 MAX CHANGE IN PIE= 1.6741708 E-05 MAX ROW ERROR=-5.3405762 E-05
ITERATION   4 MAX CHANGE IN PIE= 5.4068837 E-06 MAX ROW ERROR=-5.3405762 E-05

OUTPUT

RMS MASS BALANCE ERROR= 2.308E-07  MAX. ERROR= 7.091E-07 ON ROW H+
RMS  EQUILIBRIUM ERROR= 6.388E-07  MAX. ERROR= 8.121E-07 IN CO3=   OF  RED CELLS

OPTIMAL SOLUTION
OBJECTIVE=  -1.5404338 E 04  RT * OBJECTIVE=  -9.4932936 E 06

                AIR OUT       PLASMA        RED CELLS

X-BAR          9.9912?E 02  2.88853E 01  1.79485E 01

PH             0.           7.39233E 00  7.19446E 00

O2      MOLES  1.?1498E 02  6.74207E-05  4.18934E-05
        MFRAC  1.31613E-01  2.33408E-06  2.33408E-06

CO2     MOLES  5.26022E 01  6.95561E-04  4.32202E-04
        MFRAC  5.26482E-02  2.40801E-05  2.40801E-05
```

Table XV--Continued

```
N2        MOLES  7.54000E 02  2.16449E-04  1.34695E-04
          MFRAC  7.54659E-01  7.49339E-06  7.49339E-06

H2O       MOLES  6.10266E 01  2.87248E 01  1.78488E 01
          MFRAC  6.10799E-02  9.94442E-01  9.94443E-01

H+        MOLES -0.           2.10835E-08  2.06618E-08
          MFRAC -0.           7.29905E-10  1.15117E-09

OH-       MOLES -0.           3.07702E-07  1.21230E-07
          MFRAC -0.           1.06526E-08  6.75430E-09

CL-       MOLES -0.           5.74253E-02  2.26247E-02
          MFRAC -0.           1.98805E-03  1.26053E-03

NA+       MOLES -0.           7.64415E-02  8.37855E-03
          MFRAC -0.           2.64638E-03  4.66810E-04

K+        MOLES -0.           2.29636E-03  4.27536E-02
          MFRAC -0.           7.94995E-05  2.38202E-03

CA++      MOLES -0.           1.37960E-03  2.25398E-04
          MFRAC -0.           4.77614E-05  1.25580E-05

MG++      MOLES -0.           4.66822E-04  1.14568E-03
          MFRAC -0.           1.61612E-05  6.38314E-05

SO4=      MOLES -0.           1.84480E-04  3.40520E-04
          MFRAC -0.           6.38663E-06  1.89721E-05

HPO4=     MOLES -0.           5.02487E-04  9.27513E-04
          MFRAC -0.           1.73960E-05  5.16763E-05

UREA      MOLES -0.           1.93786E-03  1.20414E-03
          MFRAC -0.           6.70883E-05  6.70883E-05

GLUCOS    MOLES -0.           2.26105E-03  1.40495E-03
          MFRAC -0.           7.82768E-05  7.82768E-05

LACTIC    MOLES -0.           1.45841E-03  5.74590E-04
          MFRAC -0.           5.04897E-05  3.20133E-05

NH4+      MOLES -0.           4.38890E-04  4.30111E-04
          MFRAC -0.           1.51942E-05  2.39636E-05

HCO3-     MOLES -0.           1.38668E-02  6.28429E-03
          MFRAC -0.           4.80064E-04  3.50129E-04

H2CO3     MOLES -0.           9.89205E-07  6.14664E-07
          MFRAC -0.           3.42460E-08  3.42460E-08

CO3=      MOLES -0.           1.94017E-05  5.57502E-06
          MFRAC -0.           6.71680E-07  3.10612E-07

MISCPR    MOLES -0.           8.70600E-04  3.75050E-03
          MFRAC -0.           3.01399E-05  2.08959E-04

HB4       MOLES -0.          -0.           3.35368E-04
          MFRAC -0.          -0.           1.86850E-05

HB4O7     MOLES -0.          -0.           8.75463E-03
          MFRAC -0.          -0.           4.87764E-04

PUNCHX

PRINTROWS

ROW NAME                    B                B1                B2                B3                B4                B5
                                        MULT.= 1.0000000E 00  -0.  1.0000000E 03  -0.  -0.

  1  O2              1.3150683E 02    6.8300000E-03    2.0990000E-01    1.3150000E-01  -0.  -0.
  2  CO2             5.2623489E 01    2.3490000E-02    3.0000000E-04    5.2600000E-02  -0.  -0.
  3  N2              7.5400043E 02    4.3700000E-04    7.8980000E-01    7.5400000E-01  -0.  -0.
  4  H+              1.0760000E 02    4.6500000E 01  -0.                 6.1099999E-02  -0.  -0.
  5  OH-             1.0762020E 02    4.6520200E 01  -0.                 6.1099999E-02  -0.  -0.
  6  CL-             8.0050000E-02    8.0050000E-02  -0.               -0.              -0.  -0.
  7  NA+             8.4820000E-02    8.4820000E-02  -0.               -0.              -0.  -0.
  8  K+              4.5050000E-02    4.5050000E-02  -0.               -0.              -0.  -0.
  9  CA++            1.6050000E-03    1.6050000E-03  -0.               -0.              -0.  -0.
 10  MG++            1.6125000E-03    1.6125000E-03  -0.               -0.              -0.  -0.
 11  SO4=            5.2500000E-04    5.2500000E-04  -0.               -0.              -0.  -0.
 12  HPO4=           1.4300000E-03    1.4300000E-03  -0.               -0.              -0.  -0.
 13  UREA            3.1420000E-03    3.1420000E-03  -0.               -0.              -0.  -0.
 14  GLUCOSE         3.6660000E-03    3.6660000E-03  -0.               -0.              -0.  -0.
 15  LACTIC-         2.0330000E-03    2.0330000E-03  -0.               -0.              -0.  -0.
 16  NH4+            8.6900000E-04    8.6900000E-04  -0.               -0.              -0.  -0.
 17  MISCPLASMA      8.7060000E-04    8.7060000E-04  -0.               -0.              -0.  -0.
 18  MISCREDCELL     3.7505000E-03    3.7505000E-03  -0.               -0.              -0.  -0.
 19  HB4             9.0899999E-03    9.0899999E-03  -0.               -0.              -0.  -0.
 20  *PLASMA         0.              -0.             -0.               -0.              -0.  -0.
END   OF ROWS IN STORAGE

PRINTTABLEAU
```

Table XV--Continued

```
MATRIX  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
  1     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1
  2     0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  1  1  1  0  0  0  0
  3     0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  4     0  0  0  1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1-1  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  1-1  1  0  0  0  0
  5     0  0  0  1  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1  1  1  1  0  0  0  0
  6     0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  7     0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  8     0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
  9     0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 10     0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 11     0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 12     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0
 13     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0
 14     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0
 15     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0
 16     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0
 17     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0
 18     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0
 19     0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  0  0  0  0  0  1
 20     0  0  0  0  0  0  0 1-1-1  1  1  2 2-2-2  0  0-1  1-1  0-2  0**  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
```

```
PRINTMATRIX

MATRIX IN STORAGE

    AIR OUT
  1       O2              -10.940000          1.000 O2
  2       CO2              -7.690000          1.000 CO2
  3       N2              -11.520000          1.000 N2
  4       H2O             -36.600000          1.000 H+          1.000 OH-

    PLASMA
  5       O2                0.000000          1.000 O2
  6       CO2               0.000000          1.000 CO2
  7       N2                0.000000          1.000 N2
  8       H+                0.000000          1.000 H+          1.000 *PLASM
  9       OH-               0.000000          1.000 OH-        -1.000 *PLASM
 10       CL-               0.000000          1.000 CL-        -1.000 *PLASM
 11       NA+               0.002754          1.000 NA+         1.000 *PLASM
 12       K+                0.002758          1.000 K+          1.000 *PLASM
 13       CA++              0.004692          1.000 CA++        2.000 *PLASM
 14       MG++              0.004687          1.000 MG++        2.000 *PLASM
 15       SO4=              0.000000          1.000 SO4=       -2.000 *PLASM
 16       HPO4=             0.000000          1.000 HPO4=      -2.000 *PLASM
 17       UREA              0.000000          1.000 UREA
 18       GLUCOS            0.000000          1.000 GLUCOS
 19       LACTIC            0.000000          1.000 LACTIC     -1.000 *PLASM
 20       NH4+              0.000000          1.000 NH4+        1.000 *PLASM
 21       HCO3-           -21.350000          1.000 CO2         1.000 OH-        -1.000 *PLASM
 22       H2CO3           -32.840000          1.000 CO2         1.000 H+          1.000 OH-
 23       CO3=              6.260000          1.000 CO2        -1.000 H+          1.000 OH-       -2.000 *PLASM
 24       H2O             -39.390000          1.000 H+          1.000 OH-
 25       MISCPR            0.000000          1.000 MISCPL    -10.000 *PLASM

    RED CELLS
 26       O2               -0.000000          1.000 O2
 27       CO2               0.000000          1.000 CO2
 28       N2               -0.000000          1.000 N2
 29       H+                0.000000          1.000 H+
 30       OH-               0.000000          1.000 OH-
 31       CL-               0.000000          1.000 CL-
 32       NA+               2.193399          1.000 NA+
 33       K+               -2.941575          1.000 K+
 34       CA++              2.251790          1.000 CA++
 35       MG++             -0.457703          1.000 MG++
 36       SO4=             -2.000000          1.000 SO4=
 37       HPO4=            -2.000000          1.000 HPO4=
 38       UREA              0.000000          1.000 UREA
 39       GLUCOS            0.000000          1.000 GLUCOS
 40       LACTIC            0.000000          1.000 LACTIC
 41       NH4+              0.000000          1.000 NH4+
 42       HCO3-           -21.490000          1.000 CO2         1.000 OH-
 43       H2CO3           -32.840000          1.000 CO2         1.000 H+          1.000 OH-
 44       CO3=              6.120000          1.000 CO2        -1.000 H+          1.000 OH-
 45       H2O             -39.390000          1.000 H+          1.000 OH-
 46       MISCPR            0.000000          1.000 MISCRE
 47       HB4               0.000000          1.000 HB4
 48       HB4O2           -16.230000          1.000 O2          1.000 HB4
END   OF MATRIX IN STORAGE

CLEAR
```

De Haven and DeLand [21], and a much more complete model in DeLand and Magnier [22]. The human-blood model is shown here particularly to illustrate the intercompartmental relationships.

Generally, the same substance in different compartments is given the same name with a subscript; thus, Na^+_{plasma} and $Na^+_{red\ cell}$. The transfer of a substance from one compartment to another may be regarded as a pseudo-chemical reaction; i.e.,

$$Na^+_{PL} \rightleftharpoons Na^+_{RC} ,$$

and has associated with it an "equilibrium" constant, K_i; or, in the model, c_j:

$$\ln \frac{[Na^+]_{PL}}{[Na^+]_{RC}} = c_{Na^+_{RC}} - c_{Na^+_{PL}} = \Delta c_{Na^+} \qquad (24)$$

(usually, one of the compartmental c_j are zero and therefore, Δc_j is simply referred to as c_j), but the interpretation of this c_j is not obvious thermodynamically. If it is regarded as a free-energy increment--in a sense, the net work done on the species per mole in transferring it from one side of the membrane to the other--at least two problems arise: a) the implication that work has, in fact, been done on the species by an obscure mechanism; and b) in viable biological systems (where this problem arises), an equilibrium does not obtain--rather only a steady movement of species through the system--so that the usual definition of RTlogK as free energy is not applicable. To avoid these

difficulties, the parameter c(J) of Eq. (24) is defined merely as that effective free-energy parameter required under the conditions of the mathematical model to maintain the observed concentration gradient for the substance across the membrane. Again, the c_j is an effective, observed parameter. We will now derive the c_j which simulate the "active cation pumps" between red cell and plasma milieu.

Table XV, a complete printout for this blood model, has several control-card applications. Looking first at the ROWS, the first 19 rows are the components of one liter of the model blood in moles. Note that H_2O is manufactured in the model out of the components H^+ and OH^-, that there is an excess of OH^- owing to: a) the reaction producing HCO_3^- in the model uses CO_2 and OH^-; and b) the exact charge on the various protein components has not yet been determined. The MISCPLASMA is a miscellaneous plasma impermeable component in this simple model including serum albumin; similarly for MISCREDCELL, except that the hemoglobin has been separated as a separate component. Actually, in this simple model, we use the Hill equation for oxygenation of hemoglobin rather than the Adair equation [20]; and the moles of HB4 are really the moles of heme components of the corresponding hemoglobin, as though the hemoglobin had split into monomers. Therefore, there are four times too many moles of HB4 for a liter of blood, which upsets the osmotic regulation of the cell size--an error compensated by the addition of excess impermeable miscellaneous species in the plasma compartment.

Component 20, *PLASMA, having zero value, is a zero net charge restraint placed upon the plasma compartment.

This is accomplished merely by adding the moles of each species times its respective valence and requiring that the sum be zero in the plasma compartment. If the sum of the input component charges are also zero, then the red-cell sum is zero, too; the gases are obviously neutral. Note that each charged species in plasma has an entry from the *PLASMA constraint. The charges assigned here to MISCPR, the miscellaneous impermeable species in plasma, is -10.0 per molecule--an arbitrary number for this example. In a more sophisticated model, the charge on this species would be given by its titration curve and the pH (as in Ex. III, p. 78 ff.).

The constraints 21 through 24 (NA*, etc.) will dictate the amount of each of these species appearing, in this case, in the PLASMA compartment. Thus, we determine from the literature the moles of, say, Na^+ in a liter of blood--which is entered in row 7. Then, determine the Na^+ in the plasma alone of the liter of blood--this amount is entered in constraint 21. When the problem is solved, only the right fraction of the total Na^+ will appear in the plasma--because in the matrix, the component NA* occurs only in the species Na^+_{Plasma}, so that the moles of Na^+_{PL} must, by the constraint, be identical to the fictitious NA* component input.

In the Red-Cell compartment, the free-energy parameters for HCO_3^- are not quite identical to those in plasma owing to a differential solubility of the gases in Red-Cell milieu compared to that in plasma.

The c(j) entries in Na^+, K^+, Ca^{++}, and Mg^{++} are now, in the beginning, zero--as in plasma; therefore, by Eq. (24), the concentration gradient for the cations would be 1:1--

except that the constraints 21 through 24 will upset this ratio for each species.

The MULTIPLIERS indicate that we are going to equilibrate 1 liter of blood with 1000 moles of alveolar gas (see Ex. 1, p. 56 ff. above).

The number of iterations printed from the SOLVE command indicates that the VECTORX used is not very close to the final result. Therefore, we will PUNCHX later to get a better guess for the next machine pass, thus saving computer time.

The first output gives an optimal solution under the constraints; for example,

$$\frac{[NA^+]_{PL}}{[NA^+]_{RC}} = \frac{2.646}{0.4668} = 5.67 \ ,$$

which is far from 1.0 because of the constraint that 76.441×10^{-3} moles of Na^+ appear in plasma, which they do. Also one could check, if he cared to, that the plasma has zero net charge. The distribution of H_2O (see Ref. 20) indicates about 45 percent hematocrit allowing for the specific volume of the proteins in solution. Finally, under the 13.15 percent (100 mm Hg) O_2 concentration in the gases, the hemoglobin is

$$\frac{[HB_4O_2]}{[HB_4O_2] + [HB_4]} = \frac{8.755}{9.09} = 0.963$$

or 96.3 percent saturated. This is obtained using an empirical constant from the Hill equation for the reaction

$$HB_4 + O_2 \rightleftharpoons HB_4O_2 \ , \ K \ .$$

The following control card is PRINTPIE, and in the output following are the values of the Lagrange multipliers (cf. Ref. 3) for each constraint; formally, PIE*RT is the free energy per mole for each component times B, the number of moles of each component--although, again, this interpretation is entirely formal and its interpretation in the physiological system must be used with care.

Next, DELETE is called four times, deleting the fixed constraints 24 through 21 (see DELETE, p. 48); and then SOLVE. The solution now is identical to that before DELETE since the DELETE routine merely substitutes c_j values for the constraint equations.

The PRINTROWS, PRINTMATRIX, and PRINTTABLEAU commands verify that DELETE worked satisfactorily; i.e., a) in ROWS, the last four rows are now missing, and b) in MATRIX, the NA* entries in Na^+_{plasma} is missing but a c_j for Na^+_{plasma} has been obtained. PUNCHMATRIX would now give a deck of cards having the appropriate c_j values punched for the next pass, or the present deck may be altered by adding the c_j values printed in the MATRIX IN STORAGE.

Thus, concerning the intercompartmental relationships, the active cation pumps assumed between the Red Cells and Plasma have been simulated in that the Na^+, K^+, Ca^{++}, Mg^{++} now have the concentration gradients as given in the literature. Also, since the species H_2O in each compartment has the same free-energy parameter, the concentration gradient for the species H_2O is 1:1, i.e., the two compartments are iso-osmotic. Finally, since the input components have a net zero charge, and Plasma has a zero-charge constraint, the Red Cells must also have a zero net charge. This fact, along with the fixed (impermeable) anionic charge

on the various proteins, gives rise to the Gibbs-Donnan gradient. The ratio of Cl^- concentrations in this model is 1.57, which is slightly high as a measure of the normal Gibbs-Donnan gradient.[†]

[†]For further analysis, see Ref. 20.

Chapter IV

PROGRAM SUBROUTINES

GENERAL REMARKS

The instruction deck of CHEMIST consists of a set of FORTRAN IV subroutines that may be subdivided conceptually into three subgroups which, a) solve a chemical equilibrium problem, b) transform and manipulate the data of the chemical model (Control cards), and c) do housekeeping and special tasks. The average user may never become involved with the subroutines in groups a) or c) since they comprise, as it were, the submerged part of the iceberg. The subroutines of group b), which are called in natural language Control cards (see Chap. II above), have the on-line control of data flow, and "perform" for the user the required tasks in his logical order. These Control cards, therefore, provide access to the subroutines although the user need not be aware of this activity. However, situations will inevitably arise in which more detailed information about the subroutines would be advantageous. The following section outlines the interdependencies among the subroutines; and the final section of this chapter, a short description of each subroutine, its calling sequence, and its function. A listing of the entire set has been omitted since it is easily available from a compilation.

SUBROUTINE AND FUNCTION LIST

The following is a list of the subroutines and functions in CHEMIST, along with sublists of the routines and functions called by each.

1. Subroutine ARITH
   - ABS
   - BAR
   - EXIT
   - EXP
   - MATINV
   - PHCALC
   - RCALC
   - SQRT
2. Subroutine BAR(W,WBAR)
3. Subroutine BERROR(BMAX)
   - ABS
4. Function CJACOB(I,J)
   - FIND
5. Subroutine CLOG(W,WBAR)
   - ALOG
6. Subroutine DEL(W,Q)
7. Subroutine DELETE(NODEL)
   - POP
8. Subroutine ERRORS
   - ARITH
9. Subroutine FIND(I,J,IJLOC, IFLAG)
   - EXIT
10. Subroutine IMAGE(KA,L,KB)
11. Subroutine INPUT
    - DELETE
    - EXIT
    - IMAGE
    - JACOBS
    - LOOKUP
    - LP
    - MATRIX
    - OUTPUT
    - PAGE
    - PRINTP
    - PRINTT
    - PUNCHM
    - ROWS
    - SCALEC
    - SOLVE
    - START
    - VECTOR
12. Subroutine JACOBS(I,I2,I3)
    - ARITH
    - IMAGE
    - LIST
    - LOOKUP
    - PART
13. Subroutine LIST
    - IMAGE
    - LOOKUP
14. Subroutine LOOKUP(KA,K,L,LL,KB)
15. Subroutine LP(MON)
    - ALOG
    - AMIN1
    - BAR
    - EXIT
    - FLOAT
    - SIMPLE
16. Subroutine MATRIX(III)
    - EXIT
    - FIND
    - POP
    - PUSH
17. Subroutine MATINV(A,N,B,M,D, W,IP,ISING)
18. Subroutine MOVE(X1,X,N)
19. Subroutine OUTPUT
    - ERRORS
    - IMAGE
20. Subroutine PAGE
21. Function PART(I,J,KIND,KDEP)
    - ALOG
    - CJACOB
    - EXP
    - FIND
22. Function PHCALC(K)
    - ALOG
23. Subroutine POP(IJ)
24. Subroutine PRINTP
    - PAGE
25. Subroutine PRINTT
    - FIND
    - MINO
    - PAGE
26. Subroutine PUNCHM(IPUNCH)
27. Subroutine PUSH(IJ)
    - EXIT
28. Subroutine RCALC
    - FIND
29. Subroutine ROWS(KALTER)
    - EXIT
30. Subroutine SCALEC(KAA,ZZ)
    - EXIT
    - FIND
31. Subroutine SIMPLE(INFLAG,MX,NN, NCT,A,IRO,JC,B,C,KO, KB,P.JH,X,Y,PE,E)
32. Subroutine SOLVE
    - AMAX1
    - AMIN1
    - BAR
    - BERROR
    - CLOG
    - DEL
    - EXP
    - LP
    - MATINV
    - RCALC
    - SQRT
    - SSWTCH
33. Subroutine START
    - IMAGE
34. Subroutine VECTOR(IVI)

For convenience, we arbitrarily divide the subroutines and functions into the three categories mentioned above: a) solve subroutines, b) Control card subroutines, and c) subsidiary housekeeping subroutines and special functions. Communication of data into and out of the subroutines is accomplished by the following block common, equivalence, and integer statements:

```
      C
      C ISOIONIC POINT CALCULATION                                    NZS20220
      C
   47     1 DH=.001
   50       SUM1=  -X(8)-X(10)+X(11)+X(13)+X(15)-X(18)+X(19)
   51       DO 200 I=1,10
   52       BBB(1,1)=BBB(1,1)+DH
   53       CALL ROWS(-1)
   54       CALL SOLVE
   55       IF(IERROR.NE.1) CALL EXIT
   60       SUM2=  -X(8)-X(10)+X(11)+X(13)+X(15)-X(18)+X(19)
   61       IF(ABS(SUM2).LT..00001) GO TO 201
   64       DH= ( (SUM2-0.) / (SUM1-SUM2) ) *DH
   65       SUM1=SUM2
   66   200 CONTINUE
   70   201 CALL OUTPUT
   71       WRITE(6,300) NR(1,1),NR(1,2),BBB(1,1)
   72   300 FORMAT(1X,2A6,3H = 1PE20.8)
   73       WRITE(6,301) KN(11),X(11),KN(8),X(8),KN(13),X(13),KN(10),X(10),
           X   KN(15),X(15),KN(18),X(18),KN(19),X(19)
   74   301 FORMAT(1H0,16X,1H+,18X,1H-/(6X,A6,1PE13.5,1X,A6,1PE13.5/))
      C
      C REPEAT, USING REDUCED VOLUME OF SOLVENT (1.0 LITER OF SOLUTION)
   75       CALL INPUT
   76       GO TO 1
   77     7 CALL EXIT
  100       END
```

In addition, each subroutine or function may have its own peculiar dimension statements and arguments.

SOLVE SUBROUTINES

BAR	MATINV
BERROR	RCALC
CLOG	SIMPLE
DEL	SOLVE
LP	

This collection of FORTRAN IV subroutines may be used for solving chemical equilibrium problems as described in Ref. 3. The calling sequence is merely the FORTRAN statement CALL SOLVE, or, using Control cards, just the card SOLVE.

The data that must be input before CALL SOLVE is executed consist of the following:

COMMON Location	Quantity
IV(1)	M, the number of constraints.
IV(2)	MEND, = M + NCOMP.
IV(3)	NCOMP, Number of compartments.
IV(4)	N or NTOT, Number of unknown variables.
IV(5)	Number of the input unit.
IV(6)	Number of the output unit.
IV(7)	Print flag: -1 = minimal amount of messages; 0 = one message per iteration step; +1 = all messages.
IV(9)	Maximum number of iterations to be allowed.
B(i)	B_i, i=1,2,...,M.
X(j)	y_j, j=1,2,...,N, where y_j is the initial estimate of the solution. If no estimate is available, set X(j) = 0, all j.
C(j)	c_j, j=1,2,...,N, the free-energy parameters.
"A(i,j)"	See MATRIX subroutine for how the matrix entries a_{ij} are stored in common.

In addition, all species in one compartment must have consecutive subscripts. That is, species $1,2,3,\ldots,k_1$ must be in compartment 1; species $k_{1+1}, k_{1+2},\ldots,k_2$ must be in compartment 2; ...; and species $k_{p-1+1}, k_{p-1+2},\ldots,k_p$ must be in compartment p. These k's are communicated to the subroutines by setting

$$\begin{aligned} KL(1) &= 1 \\ KL(2) &= k_{1+1} \\ KL(3) &= k_{2+1} \\ &\vdots \\ KL(p) &= k_{p-1+1} \\ KL(p+1) &= k_{p+1} \ . \end{aligned}$$

In other words, KL(k) is the number of the first component in compartment k, and KL(p+1) is equal to n+1.

The above are the only numbers that need to be set so that CALL SOLVE will solve the chemical equilibrium problem. However, in order that the program can write messages (in cases of infeasibility, etc.), names for the rows, species, and compartments may be input:

COMMON Location	Quantity
NR(I,1), NR(I,2)	Two-word row name for row I.
KN(J)	One-word component name for component J.
NAM(K,1), NAM(K,2)	Two-word compartment name for compartment K.

In addition, TOL(1) through TOL(5) are tolerances used by the program. If they are zero when the program is entered,

they are set by the subroutines to nominal values. These may also be set by the user of the subroutines; if so, the nominal values will not be set in the subroutines. The tolerances are the following:

Tolerance	Nominal Value	Meaning[†]
TOL(1)	0.01	ϵ in step 3 of the first-order method.
TOL(2)	10^{-5}	δ in step 4 of the second-order method.
TOL(3)	10^{-12}	Minimum value any x_j is allowed to have.
TOL(4)	10^{-6}	Minimum starting value that any component will have is the lesser of TOL(4) and $\frac{1}{2}y_{n+1}$.
TOL(5)	10^{-8}	Problem is assumed to be degenerate if any S_k becomes less than TOL(5).

With all the above as input, the statement CALL SOLVE will attempt to solve the chemical equilibrium problem. If, upon completion of this attempt, a solution is obtained, the cell IV(10) will contain a 1, and the following data will be in storage:

COMMON Location	Data
X(i)	x_i, i=1,2,...,n (the solution).
XBAR(k)	S_k, k=1,2,...,p.
PIE(i)	π_i, i=1,2,...,m.
XMF(i)	$\hat{x}_i$, i=1,2,...,n.

[†]See Ref. 3.

If IV(10) is not 1, the subroutines have failed to solve the chemical equilibrium problem. The reason for this failure is written on output unit IV(6). In such a case, X(j) will contain the latest value of these quantities.

* * * * *

There are nine subroutines in the set used for the solution of the chemical equilibrium problem. A brief description of these subroutines follows. (For a complete description see Ref. 3.)

1. Subroutine SOLVE, the master subroutine, is divided into four functional segments (each of which calls other subroutines for specific tasks):
   a) The projection and linear programming routines for obtaining the initial solution.
   b) The first-order method.
   c) The second-order method.
   d) Output messages.
2. Subroutine BAR(W,WBAR) calculates the S_k.
3. Subroutine BERROR(BMAX) calculates

$$g_i = b_i - \sum_{j=1}^{N} a_{ij}x_j , \quad i=1,2,\ldots,M .$$

4. Subroutine DEL(W,Q) sets

$$w_j = \sum_{i=1}^{m} a_{ij}q_i , \quad j=1,2,\ldots,n .$$

5. Subroutine RCALC calculates the r_{il} array.

6. Subroutine CLOG(W,WBAR) computes

$$\alpha_j = c_j + \log \hat{x}_j \ , \quad j=1,\ldots,n \ .$$

7. Subroutine LP(MON) sets up the linear programming problems.

8. Subroutine SIMPLE(INFLAG,MX,NN,NCT,A,IRO,JC,B,C,KO, KB,P,JH,X,Y,PE,E) solves the linear programming problems. Information is communicated to this routine via a calling sequence rather than by COMMON (as in subroutines 1-7). All dimensions are dummy statements.

9. Subroutine MATINV(A,N,B,M,D,W,IP,ISING) solves simultaneous equations. As in SIMPLE, no COMMON is used. The dimension of A in MATINV should agree with that of R (not A) in SOLVE. All other dimensions are singly subscripted.

* * * * *

Subroutines 1-7 above all have a COMMON statement (labeled /SLVE/) which should be the same in all seven. The dimensions of the variables in this COMMON statement may be set to the values for the largest problem to be solved. With M, MEND, NCOMP, and N as previously defined, these dimensions must be at least:

Symbol	Minimum Dimension
IV	30
TOL	20
NR	(M,2)
B	M
KN	N
X	N+1
C	N+1
JCOMP	N+1
KL	NCOMP+1
NAM	(NCOMP,2)
PIE	MEND
V1,V2,V3,V4	MEND
XMF	N
X1,X2,X3	N+1
XBAR	NCOMP
R	(MEND,MEND)

CONTROL-CARD SUBROUTINES

DELETE	PRINTT
JACOBS	PUNCHM
MATRIX	ROWS
OUTPUT	SCALEC
PAGE	VECTOR
PRINTP	

These 11 subroutines are the master routines for corresponding Control cards. In each case, one or more Control cards call these subroutines for execution of the macro-instruction of the card. Alternatively, these subroutines may be called from the (FORTRAN) MAIN routine; consequently, even though the actions of the Control cards have been described in Chap. II above, we show here the calling sequences and the data required for each subroutine. Since it is unlikely that these subroutines will be called out of the context of CHEMIST, the requirements are given in terms of the data formats of that problem. For this

purpose, we assume the existence of a valid model of a chemical system having a feasible solution (e.g., any of the examples described above in Chap. III would serve). We also assume that either subroutine START or CLEAR has been previously called.

1. Subroutine DELETE(NODEL) performs a Lagrangian delete NODEL times. The result is described under Control card DELETE (p. 48 above). The Control card DELETE calls subroutine DELETE(1) and so must be used NODEL times to be equivalent to the FORTRAN statement CALL DELETE(NODEL).

2. Subroutine JACOBS(I1,I2,I3) computes and prints partial derivatives $\partial u/\partial v$, where u (the dependent variable) may be components of the vectors X,XMF,XBAR, or pH; and v (the independent variable) may be components of the vectors B, C, or K(K = expC).

The program maintains two lists of specifications established and altered under user control: suppressed compartments, and selected species.

Printing of partial derivatives of all quantities (mole numbers, total mole number, mole fraction, and pH) associated with compartments which are suppressed at the time when a printout occurs will be omitted. Also, only the partial derivatives with respect to the c's or k's of selected species will be printed. For this reason, the compartments are also regarded as species. (Thus, for the ℓth compartment, $\partial\bar{x}_\ell/\partial\bar{c}$ or $\partial\bar{x}_\ell/\partial\bar{k}$ are symbols meaning, respectively, the rate of change of $\bar{x}_\ell$ when each and every c_j in compartment ℓ is incremented additively an equal amount or if each k is multiplied by the same number.)

Printing of partial derivatives with respect to those b's which are exactly equal to zero are omitted; all other partial derivatives with respect to b's are printed under the b-derivative option.

Subroutine JACOBS is called by the FORTRAN statement CALL JACOBS(I1,I2,I3) where:

I1 specifies the desired independent variables as follows:

- I1 = 0 Specifies B's which are not zero.
- I1 = 1 Specifies c's associated with the selected species.
- I1 = -1 Specifies k's associated with the selected species.

I2 specifies the desired dependent variable as follows:

- I2 = 0 Specifies mole numbers of every species in every unsuppressed compartment and the total mole numbers and pH of every unsuppressed compartment.
- I2 = 1 Specifies the mole fractions of every species in every unsuppressed compartment, and the pH of every unsuppressed compartment.
- I2 = 2 Specifies the total mole numbers of every unsuppressed compartment.
- I2 = -1 Specifies the pH's of every unsuppressed compartment.

I3 specifies whether or not printing is to occur, and whether or not data cards which will create a new list of selected species are to be read:

- I3 = -1 Creates a list of selected species and/or suppressed compartments by reading data cards. If there is a list of selected species already in existence, it will be erased and a new list formed. No printing will occur. When I3 = -1, the values of I1 and I2 are irrelevant.

I3 = 0 Creates a list as discussed above, but will also compute and print the partial derivatives of every independent variable (specified by I1) with respect to every dependent variable (specified by I2) for the species on the list.

I3 = 1 Computes and prints partial derivatives discussed above; but instead of creating a new list of selected species, will use the list previously created. Use 1 for B JACOBS.

Care must be taken when using JACOBS more than once in the same run not to erase inadvertently the list of species by calling for a list to be created more than once. A second list in the same run may be created, if desired, by calling for it--but another set of data cards must be included. However, compartment-suppression cards may be added to the list without erasing the species list.

Compartment Suppression. All compartments start out in an unsuppressed state. To suppress a compartment, a data card is added (as described below) with a 1 in column 20. Similarly, a compartment may be unsuppressed by adding a data card with a 0 in column 20. I3 must, of course, be -1 or 0 to read these cards. If more than one card is read in for a given compartment, the last one read dominates.

Data Cards for JACOBS. Cards for both lists, the selected species and the compartment suppression, may be inserted together, with formats as follows: columns 1-12 have the compartment name, left-justified; columns 13-18, the species name, left-justified; and column 20, the compartment-suppression indicator. Table XVI gives allowable data card variations.

Table XVI

CONTROL CARD VARIATIONS FOR JACOB

Columns 1-12	Columns 13-18	Column 20	What it does
1) Compartment name	Species name		Selects this species for list.
2) Compartment name	C-BAR		Selects this compartment, as a separate species, for list.
3) Compartment name	ALL		Selects this compartment, as a separate species, and all species in this compartment, for the list.
4) Compartment name		1	Suppresses this compartment.
5) Compartment name		0	Unsuppresses compartment.
6) ALL		1	Suppresses all compartments.
7) ALL		0	Unsuppresses all compartments.
8) END			Ends reading of cards for list.

The last card in each list must have END in columns 1-3. The first card of format 1), 2), or 3) above read in any list erases the previous list.

Printing Format. The partial derivatives are printed in cycles of arrays, each cycle being 8 (or less in the case of the last cycle) columns wide and up to N (the number of species in the problem) rows long. The column headings are the names of the selected species (independent variables selected) and the rows are identified by either compartment or species name or pH. Thus, for each of the selected species, there is printed in a column (for all unsuppressed compartments) the partial derivatives of the compartment treated as a separate species with respect to the selected species, the pH (if any) of the compartment, and then all the partial derivatives of the species in that compartment with respect to the selected species.

Compatibility. The Control card JACOB has the effect of calling JACOBS (0,0,1). Control card MINIJACOB calls JACOB (0,2,1).

Partial Derivatives. The partial derivatives are calculated using the following equations [3]:

$$\frac{\partial x_k}{\partial b_i} = d_k \, x_k \sum_{\ell=1}^{M} a_{\ell k} \, \gamma'_{\ell i}$$

where:

d_k = +1 for k < N, -1 for k > N;

x_k = moles of species k;

$a_{\ell k}$ = matrix element at row ℓ, column k;

$\gamma'_{\ell i}$ = element of matrix R^{-1} (see RCALC, p. 104 above) at row ℓ, column i;

b_i = moles of component i.

$$\frac{\partial x_k}{\partial c_j} = d_j \, d_k \, x_j \, x_k \sum_{s=1}^{M} \sum_{t=1}^{M} a_{sj} \, a_{tk} \, \gamma'_{st} - \delta_{jk} \, d_j \, x_k$$

where:

δ_{jk} = kroenecker delta.

$$\frac{\partial x_k}{\partial a_{ij}} = -\pi_i \frac{\partial x_k}{\partial c_j} - x_j \frac{\partial x_k}{\partial b_i}$$

where:

π_i = ith Lagrange multiplier, PIE(I).

The partial of $\hat{x}$ with respect to the independent variable is obtained from the identity

$$\dot{\hat{x}}_\ell = \hat{x}_\ell \left(\frac{\dot{x}_\ell}{x_\ell} - \frac{\dot{\bar{x}}_{[\ell]}}{\bar{x}_{[\ell]}} \right) ,$$

where the dot means derivative and the subscript in square brackets indicates that $\bar{x}_{[\ell]}$ is the sum of the moles of all species in the compartment containing x_ℓ.

3. Subroutine MATRIX(III) is called by several Control cards (or by the FORTRAN statement CALL MATRIX(III), and the argument III determines the action taken according to the following equivalence:

III	CONTROL
-1	ALTERA
0	MATRIX
1	ALTERC
2	ADDC.

All four of these controls are Control cards and the action taken is described in Chap. II above. The Control card MATRIX (i.e., CALL MATRIX(0)) reads and stores the names of the compartments and species as well as the data array A(I,J). These data must be available in storage before using the other options of the argument.

The data in the matrix array A(I,J) is not stored in COMMON in an I x J matrix. Since this matrix is sparse, considerable space has been saved by storing the data in the three smaller blocks:

Name	Dimension	Meaning
AIJ	460	Coefficient of matrix entry, a_{ij}
IROW	460	Row number for entry
JCOL	460	Column number for entry.

The matrix coefficients are stored in the order in which they are read. If more than one card is used for one species, it must immediately follow the first card. The free-energy parameter from the first card is stored.

The coefficient for the entry goes into the array AIJ, the row number into IROW, and the column number into JCOL.

The arrays for the sample Soda-Pop problem (see Ex. 2, PRINTTABLEAU, p. 81 above) would look as follows:

AIJ(1) = 1.0	IROW(1) = 1	JCOL(1) = 1
AIJ(2) = 1.0	IROW(2) = 2	JCOL(2) = 2
AIJ(3) = 1.0	IROW(3) = 3	JCOL(3) = 3
AIJ(4) = 1.0	IROW(4) = 4	JCOL(4) = 4
AIJ(5) = 1.0	IROW(5) = 1	JCOL(5) = 5
AIJ(6) = 1.0	IROW(6) = 2	JCOL(6) = 6
AIJ(7) = 1.0	IROW(7) = 3	JCOL(7) = 7
AIJ(8) = 1.0	IROW(8) = 8	JCOL(8) = 8
AIJ(9) = 1.0	IROW(9) = 4	JCOL(9) = 9
AIJ(10) = 1.0	IROW(10) = 5	JCOL(10) = 9
⋮	⋮	⋮
AIJ(23) = 1.0	IROW(23) = 4	JCOL(23) = 18
AIJ(24) = -2.0	IROW(24) = 5	JCOL(24) = 18
AIJ(25) = 1.0	IROW(25) = 11	JCOL(25) = 19
AIJ(26) = 1.0	IROW(26) = 5	JCOL(26) = 19
AIJ(27) = 1.0	IROW(27) = 11	JCOL(27) = 20

Manipulation of Matrix Data. Since the matrix information is stored in three arrays, it cannot be addressed directly. If a particular a_{ij} coefficient is needed, the arrays are first examined for a value that corresponds to the i and j subscripts. If there is an a_{ij} entry, it must be located in the array. If it does not exist, then this information is required. Also, it is sometimes convenient to know the first and last locations in the array for a given species or for a given compartment.

If one of the Control cards ALTERA, ALTERC, or ADDC is used, a particular a_{ij} coefficient must be found. If it is not in the matrix, the program must find the place to insert the value that is to be altered into the data arrays. A new coefficient for species J would go after the last entry for that species and ahead of the first entry for the next species. This means that all entries must be moved down one cell starting with the next species to make room for the new a_{ij}. Conversely, if an a_{ij}

coefficient in the matrix is to be changed to zero, the code will eliminate it by moving all entries that follow by one cell.

Three subroutines are available to do the manipulation. FIND will look for the entry a_{ij} and return the appropriate information; it will also look for the beginning and end of a species. The locations for the compartment are found by looking for the ranges of the first and last species in the compartment (see FIND below). PUSH will move the matrix down one slot (see PUSH below); it gets the starting location from FIND. POP will move the matrix up one slot (see POP below); it also gets the starting location from FIND.

4. Subroutine OUTPUT is called by the Control card OUTPUT or by the FORTRAN statement CALL OUTPUT. (The action is described in Chap. II above, p. 47.) Normally the vector X and the vector XMF are printed as shown in the Chap. III examples; but the vector X1 may also be printed as a third line of output by first filling the vector with the data to be printed and then setting IV(23) = 1 before calling OUTPUT.

The subroutines called by OUTPUT and ERRORS (which calls ARITH), which prints the errors and the headings OPTIMAL or NOT OPTIMAL SOLUTION (see example, p. 65); and IMAGE, which sets up some of the captions for printing.

5. Subroutine PAGE is called by the Control card EJECT or by the FORTRAN statement CALL PAGE. Its effect is to skip the printer to the top of the next sheet and to print the last read TITLE.

6. Subroutine PRINTP is called by the Control card PRINTPIE or by the FORTRAN statement CALL PRINTP. Its effect is described in Chap. II above (p. 51).

7. Subroutine PRINTT is called by the Control card PRINTTABLEAU or by the FORTRAN statement CALL PRINTT. Its effect is described in Chap. II above (p. 50).

8. Subroutine PUNCHM(I) is called by the Control card PUNCHMATRIX, or by the FORTRAN statement CALL PUMCHM(I). Its effect is to print (I=0) or to punch (I=1) the matrix data array in the input format.

9. Subroutine ROWS(KALTER) is called by several Control cards or by the FORTRAN call statement CALL ROWS(KALTER). The value of the argument KALTER determines the action taken according to the following equivalence:

KALTER	CONTROL
-3	PUNCHROWS
-2	PRINTROWS
-1	(Update B)
0	ROWS
+1	ALTERB
+2	ADDB
+3	B

"Update B" is not a Control card, but the other six controls in the above list are. (Their action is described above in Chap. II.) Update B (i.e., CALL ROWS (-1)) updates the vector B according to

$$B(I) = \sum_{J} BBB(I,J)*BMULT(J) \quad , \quad \text{for all } I \ ,$$

and consequently the BMULT(J) must be known before CALL ROWS(-1). Similarly, if the BMULT(J) are changed in FORTRAN (e.g., in the MAIN routine) CALL ROWS(-1) must be used to update B.

Normally, the MULTIPLIERS Control card computes B; and ROWS, ALTERB, ADDB, and B also update B automatically. Since B is the matrix product BBB·BMULT, both sets of data are required for correct evaluation of B; and since the ROWS data array is read before the BMULT(J) data, the result may temporarily be irrelevant. Conversely, the MULTIPLIERS Control card (as well as ALTERB, ADDB, and B) requires the ROWS data array for execution since it does update B.

10. Subroutine SCALEC(KAA,ZZ) is called by the Control card SCALEC (followed by data; see SCALEC, Chap. II, p. 35); or by the FORTRAN statement CALL SCALEC(KAA,ZZ), where KAA is the name of the compartment to be scaled (first six alphanumeric symbols of the name only, including blanks) and ZZ=SCALE -1.0--i.e., if the compartment KAA is to be multiplied by SCALE (a real number), ZZ is set at SCALE -1.0 since the result of SCALEC is added to the present value (see p. 35 ff. above). More than one compartment may be listed as data cards after the SCALEC Control card (all followed by an END card), but CALL SCALEC(KAA,ZZ) must be used once per compartment.

11. Subroutine VECTOR (IVI) is called by two Control cards or by the FORTRAN statement CALL VECTOR (IVI) according to the following equivalence:

IVI	Control
0	VECTORX
1	PUNCHX

The action of these two Control cards is explained in Chap. II above.

SUBSIDIARY SUBROUTINES AND FUNCTIONS

ARITH	LOOKUP
CJACOB	MOVE
ERRORS	PART
FIND	PHCALC
IMAGE	POP
INPUT	PUSH
LIST	START

JOHN01
GOALN8

This is a miscellaneous collection of subroutines and functions which are not called either by SOLVE or directly by a Control card. They are called only by FORTRAN call statements or--in the cases of CJACOB, PART, and PHCALC--by FORTRAN function statements.

1. Subroutine ARITH, called by the FORTRAN statement CALL ARITH, recomputes the vectors XBAR, XMF, and R^{-1}; and then computes the current value of the free-energy functions FE=PIE·B and FE2=RT·FE--where the dot is vector dot product, and the vector pH. It then computes ERMA and ERMB, and sets the optimum solution toggle IOPT (see Table II, pp. 27-28 above).

2. Function CJACOB(I,J) is called by JACOBS to compute the partial derivative of x(I) with respect to C(J).

It may also be used directly as a function, but the inverse of the matrix R must be in storage. For this purpose, the inverse may be obtained by the statement CALL ARITH.

3. Subroutine ERRORS is called by the FORTRAN statement CALL ERRORS. This subroutine calls ARITH and then prints the resulting messages at the beginning of the OUTPUT printing (see Subroutine OUTPUT, p. 114 above).

4. Subroutine FIND(I,J,IJLOC,IFLAG) locates an element of the matrix A(I,J) in the data arrays AIJ(IJ), IROW(IJ), and JCOL(IJ) using the following four arguments:

I is set to the given row number, i (or zero).

J is set to the given column number, j.

IJLOC is set by FIND to a location (subscript IJ) within the matrix data arrays.

IFLAG is set by FIND to zero or one if $I \neq 0$. If $I = 0$, FIND will set IFLAG to a location.

The subroutine does one of two things, depending on the value of I:

1) If $I \neq 0$, search the arrays of the matrix data for the entry for a_{ij}. If it is in the matrix, set

IJLOC = IJ

IFLAG = 0 .

This means that

IROW(IJ) = I

JCOL(IJ) = J

AIJ(IJ) = a_{ij} .

If there is no entry in the matrix for a_{ij}, set

IJLOC = IJ

IFLAG = 1 ,

where IJ is the location for the first entry of the next species.

2) If I=0, search the arrays of the matrix data for the beginning and end of column J. Then set

IJLOC = L

IFLAG = LL .

This means that all the a_{ij} entries for species J are located in the array AIJ from AIJ(L) to AIJ(LL) inclusive.

This second option can be used to find the range for all the entries of a given compartment K. Call the subroutine twice. The first time, set J+KL(K), which is the first species in the compartment. The calls will be as follows:

```
CALL FIND (0,KL(K),KTA,KTB)
CALL FIND (0,KL(K+1)-1,KTC,KTB)
```

This means that all of the a_{ij} entries for the compartment are located in the array AIJ from AIJ(KTA) to AIJ(KTB).

Note that the third parameter in the two calling sequences must have a different name.

5. Subroutine IMAGE(KA,L,KB), called by CALL IMAGE(KA,L,KB), merely stores the name in KB(L) in location KA.

6. Subroutine INPUT is the master Control routine for the Control cards. From the user's view, INPUT is always called by the FORTRAN statement CALL INPUT, and the action is always apparently to transfer control to subsequent Control cards in the data deck. However, control of the machine pass actually remains in INPUT where the statement

```
701 READ(NIT,71)(KA(K),I=1,12)
```

reads each Control card in turn; and, using subroutine LOOKUP, identifies the card and then calls the proper subroutines for its execution. Upon completion of each Control card a GO TO statement transfers back to read the next card.

7. Subroutine LIST, called by JACOBS, handles the list structuring of compartments, species, and parameters as described above under Subroutine JACOBS (p. 106 ff.).

8. Subroutine LOOKUP(KA,K,L,LL,KB) looks in the array KB(I) from the locations of I=L to I=LL for the first word that is equal to the word in KA. If one is found, set K=I. If KB(I) does not contain the word in KA, set K=LL+1.

- L can be greater than one, but it must not be greater than LL.
- LL must be within the range of KB array and should not be less than L.
- KB is an array name, or it may be an alpha-numeric list.

The following uses are legal:

```
CALL LOOKUP (LC,J,1,5,KN)
CALL LOOKUP(LD,J,KL(K),KL(K+1)-1,KB)
CALL LOOKUP (KA,K,1,3,18HINPUTSOUTPUTFINISH)
CALL LOOKUP (KA,K,2,4,30HFIRST SECONDTHIRD
     FOURTH FIFTH SIXTH)
```

Using the last CALL LOOKUP statement as an example, the following demonstrates how it works:

```
If KA is THIRD    K = 3,
If KA is SAMPLE   K = 5,
If KA is FIRST    K = 5.
```

FIRST is in the list, but the CALL asks the subroutine to look at words 2, 3, and 4 only.

9. Subroutine MOVE(X1,X,N) sets X(J)=X1(J) for J=1,N.

10. Subroutine PART(I,J,KIND,KDEP) computes partial derivatives, $\partial u/\partial v$, according to the following values of the arguments:

I = the independent variable: the row or species number, or in the case of compartments, the negative of the compartment number.

J = the dependent variable; the species number or the negative of the compartment number.

KIND = the kind of independent variable:

KIND = 0 for derivatives with respect to b
KIND = 1 for derivatives with respect to c
KIND = -1 for derivatives with respect to K = exp c

KDEP = the kind of dependent variable:

KDEP = 0 for mole number or total mole number (x or $\bar{x}$)
KDEP = 1 for mole fraction ($\hat{x}$)
KDEP = -1 for pH .

Compartment numbers, as described above, are differentiated from species numbers by a minus sign. The input for a pH must have a compartment number (negative) for J.

When PART is called directly, ARITH must be called first in order to make R^{-1} available.

The c's and K's Associated with a Compartment. There is a generalized definition of the free-energy parameter (and K=exp C) which applies to compartments as a whole. A precise pragmatic statement is that for any dependent variable, u,

$$\frac{\partial u}{\partial c} = \sum_j \frac{\partial u}{\partial c_j}$$

where the summation extends over all species in the compartment with which c is associated.

Also,

$$\frac{\partial u}{\partial k} = \frac{\partial u}{\partial c} e^{-c_j} = \frac{\partial u}{\partial c} .$$

11. Function PHCALC(K) computes the pH for compartment K. The normal way to use it would be PH(K) = PHCALC(K). The formula for the computation of the pH is as follows:

$$pH = - \frac{\log(x_{H^+} * \text{ALITER} * \text{BLITER})}{\log 10} ,$$

where the product of the constants ALITER times BLITER times the concentration of H^+ will give the correct pH at temperature $37^{\circ}C$.†

12. Subroutine POP(IJ) eliminates the matrix data entry AIJ(IJ) that is equivalent to a_{ij}. The value of IJ is normally obtained from a CALL FIND(I,J,IJ,IFLAG) statement. To remove the entry from the matrix, the subroutine shifts the data in the arrays from

AIJ(IJ+1) to AIJ(NAIJ)
IROW(IJ+1) to IROW(NAIJ)
JCOL(IJ+1) to JCOL(NAIJ)

---

†Cf. Ref. 5, pp. 348-349.

up by one position so that it is now located in

```
AIJ(IJ) to AIJ(NAIJ-1)
IROW(IJ) to IROW(NAIJ-1)
JCOL(IJ) to JCOL(NAIJ-1) .
```

Set the values of

```
AIJ(NAIJ) = 0
IROW(NAIJ) = 0
JCOL(NAIJ) = 0
```

and then reduce NAIJ by one since we have eliminated an entry.

13. Subroutine PUSH(IJ) adds a matrix data entry, a_{ij}, to the arrays AIJ(IJ), IROW(IJ), and JCOL(IJ). The value of IJ is normally obtained from a CALL FIND(I,J,IJ, IFLAG) statement. Adding an entry, PUSH moves all of the entries down by one position so that the data that was originally in

```
AIJ(IJ) to AIJ(NAIJ)
IROW(IJ) to IROW(NAIJ)
JCOL(IJ) to JCOL(NAIJ)
```

is now located in

```
AIJ(IJ+1) to AIJ(NAIJ+1)
IROW(IJ+1) to IROW(NAIJ+1)
JCOL(IJ+1) to JCOL(NAIJ+1)
```

and set NAIJ = NAIJ+1.

14. Subroutine START, normally called once in the MAIN routine, clears all of COMMON and sets the nominal values for all parameters. Since the Control card CLEAR calls START, its action is the same. The parameters set are:

```
      NIT =   5
      NOT =   6
     MAXM =  60
     MAXN = 169
   MAXAIJ = 460
     MAXP =  25
    MAXMD =  75
 BMULT(1) =   1.0
   ALITER =  55.139673
       RT = 616.27403 .
```

15. Subroutine JOHN01(I,I2,I3) is an example of a special-purpose routine which simulates the movement of species into and out of a biological system. Components of one hour's urine are added to the system and, simultaneously, the previous hour's production of urine is deducted. The calculation will thus follow the time course of events in the biological system [2].

16. Subroutine GOALN8(METH,LOCIND,KAIK,KNDIND, LOCDEP,KNDDEP,GOAL,IGRUP) is a modification of the previously described PHSOLVE (see Chap. III, p. 74 above) with the following added characteristics:

1) The A(I,K) values can be used as an independent variable to reach a goal (dependent variable value).
2) More than one species can be combined to form a compound, which will be the independent variable. Each component of the compound will be incremented equally as the routine seeks a desired goal.
3) The routine and all its goals can be defined from any other routine or from a macro in the data deck plus the appropriate data cards.
4) The maximum iterations allowed in attempting to reach a goal is a variable.
5) The printed output generated by the routine is segmentally suppressible.

6) The computed step size (increment) of a C(I) can be attenuated if the routine oscillates or blows up.

The arguments of the subroutine have the following definitions:

METH--Indicates how information will be supplied to the subroutine.

METH = 0 The program will start reading data cards until an END is read into columns 1-3. One data card is used for each independent variable.

METH = 1 The program expects only one dependent and one independent value whose characteristics are specified by the remaining arguments of the CALL GOALN8 (1 , - , - , - , - , - , - , -) statement.

METH = 2 The program stores this set of dependent and independent arguments until the complete set of arguments has been supplied by a series of call statements.

METH = 3 Indicates that all the arguments have been supplied by call statements and the routine should start iterating in an attempt to reach the specified goals.

LOCIND--Specifies a location of the independent variable. This is the location of the Ith component in ROWS, the Ith "c" value, or the Ith row of the A(I,K) matrix.

KAIK--Also specifies the location of the independent variable: the Kth column of the "BBB" matrix or the Kth column of the A(I,K) matrix. Enter a "1" for c(j) goal.

KNDIND--The type of independent variable:

0 = BBB(I,J)
1 = c(K)
2 = A(I,K) .

LOCDEP--Specifies the location of the dependent variable, either its compartment or species number. Compartments are to be specified by negative integer; i.e., -1, -2, etc.

KNDDEP--Indicates the type or units of the independent variable:

-1 = pH
0 = moles
1 = mole fractions .

GOAL--The desired goal value.

IGRUP--Specifies the Ith independent group to be used to reach the Ith goal.

A group could be:

1) The moles of a single species, as H^+.
2) The combined moles of more than one species to form a "compound" such as $H^+ + Cl^-$.
3) A combined set of equally incremented C(I)'s or A(I,K)'s.

The maximum number of components to a compound is six. Each compound must have the same IGRUP number, but each group must be sequentially incremented by one. The maximum total number of goals, counting each component of a compound, is 19. The partial derivative of a dependent variable with respect to a compound is the sum of the derivatives with respect to each component of the compound. Thus, since a_{ij} may be negative, the sequence number is written with a minus sign as required.

The set of values on the END card or the CALL card with METH = 3 can be used to suppress undesired output:

LOCIND > 0 Suppresses the output of the input table.

KAIK > 0 Suppresses the output of the partials with respect to A(I,K).

KNDIND = 1	Suppresses the iterated output of the incremented independent variables. The final increment is printed.
KNDIND = 2	Suppresses all output of the incremented independent variables.
LOCDEP > 0	Suppresses the matrix of partials used to determine the independent variable increment.
KNDDEP > 0	Suppresses the set of partials that are summed to form the partial of a compound.
IGROUP > 0	This number will determine the number of cycles the program goes through to reach a goal. A zero or default option implies a maximum of five cycles.
GOAL > 0.0	Specifies the attenuation factor for a C(I), or A(I,K) value. A zero or default option implies a 0.8 attenuation.

REFERENCES

1. The Scientific Papers of J. Willard Gibbs, Vol. I, Thermodynamics, Dover, New York, 1961.

2. De Haven, J. C., and N. Z. Shapiro, Intrinsic Control of Body Fluid and Electrolyte Distribution and Urine Formation, The RAND Corporation, RM-4609-PR, July 1965.

3. Clasen, R. J., The Numerical Solution of the Chemical Equilibrium Problem, The RAND Corporation, RM-4345-PR, January 1965.

4. White, W. B., S. M. Johnson, and G. B. Dantzig, Chemical Equilibrium in Complex Mixtures, The RAND Corporation, P-1059, October 1957; also published in J. Chem. Phys., 28 (1958), 751-755.

5. Dantzig, G. B., J. C. De Haven, I. Cooper, S. M. Johnson, E. C. DeLand, H. E. Kanter, and C. F. Sams, "A Mathematical Model of the Human External Respiratory System," Perspectives in Biol. & Med., 4 (1961), 324-376.

6. Shapiro, N. Z., and L. S. Shapley, Mass Action Laws and the Gibbs Free Energy Function, The RAND Corporation, RM-3935-1-PR, September 1964.

7. Shapiro, N. Z., Conditions for a Homogeneous Mixture to be Ideal, The RAND Corporation, RM-3677-PR, June 1963.

8. -----, A Generalized Technique for Eliminating Species in Complex Chemical Equilibrium Calculations, The RAND Corporation, RM-4205-PR, September 1964.

9. -----, "Variations in the Parameters of a Chemical Equilibrium System" (in preparation).

10. Shapiro, N. Z., and L. S. Shapley, On Membrane Equilibria, The RAND Corporation, RM-4464-PR, July 1966.

11. De Haven, J. C., and N. Z. Shapiro, On the Formation of Urine, The RAND Corporation, P-3254, November 1965.

12. Shapiro, N. Z., and L. S. Shapley, "Mass Action Laws and the Gibbs Free Energy Function," J. Soc. Indust. App. Math., Vol. 13, No. 2, June 1965.

PRECEDING PAGE BLANK

13. Assali, N. S., and N. Z. Shapiro, "Oxy-Hemoglobin Dissociation Characteristics of Human and Sheep Maternal and Fetal Blood," Amer. J. Obs. & Gyn., 1965.

14. Shapiro, N. Z., Analysis by Migration in the Presence of Chemical Reaction, The RAND Corporation, P-2596, June 1962.

15. Denbigh, K. D., Principles of Chemical Equilibrium, 2nd Edition, Cambridge University Press, 1966.

16. Glasstone, S., Textbook of Physical Chemistry, 2nd Edition, D. Van Nostrand, New York, 1946.

17. De Haven, J. C. (ed.), "Practical Prerequisites for Building a Chemical Model from Soda Water to Biological Fluids" (unpublished ms.).

18. Edsall, J. T., and J. Wyman, Jr., Biophysical Chemistry, Vol. 1, Academic Press, Inc., New York, 1958.

19. Magnier, E.A.H., "Unique Models of Individual Blood" (Doctoral thesis), Tulane University, Philadelphia, 1968.

20. DeLand, E. C., The Classical Structure of Blood Biochemistry--A Mathematical Model, The RAND Corporation, RM-4962-PR, July 1966.

21. De Haven, J. C., and E. C. DeLand, The Reactions of Hemoglobin and Steady States in the Human Respiratory System: An Electronic Computer, The RAND Corporation, RM-3212-PR, December 1962.

22. DeLand, E. C., and E.A.H. Magnier, "The Classical Structure of Blood Biochemistry--A Mathematical Model--II" (in preparation).

SELECTED BIBLIOGRAPHY

Bradham, G., et al., "Isotope Dilution and Thermodynamics in the Study of Intercompartmental Body Fluid Exchange," Surg. Cyn. Obs., Vol. 119, November 1964, p. 1062.

Bradham, G., J. C. DeHaven, E. C. DeLand, and J. V. Maloney, "Mathematical Analysis of Complex Biochemical Systems," Fed. Proc., Vol. 22, 1963, p. 573.

Clasen, R. J., The Fitting of Data by Least Squares to Non-Linearly Parameterized Functions, The RAND Corporation, P-3252, October 1965.

-----, The Linear-Logarithmic Programming Problems, The RAND Corporation, RM-3707-PR, June 1963.

Dantzig, G. B., and J. C. De Haven, On the Reduction of Certain Multiplicative Chemical Equilibrium Systems to Mathematically Equivalent Additive Systems, The RAND Corporation, P-2419, August 1961; also published in J. of Chem. Phys., Vol. 38, No. 10, 1962, pp. 2620-2627.

Dantzig, G. B., J. H. Folkman, and N. Z. Shapiro, On the Continuity of the Minimum Set of a Continuous Function, The RAND Corporation, RM-4657-PR, February 1966; also in J. Math. Anal. & Appl., Vol. 17, No. 3, March 1967.

De Haven, J. C., E. C. DeLand, N. S. Assali, and W. Manson, Physiochemical Characteristics of Placental Transfer, The RAND Corporation, P-2565-1, February 1965.

DeLand, E. C., Simulation of a Biological System of an Analog Computer, The RAND Corporation, P-2307, May 1961.

-----, Some Experiments and Problems in Mathematical Biology, The RAND Corporation, P-2191, April 1961.

DeLand, E. C., and G. Bradham, Fluid Balance and Electrolyte Distribution in the Human Body, The RAND Corporation, RM-4347-PR, January 1965.

DeLand, E. C., and M. Wolf, The Simulation of Multi-Component Distillation, The RAND Corporation, RM-3258-PR, October 1962.

Kirschbaum, T. H., N. Z. Shapiro, and N. S. Assali, A Mathematical Model of Placental Oxygen Transfer, The RAND Corporation, RM-5262-PR (in preparation).

Maloney, J. V., J. C. De Haven, et al., Analysis of Chemical Constituents of Blood by Digital Computer, The RAND Corporation, RM-3541-PR, April 1963; also published in Simulation, Vol. 2, No. 6, June 1963, pp. R10-R22, and Surgery, Vol. 54, 1963, p. 158.

-----, "Examples of a Large-Model Simulation of the Blood Biochemical System," J. Chronic. Dis., Vol. 19, 1966, pp. 411-425.

Wolf, et al., "Laboratory and Digital Computer Study of pH of Blood at Hypothermic Levels," San Diego Symposium, August 1964.

DOCUMENT CONTROL DATA

1. ORIGINATING ACTIVITY	2a. REPORT SECURITY CLASSIFICATION
THE RAND CORPORATION	UNCLASSIFIED
	2b. GROUP ---

3. REPORT TITLE

CHEMIST--THE RAND CHEMICAL EQUILIBRIUM PROGRAM

4. AUTHOR(S) (Last name, first name, initial)

DeLand, E. C.

5. REPORT DATE	6a. TOTAL No. OF PAGES	6b. No. OF REFS.
December 1967	143	

7. CONTRACT OR GRANT No.	8. ORIGINATOR'S REPORT No.
F44620-67-C-0045	RM-5404-PR

9a. AVAILABILITY / LIMITATION NOTICES	9b. SPONSORING AGENCY
DDC-1	United States Air Force Project RAND

10. ABSTRACT

A detailed report on the structure and use of CHEMIST, a RAND computer program designed to simulate complex chemical equilibria. The study was compiled in response to a growing demand for a reference manual to accompany and document the program. CHEMIST is a program for use by professionals not trained in computer programming. Communication with the program is in English, chemical, and FORTRAN languages. The computer code currently exists in FORTRAN IV for the IBM 7044. In its present form it occupies approximately 25,000 words for the principal part. Additional specialized subroutines not essential to the operation can increase space requirements. The program uses an iterative mathematical programming technique to determine the composition that minimizes the total free energy of a chemical system, subject to system constraints. A detailed program description with examples is given, along with the program subroutines. The References and Selected Bibliography comprise as complete a listing of the literature as is currently possible. This manual will be updated as the CHEMIST program evolves further.

11. KEY WORDS

Chemistry
Physiology
Computer simulation
Models
Computer programs