CALPHAD Vol. 11, No. 1, pp. 1-55, 1987
Printed in the USA.

0364-5916/87 $3.00 + .00

AN ALGORITHM AND COMPUTER PROGRAM FOR CALCULATING COMPOSITION PHASE DIAGRAMS

J. A. D. Connolly* and D. M. Kerrick
Department of Geosciences, The Pennsylvania State University, University Park, PA 16802 USA

ABSTRACT-An algorithm has been developed to determine the minimum Gibbs energy surfaces and composition phase diagrams of chemical systems. In order to apply the algorithm, phases are represented by points in the parametric space of a system. This requires that the compositional variation of solutions be described by a series of "pseudo-compounds" of differing compositions. All phase regions of an approximated c component system are thus linear as they are defined by the coordinates of c point phases (i.e., phases of fixed composition), which may correspond in part or entirely to a single solution phase. These regions form c-1 dimensional simplicial facets on the Gibbs energy-composition surface of the system and are identified by an abbreviated combinatorial method. This method is feasible because the chemical potentials of the components are constant in any region and can be rapidly determined from linear algebraic techniques. The true phase rule variance of phase regions is found by counting the number homogeneous phases, which may be represented by one or more point phases, in each region as identified by the algorithm. The algorithm can be generalized to other thermodynamic state functions for systems with additional extensive independent variables, such as volume and entropy. The procedure has been coded as a FORTRAN computer program, Bounds, which is capable of treating five component systems with up to eight hundred point phases. Because the calculation of stable phase equilibria is assured by the algorithm, Bounds can be used to calculate composition phase diagrams for systems with complex phase relations, e.g., multinodal solvi. Bounds also provides the basis for a simple method of calculating phase diagrams as a function of both composition and variables like pressure, temperature, and chemical potentials.

Introduction

This communication describes an efficient algorithm and computer program, Bounds, for deriving the minimum Gibbs energy-composition (G-X) surface of a chemical system. This surface represents the equilibrium states of a system for which compositional parameters are independent variables, and projection of the surface onto composition space defines a composition phase diagram. The algorithm was originally developed for subsolidus silicate systems which characteristically contain a large number of potentially stable condensed phases. Typically, only a few of these phases exhibit even limited solution behaviour. The goal of this work was to accommodate such systems by an easily automated means which could take advantage of the limited extent of silicate solution. Composition phase diagrams are of considerable value in themselves, but perhaps more importantly, efficient description of them is virtually essential for the automated computation of Schreinemakers-type phase diagrams for complex systems.

In the past two decades, refinements of thermodynamic equations of state have made it possible to accurately locate many silicate phases in G-X space. With these constraints the stable phase assemblages for a given composition can be predicted from thermodynamic principles. The necessary computations, though simple in concept, are generally so tedious that they cannot be carried out by hand. This complication has been overcome by the development of computer algorithms which identify the stable phase configuration at a point in the parametric space characterizing the state of the system (e.g., 1-8). These algorithms possess a mathematical sophistication which make computations in complex multicomponent systems possible. However, their use in incremental mapping of phase regions when the independent parameters of a system are unconstrained generally involves redundant calculations (e.g., 4, 5, 9, 10) or utilizes some special, and often assumed, characteristic of a system (e.g., 11, 12). In addition, these methods are not very robust in that often: (i) some a priori knowledge of the system is necessary; (ii) computer programs are often limited to a specific model of solution behavior (e.g., Margules formulations); and (iii) the output cannot readily be expressed analytically. The fundamental difficulty, though, is that a system has an infinite number of possible compositions and mapping is done by finding the assemblage for a given composition. The alternative is to identify the finite number of phase regions which represent all compositions of a system. The G-X surfaces of most systems are nonlinear, so such an approach would be difficult to apply directly, but the approach can be used for linear

----

*Current address: Institut fur Mineralogie und Petrographie, Eidgenossische Technische Hochschule, CH-8092 Zurich, SWITZERLAND.

Received 22 May 1985, in final form 26 November 1986

TABLE 1

List of Symbols

c	number of inert components in a system
G	Gibbs energy
G_i^m	G per molar unit of the ith phase
n_j	number of moles of the jth component
n_{ij}	n_j in the ith phase
p	number of equilibrium phases
α_j	stoichiometric coefficient of the jth phase
μ_j	chemical potential of the jth component
Π	total number of potentially stable phases
σ	number of inert components in a solution phase
P	pressure
S	entropy
T	temperature
U	internal energy
V	volume
X_j	mole fraction of the jth component.
μ_{ij}	μ_j in the ith phase
ν_{ij}	stoichiometric coefficient of the jth component per molar unit of the ith phase

approximations of such surfaces. These approximations can be justified for the multicomponent systems and solutions of interest here.

Computational Method

The following three sections provide an overview of the thermodynamic fundamentals, and geometric constraints of the minimum Gibbs energy surface of chemical systems. More detailed discussions of these subjects may be found in many texts; the works of Palatnik and Landau (13) and Korzhinskii (14) are particularly appropriate.

Geometry of Gibbs Energy Surfaces

The equilibrium state and extent of an isobaric-isothermal system is described by either the chemical potential (μ) or the mass (n) of each of its components, with the proviso that the mass of at least one component must be specified. A component is designated as perfectly mobile or inert depending on whether its chemical potential or mass, respectively, is an independent variable of the system (14 p. 61). The problem of concern here is to determine the equilibrium states of a system at arbitrarily fixed P-T-μ conditions as a function of the remaining independent variables, i.e. the inert components. For the sake of brevity, such systems will be described as c component systems, where c denotes the number of inert components.

The state of a c component system can be represented in a c+1 dimensional parametric space defined by the independent parameters n_1, ..., n_c, and the Gibbs energy[1]. The extent of a system is unimportant in the description of phase relations so the dimension of the parametric space can be reduced by one with no loss in relevant information. This will be accomplished here by taking a section through the space such that the total number of moles of the components of the system is always unity. The n_j in this section are thus numerically equivalent to the proportion of each component and will be denoted by X_j to distinguish them from unconstrained n_j. The advantage of this definition is that the X_j are extensive properties which obviates the necessity of creating molar variables for the system (cf. 15). However, to avoid confusion, intensive variables of state will be called potential variables (16, 13 p. 139) to distinguish them from molar variables, which are also intensive. For practical applications the composition space of any system considered must be finite, thus it is necessary to impose the additional restriction that $X_j \geq 0$ (j = 1, ..., c). The allowed composition space, as defined by the X_j, is a c-1 dimensional simplex. The vertices of this simplex represent compositions equivalent to those of the components, and any permutation of less than c vertices defines a compositional join.

In the parametric space of a heterogeneous system there is a unique G-X surface corresponding to each phase. Two assumptions are sufficient to determine the relation of the G-X surface of a system to the surfaces of its constituent phases. These are the assumption of the validity of the Gibbs equilibrium criterion, and the assumption that any extensive property of a system is the weighted sum of that property in each coexistent phase of the system. This surface can be conceived of as the successive tangents, between surfaces of coexistent phases, formed by the motion of a c-1 dimensional plane as it is rolled across the phase surfaces. The different phase configurations on this surface divide it into phase regions, and the projection of these regions onto composition space is a composition phase diagram. The variance in such diagrams is sometimes given as the degrees of freedom in the independent parameters (X_j) within a phase region (16). This usage has little descriptive power as it is evident that, in general, all phase regions will be of the same dimension regardless of the nature of the equilibrium within them. Instead, variance will be used here to indicate the number of degrees of freedom in the intensive parameters of the system, which are equivalent to the number of possible "variations of phase" as described by Gibbs (15 p. 96). A region of c phases is thus invariant, whereas a one phase region has a variance of c-1.

The partial derivatives $(\partial^2 G / \partial X_j^2)$ (j = 1, ..., c) are positive for any compositionally nondegenerate phase region of less than c phases. The G-X surfaces of heterogeneous systems thus have complex nonlinear geometries which are difficult to describe analytically. However, if a system contains only phases with fixed stoichiometries, i.e. point phases, then the only

----

[1] The Gibbs energy is the correct state function only when all components of a system are inert. If one or more components are perfectly mobile, i.e. the chemical potentials of these components are independent variables of the system, then it is necessary to derive a state function with the corresponding potentials as independent variables by means of Legendre transformations. To avoid an unnecessarily elaborate discussion, the Gibbs energy will be used here with the understanding that an alternative state function should be substituted when appropriate.

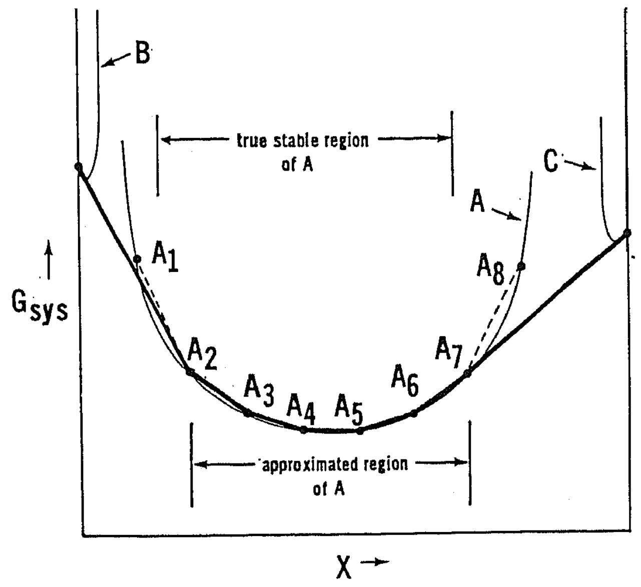


Figure 1. G-X diagram for a binary system with three stable phases, A, B, and C. "True" G-X surfaces of the phases are shown as light, solid lines; pseudo-compound coordinates are indicated by dots; and the minimum G-X surface of the system, consistent with the pseudo-compounds, is shown as a heavy solid line.

phase regions[2] are invariant and linear functions of composition. The G-X surface of such a system is therefore a piecewise linear simplicial complex. The vertices of each c-1 dimensional simplex represent the phases of an invariant region. The geometry of this complex is easily and completely described by listing the phase configurations of each region and specifying the coordinates of the stable phases.

The algorithm was originally developed to determine the geometry of bounding simplicial complexes because the only stable phases of many silicate systems are essentially point phases. Due to the efficiency of the algorithm, it can be applied to systems which contain solutions that are approximated by closely spaced point phases. To emphasize the distinction between a true point phase and one which corresponds to a part of a series devised to describe the compositional variation of a solution phase, the latter will be termed "pseudo-compounds". It is important to recognize that the compositional coordinates of pseudo-compounds are arbitrarily specified within the compositional limits of the solution which they represent. By decreasing the spacing of pseudo-compounds it is possible to obtain greater detail of the phase relations for the corresponding solution phase. The application of pseudo-compounds is illustrated by Figure 1 which shows a G-X diagram for a hypothetical binary system. The system contains three true phases; two (B and C) have virtually invariant stoichiometry and are adequately represented by point phases. The other phase (A) is a solution which is subdivided into a series of pseudo-compounds (A_1, ..., A_8). The surface of the system consists of two true invariant regions (B-A_2 and C-A_7) and four pseudo-invariant regions (A_2-A_3, ..., A_6-A_7) which approximate the univariant region of the solution A. In the remainder of this paper the prefix "pseudo" will be used to describe the variance of regions in which two or more of the constituent pseudo-compounds represent the same true phase. For example, an assemblage of three compositionally adjacent pseudo-compounds corresponding to a solution phase in a ternary system is pseudo-invariant; since the assemblage represents part of a homogeneous region it is actually divariant.

Input to the algorithm consists only of point phase coordinates; thus, solutions must be subdivided into pseudo-compounds prior to its application. Because only point phases are considered, the algorithm is independent of the compositional equations of state used to establish the G-n coordinates of pseudo-compounds. Computations can thus be done on phases modeled by any kind of equation of state, e.g., non-ideal molecular fluids and ionic solutions.

Thermodynamic Principles and Criteria for Stability

The Gibbs criterion for equilibrium implies that the stable phase configurations of a system are those which minimize its Gibbs energy. If it is assumed that the differential of the Gibbs energy of a system is a first order homogeneous differential with respect to the masses of the constituent phases, then it follows from Euler's theorem on homogeneous functions that:

[2] The compositional variation of any configuration of less than c point phases can be described by fewer than c components; therefore, such configurations do not constitute a region in a c component system. Degeneration of high variance regions into boundaries leads to an exception of the "contact rule" (14, p. 236). Some workers attempt to explain such exceptions by arguing that complete invariance of phase stoichiometry is impossible on the basis of the "phase rule" (17). That this argument is inconsistent with the work of Gibbs (15 p. 97), has been recognized by many authors (14, 18). The relation between neighboring phase regions involving degenerate phases can, however, be correctly deduced according to the method outlined by Zhao (19).

$$G = \sum_{i=1}^{p} \sum_{j=1}^{c} \mu_{ij}\, n_{ij}, \tag{1}$$

where the chemical potentials (μ_{ij}) of each component in each phase are defined as:

$$\mu_{ij} \equiv (\partial G_j \,/\, \partial n_{ij}). \tag{2}$$

The G-n surface of any phase is therefore described by the relation:

$$G_i = \sum_{j=1}^{c} n_{ij}\, \mu_{ij}. \tag{3}$$

The ratios between the n_{ij} are fixed in a point phase; for such phases equation (3) defines a ray which emanates from the origin of G-n space. This ray can be located if the coordinates of the phase are known at any point other than the origin, e.g., the representative point defined by the molar stoichiometry of the phase ($G = G_i^m$, $n_1 = \nu_{i1}$, ..., $n_c = \nu_{ic}$). Because equation (3) is an equation in c unknowns and one independent variable, the μ_{ij} are not defined for a single point phase. However, from the conditions formulated by Gibbs, the chemical potential of each component, in every phase in which it occurs, must be equal in a heterogeneous equilibrium. Thus, for equilibrium equation (1) may be repaced by:

$$G = \sum_{j=1}^{c} \mu_j\, n_j. \tag{4}$$

Any compositionally nondegenerate assemblage of p = c point phases can be used to set up a system of linear equations which define equilibrium chemical potentials:

$$\begin{vmatrix} \nu_{11} & \cdots & \nu_{1c} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \nu_{p1} & \cdots & \nu_{pc} \end{vmatrix} \begin{vmatrix} \mu_1 \\ \cdot \\ \cdot \\ \cdot \\ \mu_c \end{vmatrix} = \begin{vmatrix} G_1^m \\ \cdot \\ \cdot \\ \cdot \\ G_p^m \end{vmatrix} \quad \text{or} \quad \boldsymbol{\nu}\ \boldsymbol{\mu} = \tilde{G}. \tag{5}$$

These equations locate a plane in G-n space tangent to the G-n surfaces of the coexistent phases. If the phase assemblage is stable, then the plane must lie at lower values of Gibbs energy than the surface of any phase not in the assemblage. This conditional may be expressed as:

$$G_i^m \geq \sum_{j=1}^{c} \nu_{ij}\, \mu_j \quad (i = 1, ..., \Pi), \tag{6}$$

which must be true for all, Π, possible phases of a system. Phases must have non-negative masses, therefore, only the portion of a given G-X plane, located by equation (5), for which the condition (6) is true defines the compositional limits of a phase region. Valid spanned compositions (X_1, ..., X_c) are such that the stoichiometric coefficient of each phase in a configuration (α_1, ..., α_p) determined by solving:

$$\begin{vmatrix} \nu_{11} & \cdots & \nu_{p1} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \nu_{1c} & \cdots & \nu_{pc} \end{vmatrix} \begin{vmatrix} \alpha_1 \\ \cdot \\ \cdot \\ \cdot \\ \alpha_p \end{vmatrix} = \begin{vmatrix} X_1 \\ \cdot \\ \cdot \\ \cdot \\ X_c \end{vmatrix} \quad \text{or} \quad \boldsymbol{\nu}^t\ \boldsymbol{\alpha} = \tilde{X}, \tag{7}$$

satisfy the condition:

$$\alpha_i \geq 0 \quad (i = 1, ..., p). \tag{8}$$

Chemical multisystems consisting only of point phases can be visualized as a bundle of rays in G-n space, or as a set of points in G-X space, where the geometric elements represent possible phases. The problem of interest here is the identification of the bounding conic polyhedral surface that is strictly concave with respect to the G ordinate. The facets of this surface determine phase assemblages and regions for which the conditions (6) and (8), based on equations (5) and (7), are true, for all phases and compositions, respectively.

These conditions may suggest brute force is not an untenable means of determining the configurations of the surface. This method would involve enumeration of all possible c phase configurations. Any configuration with a concentration matrix, $\boldsymbol{\nu}$, of full rank, could be solved for $\boldsymbol{\mu}$. If the condition (6) were then found true for all additional phases the configuration would be accepted as a stable assemblage. Unfortunately, the number of possible configurations, given by

$$\Pi! \,/\, [(\Pi\text{-}c)!\ c!]. \tag{9}$$

increases combinatorially as a function of both the number of components and Π, the total number of potentially stable phases. Thus, while the simplicity of such an approach is attractive, some attempt must be made to limit the number of trial configurations considered in order to deal with complex systems.

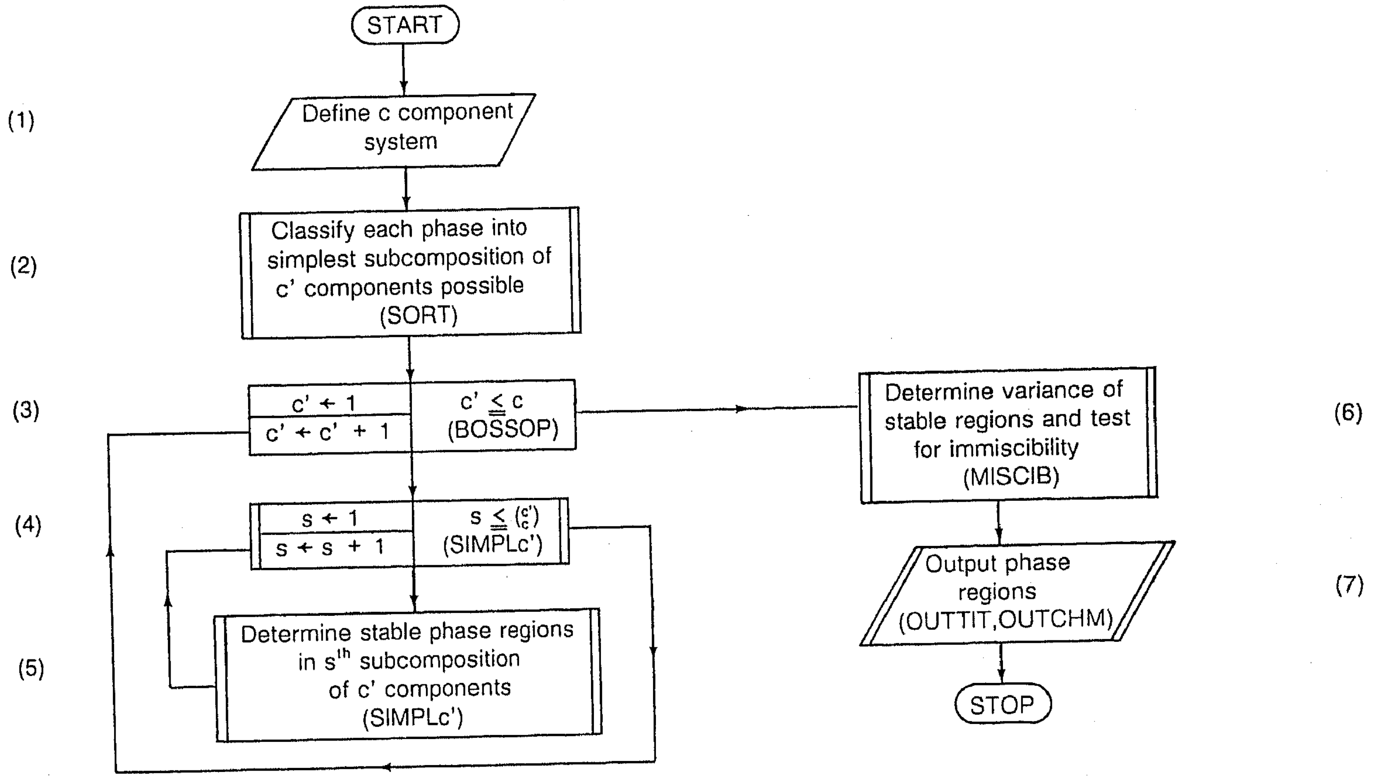

START
(1)
Define c component system
(2)
Classify each phase into simplest subcomposition of c' components possible (SORT)
(3)
c' ← 1
c' ← c' + 1
c' ≤ c (BOSSOP)
(4)
s ← 1
s ← s + 1
s ≤ (c' c) (SIMPLc')
(5)
Determine stable phase regions in s^th subcomposition of c' components (SIMPLc')
(6)
Determine variance of stable regions and test for immiscibility (MISCIB)
(7)
Output phase regions (OUTTIT,OUTCHM)
STOP

Figure 2. Flow chart illustrating the overall structure of Bounds. Subprogram references are indicated parenthetically (see Appendix B for additional description).

Geometric Constraints

The strategy adopted in the Bounds algorithm is to utilize the constraints imposed by the simplicial geometry of the G-X surface to eliminate as many trial phase configurations as possible. This is accomplished by evaluating the G-X surfaces of progressively more complex subsystems of the components as shown by the flow chart in Figure 2. The surfaces so defined must be stable in the full system; therefore, phases which are metastable in a subsystem may be disregarded in subsequent calculations. Furthermore, each invariant configuration in a c-1 component subsystem forms a phase region boundary in the c component system. Thus, only one phase[3] in the interior of the adjacent composition space need be found, in addition to the assemblage on this boundary, in order to define a phase region. Similarly, each configuration found in the interior of a c component composition space generates up to c-1 new boundaries.

By keeping account of phase region boundaries, the combinatorial problem of determining stable configurations is reduced to that of finding a single phase that is compatible with a boundary assemblage. This process is outlined for a three component system in Figure 3 and is illustrated in the composition phase diagram of figure 4a. Once the stable binary joins have been established, a binary assemblage, e.g., A-B , can be used to identify a ternary configuration (steps 4-16, Fig. 3). If the system contains no ternary phases there are six trial configurations compatible with A-B as shown by the dashed lines in figure 4a. On average half the possible configurations will be tested before the stable one is identified; for this example roughly 35 trials would be necessary to define the ternary diagram as opposed to 165 by brute force. The example shows brute force in its best light; in more complex systems the difference in the number of trials has ranged up to four orders of magnitude in computer experiments on ternary systems.

Additional efficiency can be achieved by noting several consequences of the geometry of the Gibbs energy surface, which follow from the fact that phase region boundaries are geometric elements of one dimension less than phase regions: (i) Any boundary that does not lie on a degenerate join, and the phase assemblage which defines it, can be common to two and only two regions. The c-1 dimensional element which spans such a boundary divides the composition space of a system into two parts and only one region adjoining the boundary may occur in each part. (ii) From (i) it follows that only phases on the opposite side of a boundary from a phase known to coexist with the boundary assemblage may form a second stable configuration with the boundary assemblage. (iii) A trial configuration, consisting of a boundary assemblage and a trial phase, which spans the composition of a stable phase must be metastable with respect to the configuration of the spanned phase plus the

----

[3] It will sometimes be convenient to use language that attributes to geometric elements properties which belong to the associated state of a system or a constituent phase, and vice versa. Thus it can give rise to no ambiguity, to state that a phase which lies within a region, instead of stating that a region spans the coordinates associated with the phase.

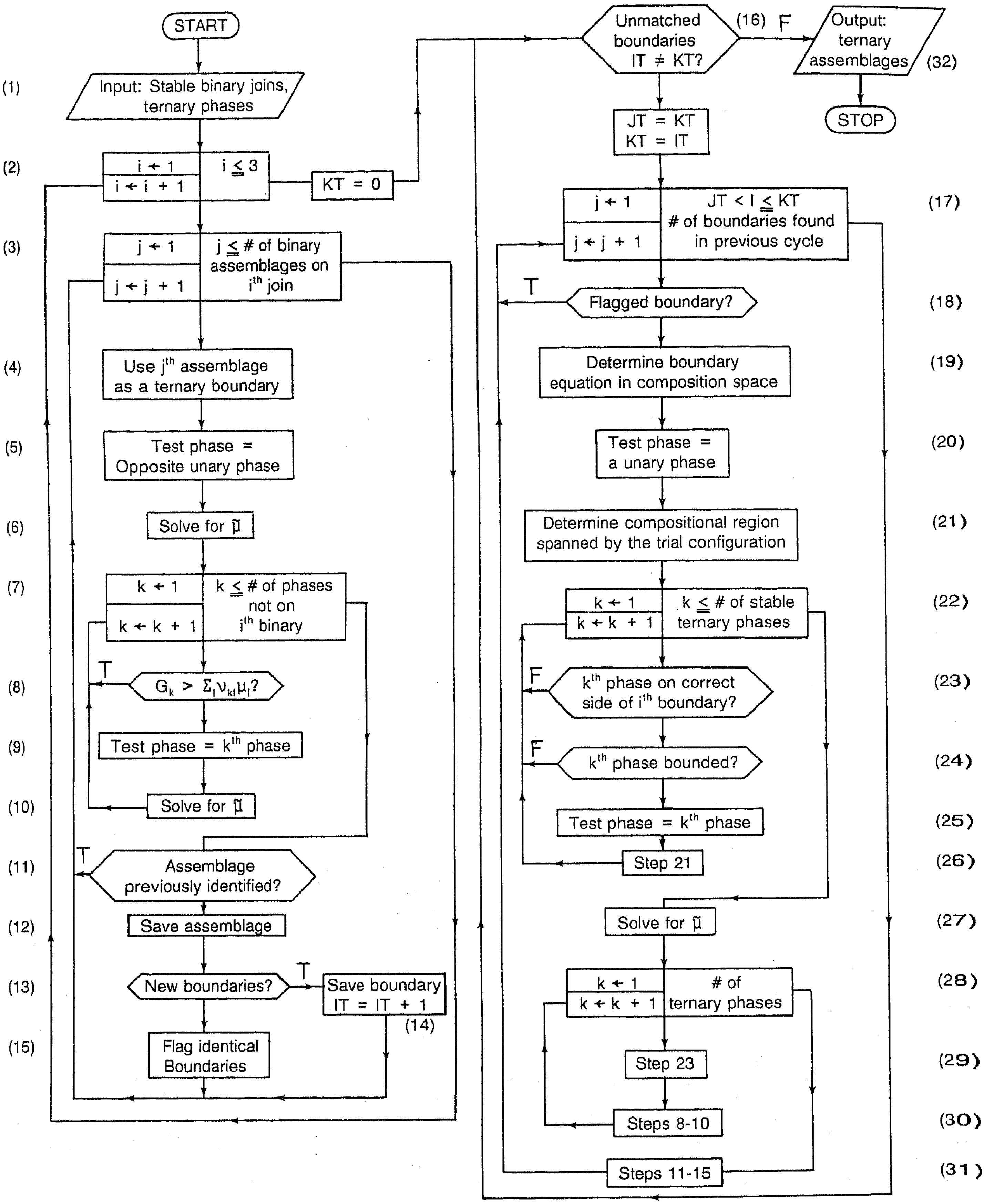

START
(1) Input: Stable binary joins, ternary phases
(2) i ← 1 | i ← i + 1 | i ≤ 3
KT = 0
(3) j ← 1 | j ← j + 1 | j ≤ # of binary assemblages on i^th join
(4) Use j^th assemblage as a ternary boundary
(5) Test phase = Opposite unary phase
(6) Solve for μ̃
(7) k ← 1 | k ← k + 1 | k ≤ # of phases not on i^th binary
(8) $G_k > \Sigma_l \nu_{kl}\mu_l$?
T
(9) Test phase = k^th phase
(10) Solve for μ̃
(11) Assemblage previously identified?
T
(12) Save assemblage
(13) New boundaries?
T
Save boundary IT = IT + 1
(14)
(15) Flag identical Boundaries
Unmatched boundaries IT ≠ KT?
(16) F
Output: ternary assemblages
(32)
STOP
JT = KT
KT = IT
(17) j ← 1 | j ← j + 1 | JT < l ≤ KT # of boundaries found in previous cycle
(18) Flagged boundary?
T
(19) Determine boundary equation in composition space
(20) Test phase = a unary phase
(21) Determine compositional region spanned by the trial configuration
(22) k ← 1 | k ← k + 1 | k ≤ # of stable ternary phases
(23) k^th phase on correct side of i^th boundary?
F
(24) k^th phase bounded?
F
(25) Test phase = k^th phase
(26) Step 21
(27) Solve for μ̃
(28) k ← 1 | k ← k + 1 | # of ternary phases
(29) Step 23
(30) Steps 8-10
(31) Steps 11-15

Figure 3. Flow chart of the ternary system algorithm. A "test phase" is the phase which, in combination with the two phases on a phase region boundary, forms a potentially stable ternary "trial configuration".

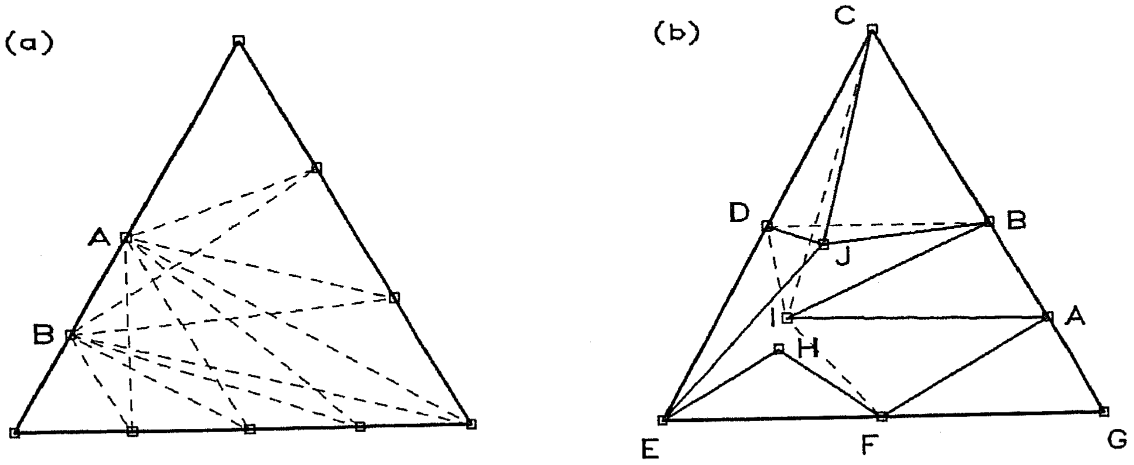


Figure 4. Chemographic relations for two hypothetical ternary systems, compositions of phases are indicated by open squares. (a) Possible boundaries (dashed lines) compatible with the binary assemblage A-B. (b) Stable boundaries consistent (solid lines) with the binary assemblages and some incompatible ternary boundaries (dashed lines) as discussed in the text.

boundary assemblage. A corollary to (iii) is that if a trial configuration is metastable with respect to a phase which is within the compositional limits of the configuration, then the configuration of the spanned phase and the boundary assemblage will be relatively more stable.

Application of these constraints is demonstrated by the example shown in Figure 4b. This figure illustrates a computational stage at which the seven stable binary assemblages have been paired with ternary phases so as to define six ternary phase regions delimited by solid lines in figure 4b (A-B-I, B-C-J, C-D-J, etc., steps 2-15, Fig. 3). These regions, in turn, generate nine two-phase boundaries within the ternary system (A-I, B-I, B-J, etc.). From the first constraint (i, above) the boundaries D-J and C-J need not be considered since both are shared by two regions. Identification of each ternary assemblage will eliminate at least one other boundary, e.g., if A-I is stable with F then trials with A-F would be redundant (step 19, Fig. 3). Of the remaining trial configurations, roughly half could be eliminated by the second constraint (ii, above, steps 19, 24, 30, Fig. 3). For example, consider the boundary assemblage B-I which coexists with A; only D, J, and C, which lie on the opposite side of B-I from A, may form stable configurations. Of the three possibilities both B-I-D and B-I-C bound J and thus must be metastable relative to B-I-J (steps 23, 24-27, Fig. 3) which must be stable by default. Given that B-I-J and A-I-F are stable assemblages, then the only possible boundary, consistent with boundaries F-H, E-H, and E-J, is, by the corollary to the third constraint, H-I. If H-I is a stable boundary, every boundary, within the ternary, is shared by two regions and the topology of the diagram is completely defined (step 17, Fig. 3).

Although it is possible to make more frequent tests for the feasibility of configurations, and to perform tests other than are outlined in Figure 3, none were found which resulted in a significant increase of efficiency. An occasional exception to this statement arises when the proportion of metastable phases is relatively high, i.e. greater than ninety-five percent of the total number. In this case it is useful to flag metastable phases as they are identified so that these phases may be omitted from subsequent tests.

Variance and Immiscibility in Pseudo-Compound Assemblages

Assemblages in which two or more pseudo-compounds represent the same solution are in general only pseudo-invariant and therefore define only a fraction of a true phase region. The exception to this generality occurs when a solution is immiscible, since it is then possible for pseudo-compounds to represent the multiple phases of a solution separated in composition by a solvus. If no possibility of immiscibility exists variance is obtained by simply subtracting the number of distinct phases represented in a configuration from the total number of components, as previously described. In a true σ component solution a solvus is a compositional region over which at least two and at most σ phases of the solution coexist, and within which there are no stable intermediate compositions. In terms of pseudo-compounds, solvi can be recognized by the existence of pseudo-invariant regions which span the compositions of metastable pseudo-compounds representing the same solution as the compounds defining the nodes[4] of the regions. Algebraically, this implies that if the composition of a metastable pseudo-compound can be described by a positive linear combination of the stable pseudo-compounds, representing the same solution, then the solution is immiscible. Thus, if a solution is heterogeneous within a region then there must be at least one metastable pseudo-compound such that the stoichiometric coefficients of the stable pseudo-compounds representing the solution (α_i, ..., α_o) are greater than or equal to zero. These coefficients are determined by solving:

$$\tilde{\nu}^t \tilde{\alpha} = \tilde{n}, \tag{10}$$

[4]The term node is used here to indicate the point of tangency between the thermodynamic surface of a phase and that of the system (13 p.131).

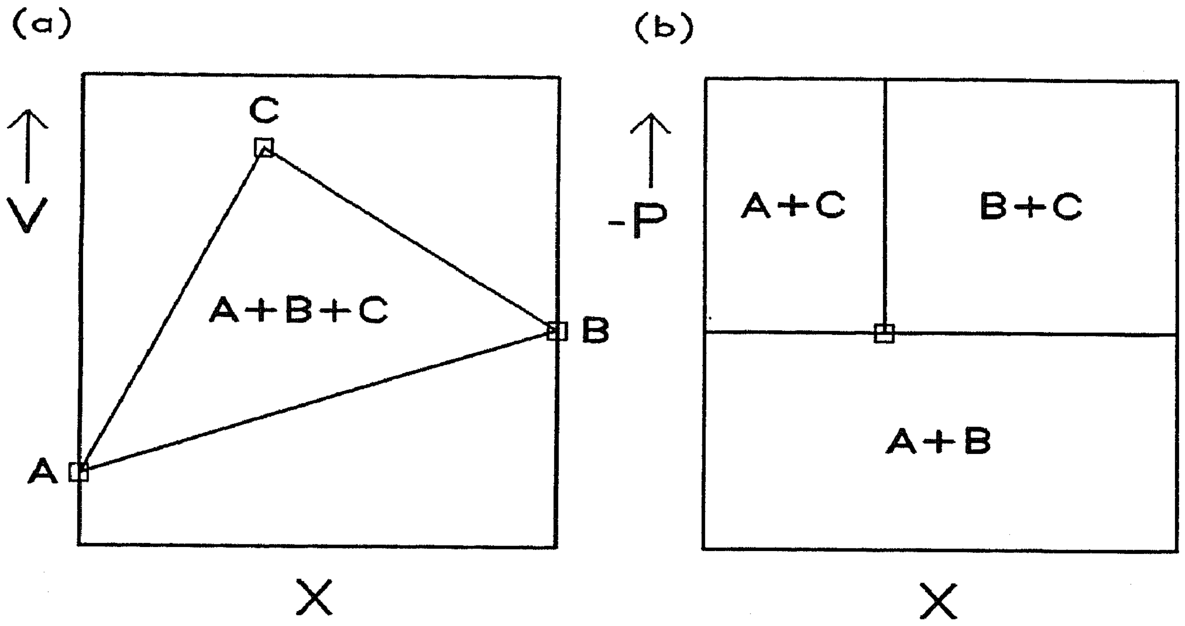


Figure 5. Thermodynamically equivalent V-X (a) and P-X (b) phase diagrams for a hypothetical binary three phase system. The phases A, B and C, have invariant compositions and molar volumes.

where $\boldsymbol{\eta}$ is a concentration vector for the metastable pseudo-compound and $\boldsymbol{\nu}^t$ is the transpose of the concentration matrix for the phase configuration defining the region (cf., eq. (7)). In regions where $\sigma < c$ it is also necessary to stipulate that the stoichiometric coefficients of any pseudo-compounds or phases which do not correspond to the solution in question must be zero. This algebraic method provides a simple technique for determining if immiscibility occurs in a system, within the compositional resolution of the pseudo-compounds, once the stable phase configurations have been determined. However, it must be noted that if a configuration of pseudo-compounds describes a region of immiscibility the true variance cannot be found directly, as it is not possible to distinguish between two phase (binodal) and more complex (multi-nodal) solvi.

Limitations

Limitations to the use of the Bounds algorithm fall into theoretical and practical categories. Theoretical limitations follow from the assumption that if the composition of a system lies within a join defined by the selected components, then the compositions of its constituent phases are also in the join. This requires that each boundary of a composition phase diagram is itself a phase diagram for a subsystem of the components. The assumption implies that no phase region boundary crosses a join defined by a subcomposition of a system. Components, therefore, must be chosen so as to validate the assumption, thus, some a priori knowledge of the phase relations of a system is necessary to apply Bounds. This knowledge, and knowledge of the potential phases of a system, is essential for most phase equilibrium calculations and is not a specific drawback of Bounds. A second consequence of the assumption is the impossibility of constructing a phase diagram section at a constant level of an extensive parameter (S-V-n). This is because there is no thermodynamic relation between the extensive parameters of a system and those of its constituent phases. Thus, the parametric coordinates of phases are not constrained to lie in the reduced parametric space of such a section. In contrast, phase coordinates are constrained to lie in the parametric space of an iso-potential (P-T-μ) section because of the equality of potentials at equilibrium. This distinction may be clarified by comparison of the thermodynamically equivalent V-X and P-X phase diagrams shown in Figure 5. The diagrams are for a hypothetical binary system with three phases of invariant molar volume and composition. It is evident from Figure 5a that there are no nondegenerate isochoric sections in which the V-X coordinates of the phases can be defined in the section. However, because the pressures of the system and its constituent phases are equal, the P-X coordinates of phases can be defined in any isobaric section. Thus, although it is possible to use Bounds to compute the thermodynamic surface over any iso-potential section of a parametric space; it is not, in general, possible to construct isoplethal, isochoric, or isentropic sections.

Bounds can easily be generalized to any number of components; however, because Bounds is fundamentally a combinatorial algorithm, practical considerations of computation time and memory requirement limit its use. The nature of these restrictions is largely dependent on the computational facilities available to the user. These complications are an outcome of the large number of pseudo-compounds necessary to represent a solution phase, and the number of subsystems that can be generated from the components of a system. The latter number can be deduced from the Newton binomial formula. Thus, for a six component system it would be necessary to allocate sufficient memory for 15 binary, 20 ternary, 15 quaternary, and 6 quinary joins (step 4, Fig. 2). The number of phases is a more important factor. This is demonstrated by iterative barycentric subdivision (20) of a solution into pseudo-compounds. In such a scheme the number of pseudo-compound assemblages after the k^{th} subdivision of a σ component solution is given by:

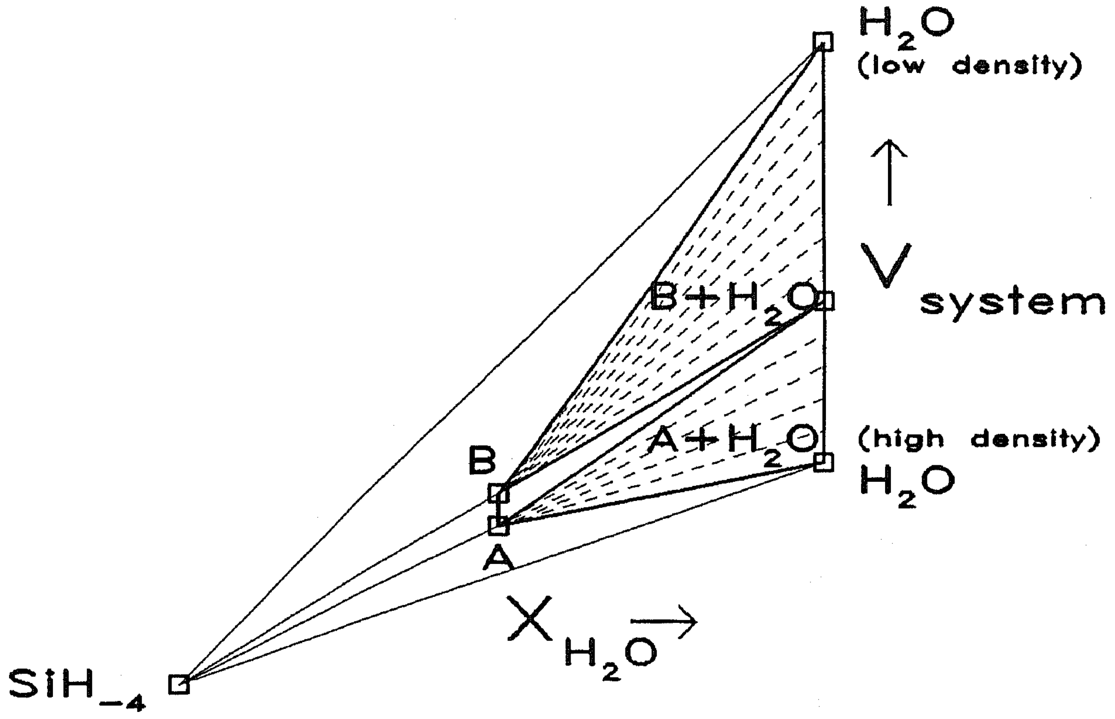


Figure 6. Schematic V-X diagram for the system water-SiH$_{-4}$. The system has three stable phases, water, and two silica polymorphs, A and B. Tielines are shown as dashed lines, and phase region boundaries are depicted with solid lines.

$$(\sigma l)^k. \tag{11}$$

For $\sigma = 5$ and $k = 2$ (for $k = 2$ the solution is represented by five pseudo-compounds along each binary join) 14,400 pseudo-compound assemblages would be generated. In comparison, the fourth subdivision of a ternary solution generates only 1296 assemblages from 673 pseudo-compounds, a fairly manageable quantity. For the kinds of computers currently available at most large computational facilities, the limitations will not be a significant factor for systems with solutions of fewer than four components. The limitations can be reduced if assumptions are made about the behaviour of the solutions to be modeled (e.g., ideal, regular, or subregular models). Such assumptions detract from the generality of the algorithm and so are not considered here.

Application to Other Thermodynamic State Functions

The algorithm has been discussed with specific reference to the Gibbs energy and implicitly with regard to state functions derived from the Gibbs function for systems with perfectly mobile components. This approach was taken because the independent variables of the Gibbs function (P, T, n_1, ..., n_c) are the most widely used in the physical sciences. In some cases, though, it is desireable to consider systems with additional independent extensive parameters such as volume (V) and entropy (S). This can be accomplished with Bounds by creating an augmented parametric space with the desired independent parameters, and substituting a suitable state function. In this regard it may useful to consider "physical" components associated with the conjugate pairs V-P and S-T, just as n-μ pairs can be associated with "chemical" components. In the most general case, where all the independent parameters of a system are extensive, internal energy (U) replaces the Gibbs energy, so that equation (5) becomes:

$$\begin{vmatrix} \nu_{11} \cdots & \nu_{1c} & S_1^m & V_1^m \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \nu_{p1} \cdots & \nu_{pc} & S_p^m & V_p^m \end{vmatrix} \begin{vmatrix} \mu_1 \\ \cdot \\ \cdot \\ \mu_c \\ T \\ -P \end{vmatrix} = \begin{vmatrix} U_1^m \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ U_p^m \end{vmatrix} \tag{12}$$

where $\boldsymbol{\nu}$ and $\boldsymbol{\mu}$ are replaced by generalized "concentration" and "potential" matrices (cf., 16). In a similar manner, the geometric relations and restrictions in this augmented parametric space can be generalized to be exactly analogous to those in G-X space.

Many condensed phases can be represented by point phases in V-X space because of their low expansivity and compressibility. In comparison, heat capacities are energetically more significant than volumetric properties, so the entropy of the

TABLE 2

Point Phase Notation

Symbol	Phase	Composition	Symbol	Phase	Composition
AC3	tricalcium aluminate	$Ca_3Al_2O_6$	Lm	lime	CaO
Ad	andradite	$Ca_3Fe_2Si_3O_{12}$	Mt	magnetite	Fe_3O_4
An	anorthite	$CaAl_2Si_2O_8$	Mu	muscovite	$KAl_3Si_3O_{10}(OH)_2$
And	andalusite	Al_2SiO_5	Pa	paragonite	$NaAl_3Si_3O_{10}(OH)_2$
CA	calcium aluminate	$CaAl_2O_4$	Ep	epidote	$Ca_2FeAl_2Si_3O_{12}(OH)$
CA2	calcium dialuminate	$CaAl_4O_7$	Pw	pseudowollastonite	$CaSiO_3$
CA6	calcium hexaluminate	$CaAl_{12}O_{19}$	Q	quartz	SiO_2
Co	corundum	Al_2O_3	Rn	rankinite	$Ca_3Si_2O_7$
Cr	cristoballite	SiO_2	SC3	tricalcium silicate	Ca_3SiO_5
Ge	gehlenite	$Ca_2Al_2SiO_7$	Tr	tridymite	SiO_2
Gr	grossularite	$Ca_3Al_2Si_3O_{12}$	Wo	wollastonite	$CaSiO_3$
Hm	hematite	Fe_2O_3	Zo	zoisite	$Ca_2Al_3Si_3O_{12}(OH)_2$
La	larnite	Ca_2SiO_4			

Solution Phase Notation

Solution phases, of σ components, are designated by the symbols for σ-1 endmember components with percent molar proportions indicated by subscripts.

Symbol	Phase	Composition
Gr_x	grandite garnet	Gr_xAd_{1-x}
Mi_x	supercritical mica	Mu_xPa_{1-x}
Mu_x	subcritical Mu rich mica	Mu_xPa_{1-x}
Pa_x	subcritical Pa rich mica	Pa_xMu_{1-x}
S_x	CS melt	$(CaO)_{1-x}(SiO_2)_x$
S_xA_y	CAS melt	$(CaO)_{1-x-y}(Al_2O_3)_y(SiO_2)_x$
Zo_x	zoisite	Zo_xEp_{1-x}

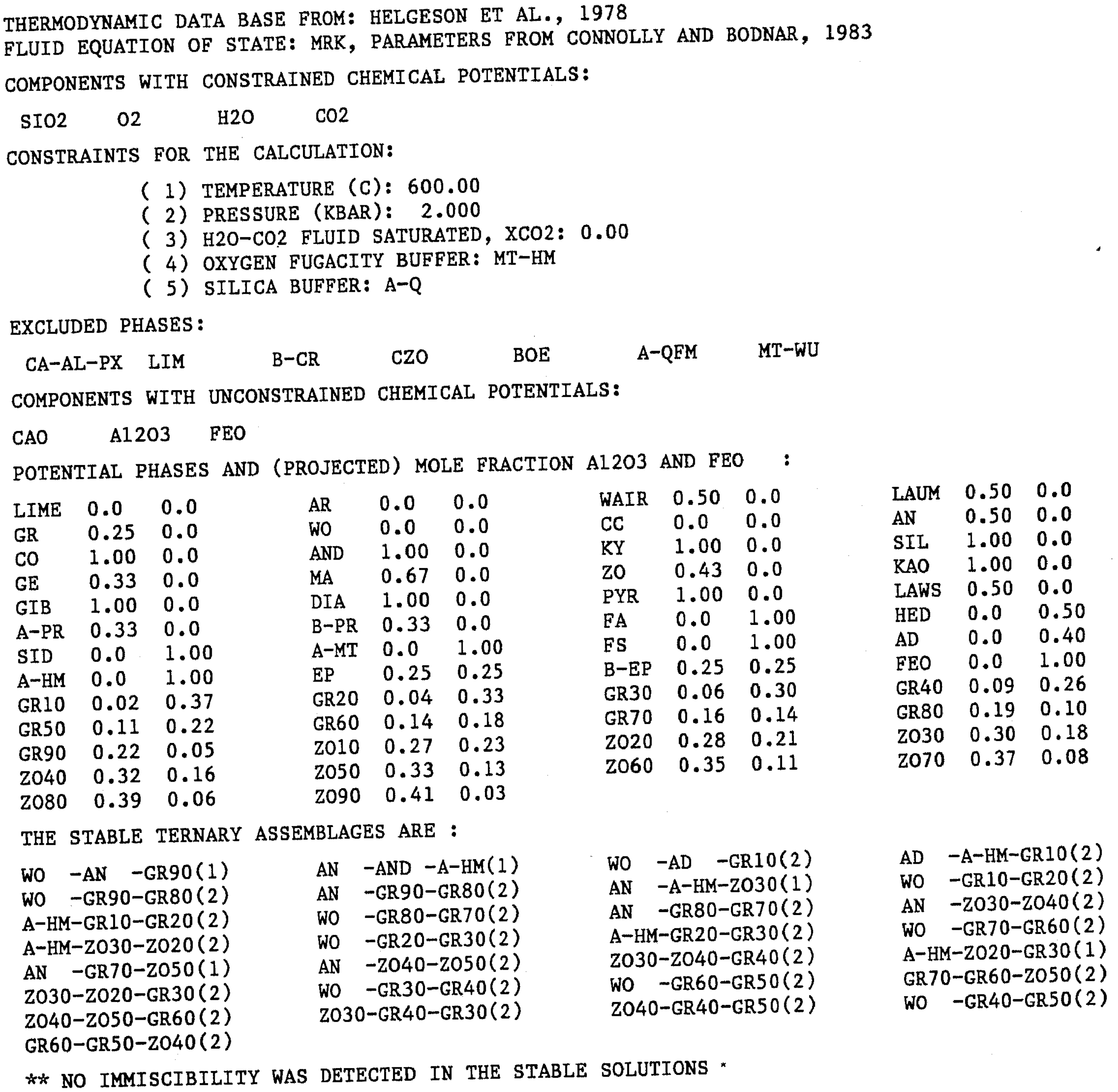
constituent phases of a system cannot be regarded as being independent of the temperature of the system. Point phases in composition space are therefore at least line phases in S-X and S-V-X spaces. Representation of internal energy and enthalpic surfaces thus may require many more pseudo-compounds than necessary to describe the Gibbs and Helmholtz surfaces by the Bounds algorithm.

The assumption, necessary to apply Bounds, that the boundaries of a phase diagram are in the configuration of the joins of a simplex poses a minor difficulty, because "physical" components do not provide for the same natural divisions into subspaces as do chemical components such as the oxides. This difficulty can be circumvented by introducing fictive point phases. For example, consider the isothermal binary system H_2O-SiO_2, under conditions for which the only phases are two pure silica polymorphs and pure water as illustrated schematically in Figure 6. The geometry of this diagram can be "transformed" into a simplex by introducing a fictive phase and component $SiH_{.4}$ (which is assigned appropriate thermodynamic parameters). The transformed diagram, for purposes of the algorithm, can be interpreted as a three component system, maximum density water-minimum density water-$SiH_{.4}$ with the caveat that configurations involving $SiH_{.4}$ do not represent true states.

Details of the Program

The algorithm has been coded as a FORTRAN computer program, BOUNDS, which is listed in Appendix C. The program was written for use on an IBM System model 3081 computer and requires a region of memory of about 75K bytes for the object code and 200K bytes for array area. Optimum performance has been obtained using the IBM VS FORTRAN compiler, but the code should be compatible with most other compilers. The version in Appendix C is dimensioned for 800 phases and coded for systems with up to 5 compositional degrees of freedom. Many of the subprograms were written for chemical systems of specific dimensionality. These programs were not condensed into more general programs because condensation would result in increased execution time. Execution times are strongly dependent on the problem of interest; for five component systems with more than 200 phases, execution times are typically on the order of one to ten seconds. No provision is made to evaluate equations of state or to subdivide solutions into point phases since these functions are not readily generalized (several subprograms for specific functions are available from the authors). Most envisioned applications of the program entail its use as a subprogram. The main program in Appendix C serves primarily as an example of how the remainder of the program may be used.

Experience with the program has led to the following suggestions which the reader may wish to take note of: (i) It is most efficient to carry out initial calculations with coarse subdivision of solutions. Resolution may be increased later for compositional regions where the solution is stable. (ii) Iterative barycentric or cartesian subdivision are the most useful schemes for ideal ionic solutions. Non-ideal solutions are better represented by a scheme in which the spacing of point phases varies



```
THERMODYNAMIC DATA BASE FROM: HELGESON ET AL., 1978
FLUID EQUATION OF STATE: MRK, PARAMETERS FROM CONNOLLY AND BODNAR, 1983

COMPONENTS WITH CONSTRAINED CHEMICAL POTENTIALS:

 SIO2    O2      H2O     CO2

CONSTRAINTS FOR THE CALCULATION:

           ( 1) TEMPERATURE (C): 600.00
           ( 2) PRESSURE (KBAR):  2.000
           ( 3) H2O-CO2 FLUID SATURATED, XCO2: 0.00
           ( 4) OXYGEN FUGACITY BUFFER: MT-HM
           ( 5) SILICA BUFFER: A-Q

EXCLUDED PHASES:

 CA-AL-PX  LIM       B-CR      CZO       BOE       A-QFM     MT-WU

COMPONENTS WITH UNCONSTRAINED CHEMICAL POTENTIALS:

CAO      A12O3    FEO

POTENTIAL PHASES AND (PROJECTED) MOLE FRACTION A12O3 AND FEO   :

LIME   0.0   0.0      AR     0.0   0.0      WAIR  0.50  0.0      LAUM  0.50  0.0
GR     0.25  0.0      WO     0.0   0.0      CC    0.0   0.0      AN    0.50  0.0
CO     1.00  0.0      AND    1.00  0.0      KY    1.00  0.0      SIL   1.00  0.0
GE     0.33  0.0      MA     0.67  0.0      ZO    0.43  0.0      KAO   1.00  0.0
GIB    1.00  0.0      DIA    1.00  0.0      PYR   1.00  0.0      LAWS  0.50  0.0
A-PR   0.33  0.0      B-PR   0.33  0.0      FA    0.0   1.00     HED   0.0   0.50
SID    0.0   1.00     A-MT   0.0   1.00     FS    0.0   1.00     AD    0.0   0.40
A-HM   0.0   1.00     EP     0.25  0.25     B-EP  0.25  0.25     FEO   0.0   1.00
GR10   0.02  0.37     GR20   0.04  0.33     GR30  0.06  0.30     GR40  0.09  0.26
GR50   0.11  0.22     GR60   0.14  0.18     GR70  0.16  0.14     GR80  0.19  0.10
GR90   0.22  0.05     ZO10   0.27  0.23     ZO20  0.28  0.21     ZO30  0.30  0.18
ZO40   0.32  0.16     ZO50   0.33  0.13     ZO60  0.35  0.11     ZO70  0.37  0.08
ZO80   0.39  0.06     ZO90   0.41  0.03

THE STABLE TERNARY ASSEMBLAGES ARE :

WO   -AN   -GR90(1)     AN   -AND -A-HM(1)     WO   -AD   -GR10(2)     AD   -A-HM-GR10(2)
WO   -GR90-GR80(2)      AN   -GR90-GR80(2)     AN   -A-HM-ZO30(1)      WO   -GR10-GR20(2)
A-HM-GR10-GR20(2)       WO   -GR80-GR70(2)     AN   -GR80-GR70(2)      AN   -ZO30-ZO40(2)
A-HM-ZO30-ZO20(2)       WO   -GR20-GR30(2)     A-HM-GR20-GR30(2)       WO   -GR70-GR60(2)
AN   -GR70-ZO50(1)      AN   -ZO40-ZO50(2)     ZO30-ZO40-GR40(2)       A-HM-ZO20-GR30(1)
ZO30-ZO20-GR30(2)       WO   -GR30-GR40(2)     WO   -GR60-GR50(2)      GR70-GR60-ZO50(2)
ZO40-ZO50-GR60(2)       ZO30-GR40-GR30(2)      ZO40-GR40-GR50(2)       WO   -GR40-GR50(2)
GR60-GR50-ZO40(2)

 ** NO IMMISCIBILITY WAS DETECTED IN THE STABLE SOLUTIONS
```

Figure 7. Output for the sample problem defined in Table 3, Appendix B.

logarithmically away from the compositional barycenter of the solution. (iii) Baryentric subdivision is topologically the most natural scheme and is more likely to faithfully depict the irregular nature of phase boundaries. However, a cartesian scheme allows the user greater freedom in choosing the spacing of pseudo-compounds and is less costly to implement.

Input and Initialization

Input to the program consists of data defining the chemical system of interest and its possible phases, i.e. the components and the molar Gibbs energy and stoichiometry of each point phase. Appendix A contains a glossary of variables and arrays initialized by the input. Input data, as read sequentially by the main program, has four parts: (i) alphanumeric title cards (TNAME); (ii) the number of components (ICP) and an alphanumeric label for each (CNAME); (iii) a flag indicating whether a graphics file for graphics applications is to be output (IGRF); and (iv) an alphanumeric label (NAMES), molar Gibbs energy (G), the molar stoichiometric coefficients of the components for each phase (CP), and a flag indicating if the phase is a true phase or a pseudo-compound (IKP).

An example of input and input formatting is given in Table 3 of Appendix B. The data are for minerals in the water and silica saturated system CaO-Al_2O_3-FeO-H_2O-O_2 at 2.0 kbar and 600°C with oxygen fugacity fixed at magnetite-hematite buffer

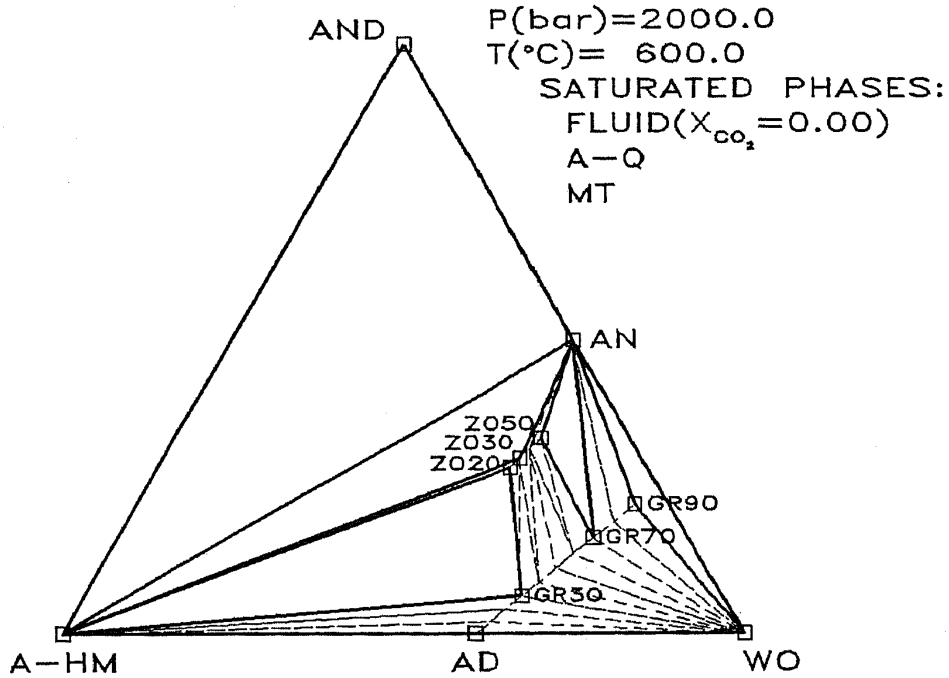


Figure 8. Computer generated graphical representation of output shown in Figure 7. Pseudo-invariant region boundaries, which approximate tielines, are shown as light dashed lines (because each tieline is drawn twice by the plotting device, dashed lines are occasionally superimposed in such a way as to appear continuous); true phase region boundaries are shown as heavy solid lines; invariant phase compositions are indicated by open squares (for phase notation see Table 2).

conditions. The constraints on the chemical potentials of silica, water, and oxygen restrict three degrees of freedom so the system is represented, by projection, in the ternary section CaO-Al_2O_3-FeO. The system contains two essentially binary solutions at these relatively oxidizing conditions, with endmember compositions (Table 2): grossularite-andradite and zoisite-epidote, which are subdivided, respectively, into pseudo-compounds symbolized by Gr, Gr_{90}, ..., Gr_{10}, Ad, and Zo, Zo_{90}, ..., Zo_{10}, Ep.

Once input has been read, control is passed to the subroutine SORT which classifies phases into subcompositions (step 2, Fig. 2). SORT also computes the mole fractions of the components in each phase. Control is then returned to the main program.

Evaluation of the Minimum Gibbs Energy Surface

The main program calls subroutine BOSSOP (step 3, Fig 2.) which evaluates Gibbs energy surfaces of progressively more complex subcompositions of c' components at a time by calls to SIMPLc' routines. The SIMPLc (step 4, Fig. 2) routines reference utility programs described in appendix A; those which are distinguished only by the suffix "c" serve identical functions but in systems of c components; e.g., SIMPL1 determines stable unary phases, SIMPL2 determines stable binary phases, etc.. The structure of SIMPL3 is shown by a flow chart in Figure 3, and is nearly identical to that of SIMPL4 and SIMPL5. The steps in this chart where each subprogram is called are indicated in appendix A. Unary and binary system phase relations are found by slightly different algorithms. Once the stable configurations of a system have been defined the routine MISCIB is called to establish if the configurations are invariant or pseudo-invariant. Pseudo-invariant regions are further classified according to whether they are part of a completely homogeneous phase region, or a heterogeneous phase region with or without immiscibility in one or more phases. This information is particularly useful for the generation of computer graphics. MISCIB can be manipulated to output the true variance of phase regions which do not include an immiscible solution. Control is then returned to the main program after which the phase configurations may be used for additional phase equilibrium calculations or, as in the present case, are directly output.

Output

Output consists of: (i) title information; (ii) a descriptive summary of the system; (iii) a listing of stable phase configurations; and, optionally, (iv) a file for computer graphics output. A glossary of variables used during output is presented in Appendix A. For binary systems stable phases are listed in order of their occurrence along the binary join. For more complex systems each (pseudo-)invariant assemblage is listed separately. Each assemblage is associated with an integer code (IASMBL) with the following significance: (1) the assemblage is invariant, (2) the assemblage is pseudo-invariant and heterogeneous, (3) the assemblage is homogeneous, and (4) the assemblage defines a portion of a solvus. Title information is printed by the routine OUTTIT and the phase assemblages are printed by OUTCHM.

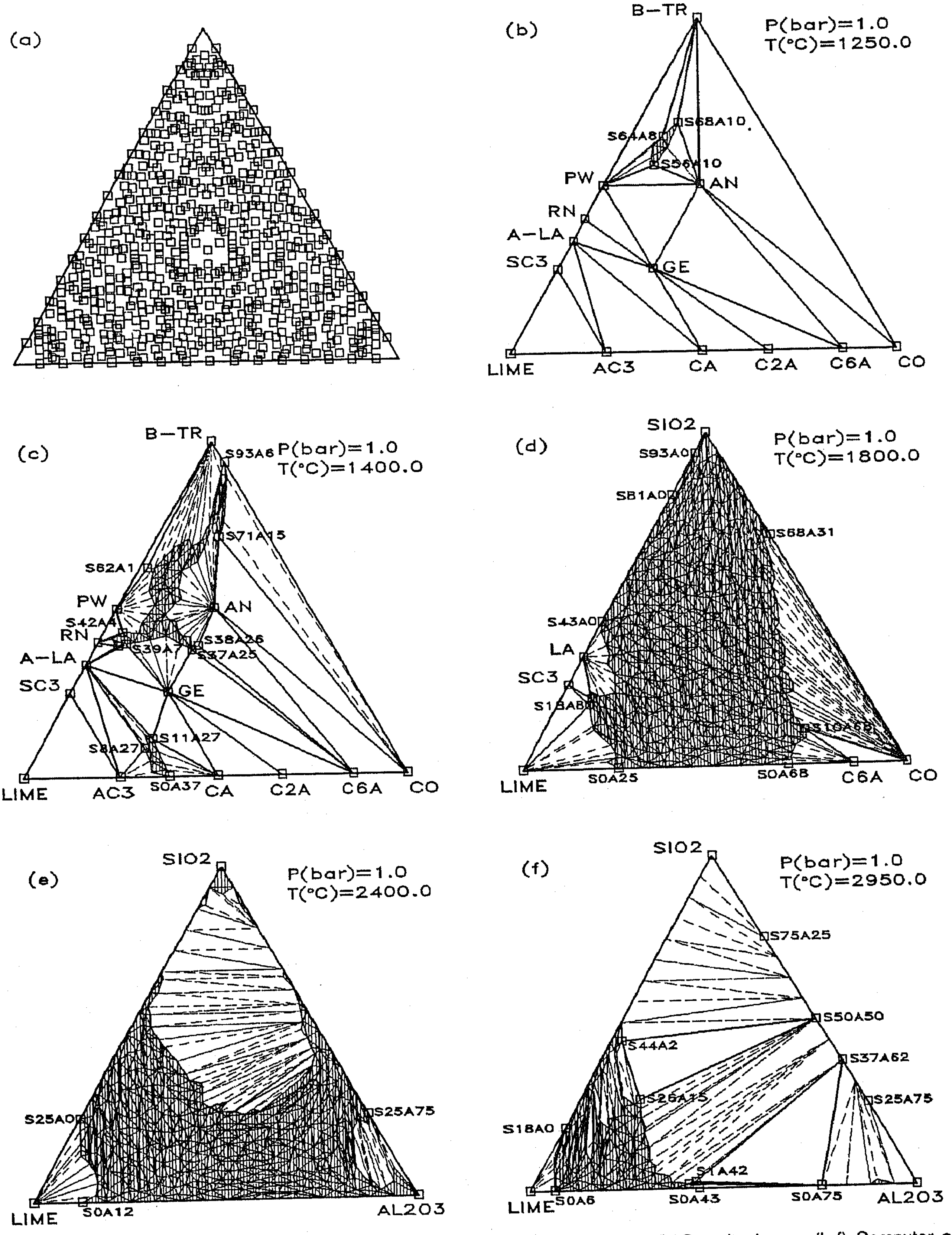


Figure 9. (a) Chemographic location of pseudo-compounds used to represent the CAS melt phase. (b-f) Computer calculated and drafted isothermal sections through the CAS system (for phase notation see Table 2). True phase region boundaries are shown as heavy solid lines; pseudo-invariant region boundaries, which approximate tielines, are shown by light dashed lines (because each tieline is drawn twice by the plotting device, dashed lines are occasionally superimposed in such a way as to appear continuous); and invariant phase compositions are indicated by open squares. The trivariant melt regions are lightly shaded, within melt regions pseudo-invariant region boundaries are drawn as heavy lines.

Output produced for the example in Table 3, Appendix B, is shown in Figure 7, and is summarized graphically on the composition phase diagram shown in Figure 8. The phase diagram consists of five invariant regions, e.g., Hm-Zo_{20}-Gr_{30}, and six univariant two phase regions represented by pseudo-invariant three phase assemblages such as Gr_{30}-Gr_{40}-Zo_{30}.

The chemographic relations in systems with four or fewer components are easily depicted graphically. The structure and an example of the file output for this purpose is given in Table 3 of Appendix B.

Applications

Direct applications of knowledge of stable chemographic relationships in chemical systems are manifold and widely known; their discussion here would therefore serve little purpose. Instead, attention is drawn to examples of less obvious uses: i.e., the construction of solvi and liquidi sections, and the computation of phase diagrams when one or two intensive variables are added to the parametric space of a system.

Solvi and Liquidi Sections

Many types of theoretical and empirical non-ideal solutions are currently in use (cf. 21). It is often difficult to judge whether the model assumptions and the models themselves are valid in a particular instance. A necessary, but not sufficient, test of validity is to determine if the predicted solvi, if any, are reasonable. This test is relatively simple to perform for binary solutions (e.g., 22, 23), but becomes exceedingly difficult for higher order solutions. Meijering (24, 25) approached the problem of ternary exsolution by analyzing the spinodal equations of regular solutions. Extension of his analysis for sub-regular solutions, however, has not been attempted, presumably because of the convoluted mathematics entailed. As an alternative, some workers have used approximate computational methods to explore the geometry of exsolution in complex solutions (e.g., 26, 27, 28, 29, 30). The Bounds algorithm provides a method of exploration which is both less costly and easier to apply.

The use of Bounds to construct isobaric-isothermal solvi and liquidi will be demonstrated for the system CaO-Al_2O_3-SiO_2 (CAS) at one bar total pressure. The data of Berman and Brown (31) for the CAS system were used without modification. Berman and Brown did not include mullite in their compilation because of difficulties they encountered in fitting the mullite liquidus. The diagrams presented here, which portray phase relations that are, in part, metastable with respect to mullite, are therefore intended only as pedagogical examples. The excess Gibbs energy of mixing in the melt was fit by Berman and Brown to a truncated Margules expansion of the form:

$$G_{excess} = \sum_{i=1}^{c-1} \sum_{j=i}^{c} \sum_{k=j}^{c} \sum_{\substack{l=j \\ l \neq i}}^{c} W_{ijkl} \, (X_i \, X_j \, X_k \, X_l). \qquad (13)$$

In order to apply Bounds, melt compositions were represented by 673 pseudo-compounds as shown in Figure 9a. Effects of solution in CAS liquidus and sub-liquidus phases, excepting mullite, can be neglected to a good approximation.

The approximate ternary minimum of the CAS system calculated with Bounds is at 1160 °C; in comparison, an exact numerical computation indicates Berman and Brown's equation of state is consistent with a minimum at 1124 °C. The equation of state is based on empirical data which includes an observation of the minimum at 1148 °C. Thus, the approximate calculations reproduce the observed phase relations with as much fidelity as is warranted by the quality of the fitting procedure used to obtain the equation of state. Figure 9b shows an isothermal CAS section at 1250 °C; at this temperature the divariant melt region extends along the An-Pw, An-Tr, and Tr-Pw cotectics, and is represented by Bounds as four pseudo-invariant assemblages: $S_{61}A_7$-$S_{64}A_8$-$S_{61}A_{11}$, $S_{61}A_7$-$S_{58}A_8$-$S_{61}A_{11}$, $S_{56}A_{10}$-$S_{58}A_8$-$S_{61}A_{11}$, and $S_{68}A_{10}$-$S_{64}A_8$-$S_{61}A_{11}$. At 1400 °C (Figure 9c) this region extends to the Al_2O_3-SiO_2 join and a second region has formed above the La-CA-AC3 eutectic. The jagged phase region boundaries in the sections shown in Figure 9 are an artifact of the pseudo-compound representation. It should be remembered that the true boundaries may occur anywhere between the representative pseudo-compounds of the melt on either side of identified boundaries. Boundaries can be smoothed with the algorithm by increasing the density of the pseudo-compounds in the vicinity of the boundaries in subsequent calculations. In figures 9c-e pseudo-invariant regions spanning several solvi in the melt phase are apparent. The boundaries of these regions give a general sense of the orientation of the binodal tielines for each solvus. The CAS system has only one known stable solvus which is largely restricted to the CaO-SiO_2 binary between temperatures of 1690 and 1910 °C (Fig. 9d). Therefore, the complex geometry of exsolution in the isothermal sections at 2400 and 2950 °C (Fig. 9e,f) most likely reflects erroneous instabilities in Berman and Brown's equation of state for CAS melts. Considering the large number of empirical parameters in the model and that the solvi, for the most part, occur at T-X conditions significantly different than those used to constrain the model, the instabilities are not surprising. It is interesting to note that the extraneous solvi were not reported by Berman and Brown for their computations over the same temperature ranges, nor were they noted by Barron (32) in his commentary on the model. The detection of the solvi here provides strong argument for, at least, exploratory use of computational schemes such as Bounds, which do not rely on assumed characteristics of chemical systems. In fact, the rather rough boundaries and conodes determined by Bounds may be all that is justified by the accuracy of the solution models in use for many applications. When this is not the case, the data from Bounds can be used to supply the initial estimates necessary for more refined computations such as those detailed by Pelton et al. (30). Bounds has the additional advantage that binodal immiscibility is not assumed; thus, trinodal, and more complex, exsolution regions can be found directly. More precise algorithms typically consume 1-12 s CPU time per binodal tieline (26, 30); CPU time for the diagrams in Figures 9b-f was, on average, 0.20 s per diagram, and was only slightly dependent on the number of conodes determined.

Mixed Variable and Schreinemakers-Type Phase Diagrams

Mixed variable phase diagrams (16, 33), such as T-X diagrams, in which potential variables are plotted against extensive

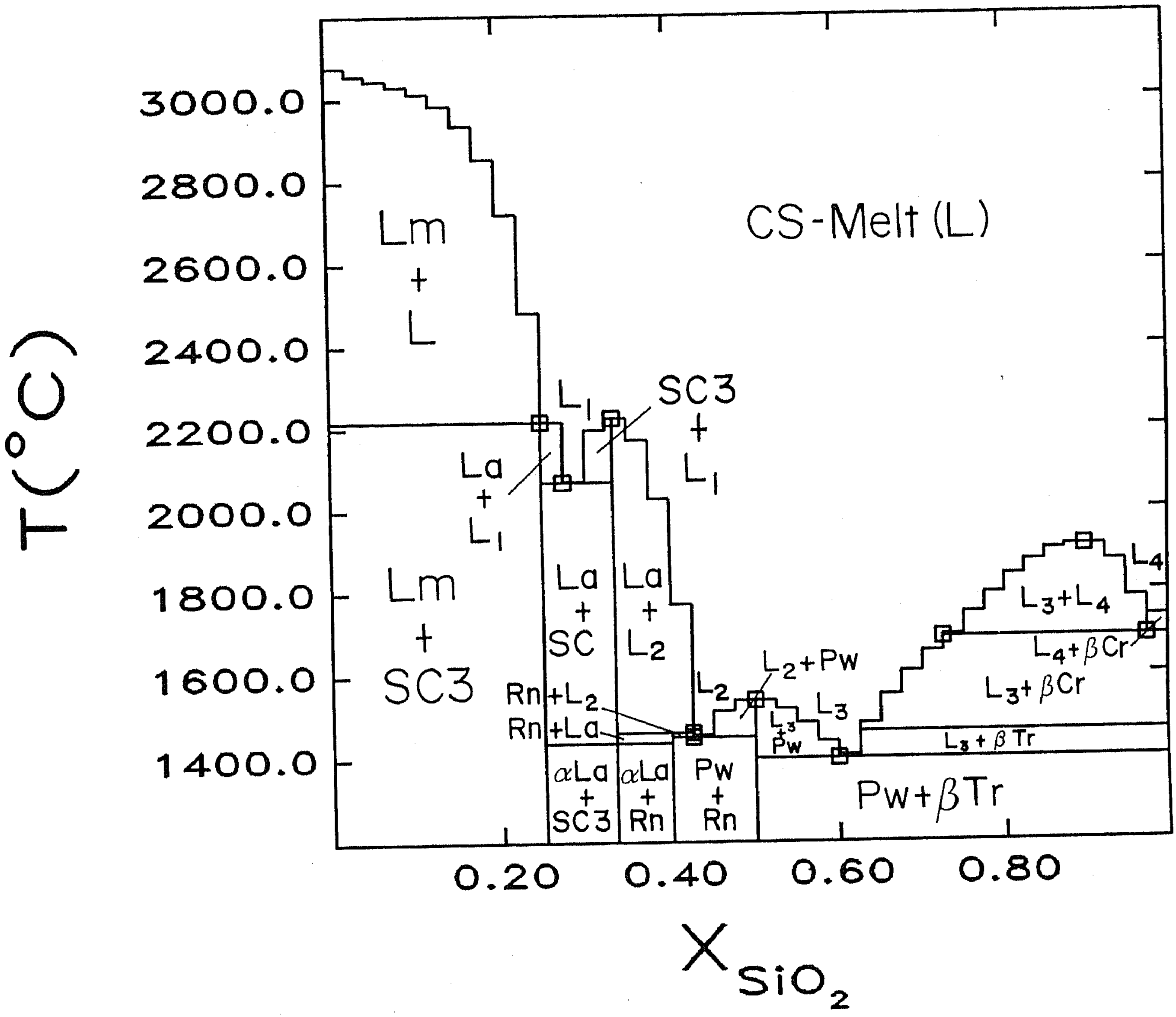


Figure 10. T-X diagram for the CS system at 1 bar pressure. The diagram was constructed using an automated version of the algorithm discussed in the text. The calculation was made with the data of Berman and Brown (31) (for phase notation see Table 2).

or molar variables can be thought of as a series of infinitely closely spaced iso-potential sections. Were such a series computed with Bounds, many of the sections would be redundant, as the piecewise linear approximation results in discontinuous changes in the geometry of thermodynamic surfaces as a function of potential variables. Therefore, it is only necessary to locate conditions where the geometric changes, which can be described by a chemical reaction, take place to establish the topology of phase diagrams consistent with the approximations of Bounds. In a system with one potential variable this can be done by a procedure consisting of four steps: (i) computation of the iso-potential surface; (ii) incrementing the potential variable until the original configuration of the surface becomes metastable; (iii) determination of the reaction relating the stable and metastable configurations; and (iv) solution for the equilibrium conditions of the reaction. Because all point phase configurations are invariant in iso-potential sections, the only reactions in a point phase system with one potential variable are invariant or pseudo-invariant. This is illustrated for the system CaO-SiO₂ as a function of temperature in Figure 10. With increasing temperature the first reaction occurs at the Pw-Tr eutectic:

$$Pw + Tr = S_x. \qquad (R1)$$

At higher temperatures the univariant liquidi are defined by series of pseudo-invariant reactions of the form:

$$Pw + S_x = S_{x-dx}. \qquad (R2)$$

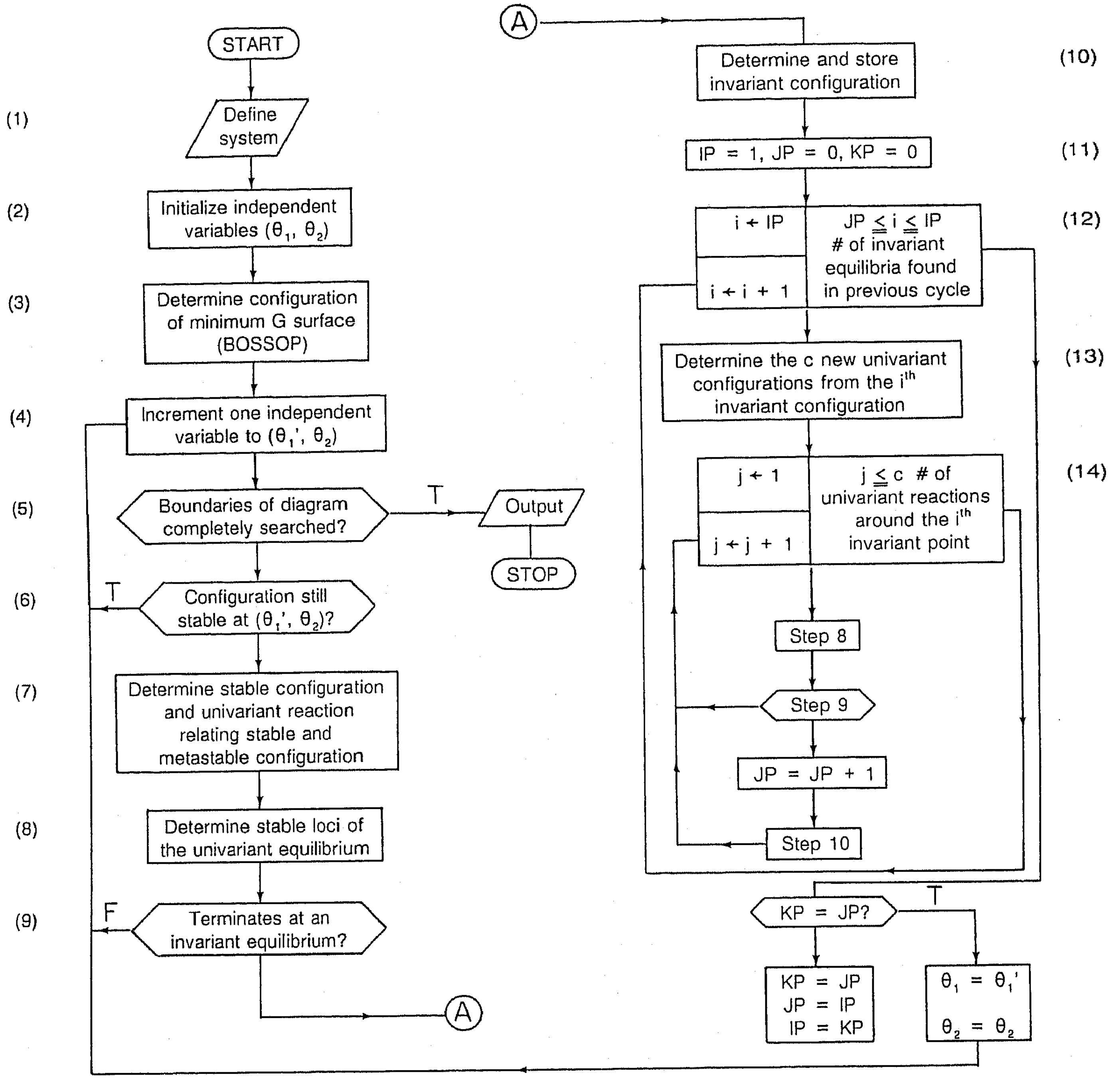

START
(1) Define system
(2) Initialize independent variables (θ1, θ2)
(3) Determine configuration of minimum G surface (BOSSOP)
(4) Increment one independent variable to (θ1', θ2)
(5) Boundaries of diagram completely searched?
T
Output
STOP
(6) Configuration still stable at (θ1', θ2)?
T
(7) Determine stable configuration and univariant reaction relating stable and metastable configuration
(8) Determine stable loci of the univariant equilibrium
(9) Terminates at an invariant equilibrium?
F
A
A
(10) Determine and store invariant configuration
(11) IP = 1, JP = 0, KP = 0
(12) i ← IP
i ← i + 1
JP ≤ i ≤ IP # of invariant equilibria found in previous cycle
(13) Determine the c new univariant configurations from the i^th invariant configuration
(14) j ← 1
j ← j + 1
j ≤ c # of univariant reactions around the i^th invariant point
Step 8
Step 9
JP = JP + 1
Step 10
KP = JP?
T
KP = JP
JP = IP
IP = KP
θ1 = θ1'
θ2 = θ2

Figure 11. Flow chart of an algorithm for constructing Schreinemakers-type phase diagrams for two intensive variables designated θ_1 and θ_2.

An inherent difficulty of this procedure is that the fixed spacing of point phases may allow misrepresentation of an equilibrium. For example, the peritectic reaction:

$$La = Lm + S_x, \tag{R3}$$

appears incorrectly in Figure 10 as the singular point of an irregular section, i.e., a section taken through an invariant point (14 p. 287). This occurs because the melt pseudo-compound closest, in composition, to the true peritectic coincides with larnite.

The algorithm for one potential variable may be extended for phase equilibrium computations within a coordinate frame defined by two potential variables, θ_1 and θ_2. The extended algorithm is outlined by the flow chart shown in Figure 11. The significant modifications of the simple algorithm discussed in the previous paragraph are: (i) configurations of c + 1 phases are univariant; (ii) the trajectory of univariant equilibria in the potential variable coordinate frame must be determined (step 8, Fig. 11); and (iii) it is necessary to identify and define conditions of any topologically related invariant and univariant equilibria

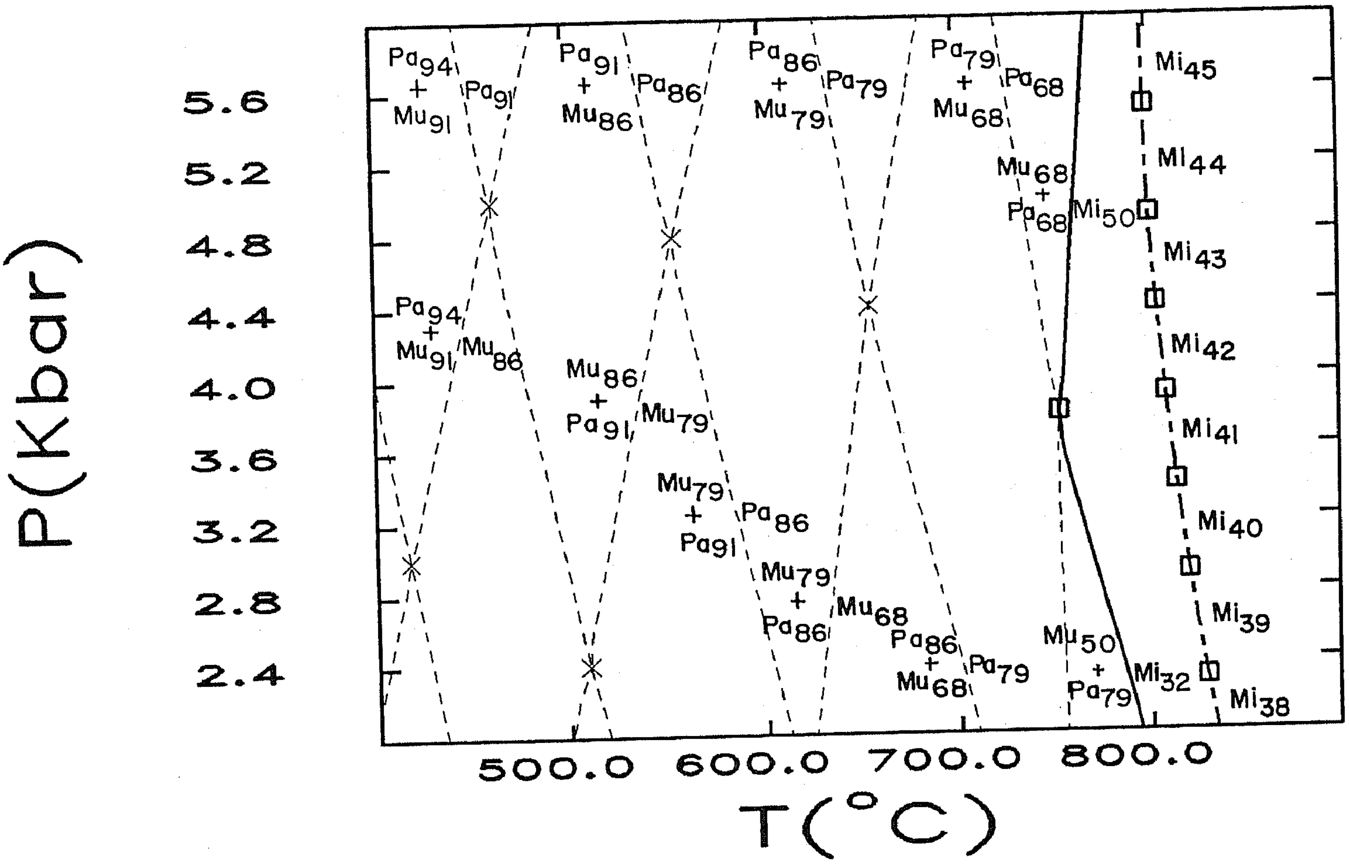


Figure 12. Schreinemakers P-T phase diagram of the muscovite-paragonite solvus (for phase notation see Table 2). Pseudo-univariant are shown as light dashed curves. To simplify the diagram, some of the pseudo-univariant curves are unlabeled; the corresponding equilibria can be determined from the labeled curves by the Schreinemakers technique (34). The critical curve of the solvus is shown as a heavy solid curve. The critical locus determined by a more detailed computation is shown by a heavy dashed curve labelled by the mica critical compositions.

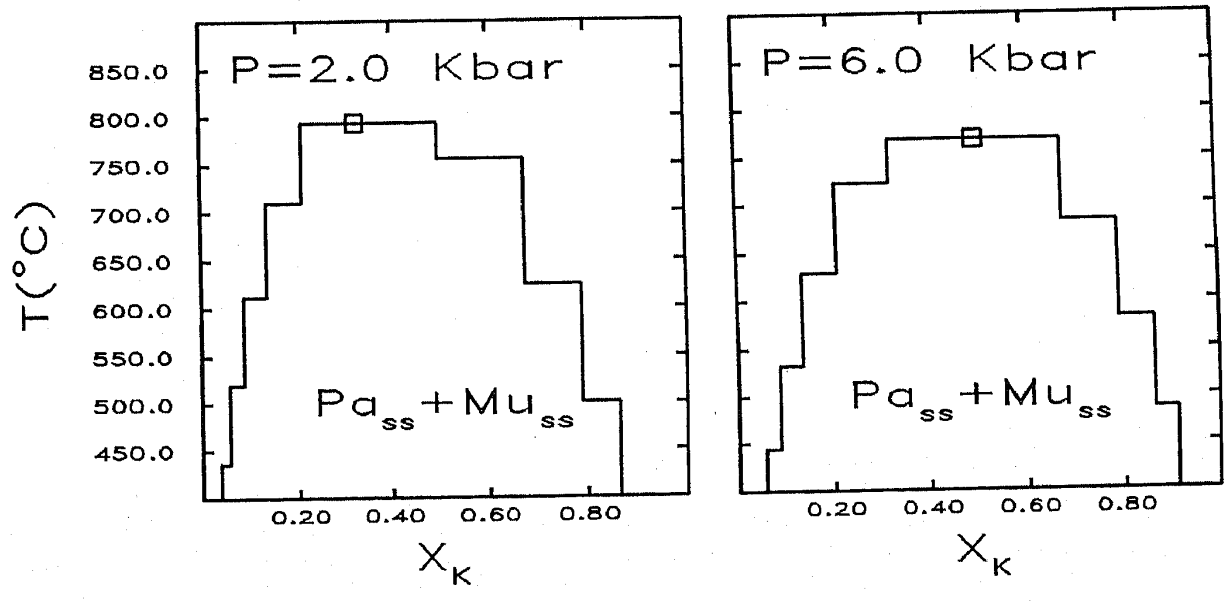


Figure 13. Isobaric sections at two and six kilobars of the phase diagram shown in Figure 12, critical compositions are indicated by open squares

(steps 10-14, Fig. 11).

The phase relations of a multidimensional parametric space can be shown by homeomorphic projection in a two dimensional diagram with axes defined by any two independent variables. The topology of such a diagram is most easily interpreted when the axes correspond to potential variables, the nature of the variables on the projected axes (i.e., potential, molar, or extensive) is irrelevant to the topology of the final diagram (16). Schreinemakers (34) originally developed this method for μ-X-n projections; more recently, Hillert (16) suggested generalizing the Schreinemakers scheme for P-V and T-S projections as well. The general algorithm set out in Figure 4, which determines the coordinates of equilibria in terms of two potential variables, θ_1 and θ_2, is, therefore, easily adapted for the construction of Schreinemakers-type phase diagrams.

If a system contains only point phases, then only invariant, univariant and divariant equilibria are possible and these project, in Schreinemakers diagrams, as points, curves and regions, respectively. Consequently, the topology of the univariant curves, and knowledge of the corresponding equilibria, completely define all the remaining equilibria of the system. One drawback of the Schreinemakers method is its unwieldiness when applied to systems which contain solutions and therefore phase regions of variance greater than two (13 p. 16; 16). In this case, regions are not distinguished because they are of greater dimension than the diagram. This difficulty may be overcome by contouring the low variance phase regions, which separate higher variance regions by isopleths. Isoplethal contours can be thought of as pseudo-univariant reactions between a homogeneous phase of incrementally different composition. The subdivision of solutions into pseudo-compounds, necessary to apply Bounds, is thus ideally suited for locating such pseudo-univariant equilibria.

The polythermal-polybaric phase diagram of the binary $KAl_3Si_3O_{10}(OH)_2$-$NaAl_3Si_3O_{10}(OH)_2$ alkali mica (Table 2) solvus shown in Figure 12 illustrates a simple application of the algorithm outlined in Figure 11. A third degree truncated Margules expansion fit to the excess properties of the micas (35) was used in the calculations. The curves drawn as light lines correspond to the P-T loci of pseudo-univariant equilibria. In a binary system, pseudo-univariant equilibria define conditions under which the system is indifferent between two point phase configurations which represent the same divariant phase region. The pseudo-univariant equilibria of Figure 12 with positive Clapeyron slopes approximate isoplethal contours of the K-rich limb of the mica solvus and may be expressed as reactions of the form:

$$Mu_x + Pa_y = Mu_{x\text{-}dx}, \tag{R4}$$

and those with negative slopes define the Na-rich limb and may be written:

$$Pa_x + Mu_y = Pa_{x\text{-}dx}. \tag{R5}$$

These relations may be clarified further by comparison of Figure 12 with the isobaric sections shown in Figure 13. The negative Clapeyron slope of equilibrium (R5), which implies increasing stability of Na-mica relative to K-mica with increasing pressure, is crystallochemically plausible considering the relative ionic radii of Na and K.

The critical curve of the solvus (shown as a heavy line on Fig. 12) is represented by a univariant equilibrium between the subcritical micas and supercritical mica:

$$Pa_x + Mu_y = Mi_z. \tag{R6}$$

Variation in the critical composition occurs discontinuously when the critical curve is intersected by pseudo-univariant curves where the critical composition shifts to ~Mi_{32} to ~Mi_{50} with increasing pressure. For the sake of simplicity, the calculation was done with relatively coarse subdivision of the solution into pseudo-compounds. Any desired resolution can be obtained by decreasing the compositional increment between pseudo-compounds. For comparison, the critical curve from a computation with better compositional resolution (one mole percent increments) is shown as a heavy dashed curve on Figure 12. The CPU time for the detailed calculation was 1 s as opposed to less than 0.25 s for the coarse calculation. It should be noted that the mica phase relations are metastable with respect to phases which do not lie in the $KAl_3Si_3O_{10}(OH)_2$-$NaAl_3Si_3O_{10}(OH)_2$ binary over much of the P-T region shown in Figure 12.

Although not explicitly demonstrated here, it should be apparent that the trajectories of cotectic equilibria in P-T-μ diagrams can be established in the same way as the critical curves of Figure 12. It is hoped that the foregoing examples have demonstrated the utility of piecewise linear representations of thermodynamic surfaces. Phase diagrams constructed using such representations effectively and efficiently show the salient features of multicomponent phase equilibria. Algorithms and computer programs for such constructions have been developed (36) and will be discussed in greater detail in a forthcoming communication.

Concluding Remarks

The Bounds algorithm makes it possible to compute stable phase assemblages as a function of the composition of a system, and can be extended to incorporate additional independent extensive variables such as S and V. The algorithm is extremely efficient when applied to systems which contain phases with fixed stoichiometry. Solution phases can be accommodated by representing their compositional variation by incrementally spaced point phases, or "pseudo-compounds". The approximate representation of solutions by pseudo-compounds, while admittedly crude, has several desirable qualities; these include: (i) The thermodynamic surface of an approximated system is a piecewise linear simplicial complex. The geometry of this complex is specified by the location of the vertices of each simplex, which correspond to the phases of each stable assemblage of the system. (ii) Because the surface is linear, changes in its geometry define singularities which project as curves on Schreinemakers-type phase diagrams. (iii) Compositions of coexisting solutions in high variance phase regions can be estimated from the pseudo-compound assemblages approximating the solutions. (iv) Any desired precision on the location of phase boundaries can be obtained by increasing the number of pseudo-compounds in the vicinity of the boundaries.

The strengths of the Bounds algorithm are: efficiency; independence from the equations of state used for the phases of a system; and that only minimal assumptions are required as to the equilibrium configuration of a system. Because Bounds determines continuous iso-potential thermodynamic surfaces, it can be used to submit equations of state to tests of greater rigor than can be easily achieved with other algorithms. It is expected that Bounds may be used profitably for exploratory analysis of the phase relations of complex systems. Once these relations have been approximated, they may be subsequently refined with mathematically more sophisticated procedures. The primary limitation of Bounds is the ability of a user to assimilate the volume of information generated. This limitation can, in part, be overcome by the use of graphical representation of the output. Alternatively, the algorithm can be used in tandem with additional automated algorithms, as in the construction of Schreinemakers-type phase diagrams, to present the information in a more easily interpreted form.

Acknowledgements

This work was made possible by a grant from the Earth Sciences Section of the National Science Foundation (NSF Grant EAR-8307682) to DMK, and computer funds provided by The Pennsylvania State University. Much of this work was undertaken while JADC was the recipient of fellowships from the Owens Corning Fiberglas Corporation and the Graduate School of the Pennsylvania State University. The authors are grateful to Robert G. Berman, Alex M. Nitkiewicz, Karl E. Spear, and Charles J. Kaiser for their comments on aspects of this paper. The authors would also like to thank an anonymous reviewer for making them aware of a particularly lucid discussion of phase diagram topology by Hillert (16).

References

1. F. J. Zeleznik and S. Gordon, Ind. Eng. Chem. 60, 27 (1968).

2. F. Van Zeggeren and S. H. Storey, The Computation of Chemical Equilibrium, Cambridge Univ. Press, Cambridge (1970).

3. G. Eriksson, Acta. Chem. Scand. 25, 2651 (1971).

4. T. H. Brown and B. J. Skinner, Am. Jour. Sci. 274, 961 (1974).

5. G. Eriksson, Chem. Script. 8, 100 (1975).

6. M. Hillert, Physica 103B and C, 31 (1981).

7. E. Schendler, CALPHAD 8, 265 (1984).

8. G. Eriksson and K. Hack, CALPHAD 8, 15 (1984).

9. A. G. Turnbull, CALPHAD 7, 137 (1983).

10. W. T. Thompson, A. D. Pelton and C. W. Bale, CALPHAD 7, 113 (1983).

11. H. L. Lukas, J. Weiss, E.-Th. Henig, CALPHAD 6, 229 (1982).

12. P.-L. Lin, A. D. Pelton and C. W. Bale, CALPHAD 4, 47 (1980).

13. L. S. Palatnik and A. I. Landau, Phase Equilibria in Multicomponent Systems, Holt, Rinehart and Winston, Inc., New York (1964).

14. D. S. Korzhinskii, Physicochemical Basis of the Analysis of the Paragenesis of Minerals, Consultants Bureau, New York (1959).

15. J. W. Gibbs, The Collected Works, vol. 1, Yale University Press, New Haven (1957).

16. M. Hillert, Int. Met. Rev. 30, 45 (1985).

17. A. Prince, Metall. Rev. 8, 213 (1963).

18. J. E. Ricci, The Phase Rule and Heterogeneous Equilibrium, D. Van Nostrand Co., Inc., New York (1951).

19. M. Zhao, CALPHAD 7, 185 (1983).

20. J. F. P. Hudson, Piecewise Linear Topology, W. A. Benjamin, Inc., New York (1969).

21. M. Hillert, CALPHAD 4, 1 (1980).

22. R. J. Charles, J. Am. Ceram. Soc. 50, (1967).

23. R. F. Brebrick, CALPHAD 2, 17 (1978).

24. J. L. Meijering, Phillips Res. Rep. 5, 333 (1950).

25. J. L. Meijering, Phillips Res. Rep. 6, 183 (1951).

26. J. F. Counsell, E. B. Lees and P. J. Spencer, Met. Sci. J. 5, 210 (1971).

27. L. M. Barron, Geochim. Cosmochim. Acta 45, 661 (1981).

28. S. K. Saxena, Thermodynamics of Rock Forming Crystalline Solid Solutions, vol. 8, Springer-Verlag, New York (1973).

29. W. C. Luth and P. M. Fenn, Am. Mineral. 58, 1009 (1973).

30. A. D. Pelton, C. W. Bale and M. Rigaud, Z. Metalkde. 68, 135 (1977).

31. R. G. Berman and T. H. Brown, Geochim. Cosmochim. Acta 48, 661 (1984).

32. L. M. Barron, Geochim. Cosmochim. Acta 49, 611 (1985).

33. V. H. Schmalzreid and A. D. Pelton, Ber. Bunsenges Phys. Chm. 77, 90 (1973).

34. F. A. H. Schreinemakers, In-, Mono-, and Divariant Equilibria, Papers by F. A. H. Schreinemakers, vol. 2, University Park, The Pennsylvania State University (1965).

35. N. D. Chatterjee and E. Froese, Am. Mineral. 60, 985 (1975).

36. J. A. D. Connolly and D. M. Kerrick, EOS Trans. 65, 287 (1984).

37. S. D. Conte and C. de Boor, Elementary Numerical Analysis, McGraw-Hill Co., New York (1980).

APPENDIX A: Glossary of Subprograms and Input/Output (I/O) Variables

FORTRAN variable types are given where appropriate. Steps where subprograms are referenced in the flow charts included in the text are indicated parenthetically.

ABLOAD - Subroutine to initialize the matrices $\tilde{G}$ and $\underset{\sim}{\nu}$ of equation (5) and solve for the vector $\underset{\sim}{\mu}$. External references: FACTOR and SUBST. (Steps 6, 10, 27, 30, Fig. 3).

ASSDc - Subroutine to determine if a stable assemblage of c phases has been identified earlier in the execution of the program. If the assemblage is new, the c-1 boundaries of the corresponding phase region are also tested for uniqueness; previously found boundaries are flagged. (Steps 11-15, 31, Fig. 3).

BOSSOP - Subroutine to call SIMPLc routines to define the minimum Gibbs energy surface. After this surface has been defined BOSSOP calls the routine MISCIB to determine the variance of phase regions, and to locate solvi. External references: SIMPLc and MISCIB. (Step 3, Fig. 2).

BOUNDc - Subroutine to obtain the upper and lower triangular decompositions of the matrix $\underset{\sim}{\nu}^t$ from equation (7). The decompositions of $\underset{\sim}{\nu}^t$ are used to solve equation (7) by routine ITESTc. External references: FACTOR. (Steps 21, 26, Fig. 3).

CNAME(I) - Alphanumeric label of the i^th component (I/O, C*5).

CP(J,I) - ν_{ji} (input, R*8).

DGPHc - Function to test the conditional (6). DGPHc takes on a negative value if the conditional is false, and is positive or zero otherwise. (Steps 8, 30, Fig. 3).

FACTOR - Subroutine to find the upper, $\tilde{U}$, and lower, $\tilde{L}$, triangular decompositions of a matrix $\tilde{A}$, if $\tilde{A}$ is nonsingular. The algorithm is modified from Conte and de Boor (37).

G(J) - G^m_j (input, R*8).

IASMBL(M) - Flag characterizing the m^th stable phase assemblage, determined by the routine MISCIB. IASMBL may take on four values with the following significance: (1) The assemblage defines an invariant phase region. (2) The assemblage defines a heterogeneous pseudo-invariant region in which none of the stable solutions are immiscible. (3) The assemblage defines a completely homogeneous region. (4) Immiscibility occurs in one or more of the stable solutions in the phase region. (Output I*2).

IBVCT(K) - Number of phases on the k^th binary join (output, I*2).

ICFCT - Number of quinary phase configurations (output, I*2).

ICP - c (I/O, I*2).

IDBF(K,L) - Index, j, of the l^th stable phase on the k^th binary join (output, I*2).

IDC(L) - Index, j, of the stable phase in the l^th unary subsystem (output, I*2).

IDCF(M,L) - Index, j, of the l^th stable phase in the m^th stable quinary assemblage (output, I*2).

IDQF(K,M,L) - Index, j, of the l^th stable phase in the m^th assemblage on the k^th quaternary join (output, I*2).

IDTF(K,M,L) - Index, j, of the l^th stable phase in the m^th assemblage on the k^th ternary join (output, I*2).

IGRF - Flag indicating whether a file for graphics applications is to be written, if this flag is assigned a value of 1 the file is not written, otherwise the file is output (input, I*2).

IKP(J) - Flag assigned a value of 0 if the j^th phase is not a pseudo-compound. If the phase is a pseudo-compound IKP is given an integer value identifying the solution phase; e.g., if a system contains three solution phases, IKP for the pseudo-compounds corresponding to each of the solutions might be assigned values of 1, 2, and 3 (input, I*2).

IPHCT - Number of possible phases defined in a chemical system (I/O, I*2).

IPOINT - Number of true point phases defined in a chemical system (I/O, I*2). IPHCT - IPOINT = the number of pseudo-compounds in the system.

IQFCT(K) - Number of stable assemblages on the kth quaternary join (output, I*2).

ITESTc - Function to determine if a given phase lies within the compositional simplex defined by a phase region; this is done by solving for the vector $\tilde{\alpha}$ of equation (7), and testing the conditional (8). ITESTc may take on three values: (0) if the phase is within the region; (1) if the phase is outside the region; and (2) if the phase is on a boundary of the region. Parameters are initialized for each region by the routine BOUND. (Step 24, Fig. 3).

ITFCT(K) - Number of stable assemblages on the kth ternary join (output, I*2).

ITITLE - Number of title cards (I/O, I*2).

MISCIB - Subroutine to determine variance of phase regions, and to test for immiscibility of solutions. MISCIB outputs the array IASMBL and references the routines FACTOR and SUBST to solve equation (10). (Step 6, Fig. 2).

NAME(J) - Alphanumeric label of the jth phase (I/O, C*8).

OUTCHM - Subroutine to output the stable phase assemblages for the defined chemical system. (Step 7, Fig. 2).

OUTTIT - Subroutine to output title information for each calculation. (Step 7, Fig. 2).

SIDc - Function to determine if a phase q lies on the same side of a phase region boundary as another phase p, where p is a phase known to coexist with the boundary assemblage. SIDc takes on the value of the parameter:

$$\zeta_q = (\alpha_o + \sum_{i=1}^{c-1} \alpha_i \nu_{qi}) / \gamma_p, \qquad (14)$$

where i indexes the phases of the boundary assemblage. If ζ_q is positive both p and q are on the same side of the boundary. If γ_q is negative q is on the opposite side of the boundary from p. Parameters for SIDc are initialized by SLOPc. (Steps 23, 29, Fig. 3).

SIMPLc - Subroutine to define the minimum Gibbs energy surfaces of the $\binom{c}{c'}$ subcompositions of c' components of a c components system. External references: BOUNDc, ITESTc, ABLOAD, ASSDc, OUTCHM, and DGPHc. (Steps 4, 5, Fig. 2).

SLOPc - Subroutine to determine the equation describing the loci of a linear phase boundary in composition space:

$$n_c = \alpha_0 + \sum_{i=1}^{c-1} \alpha_i n_i, \qquad (15)$$

where i indexes the c-1 phases defining the boundary. SLOPc also computes the parameter:

$$\gamma_p = \alpha_0 + \sum_{i=1}^{c-1} \alpha_i \nu_{pi}, \qquad (16)$$

where p indexes a phase, which, in combination with the boundary assemblage, defines a stable phase region. The parameters α_0, ..., α_{c-1} and λ_p are used by SIDc. External references: SUBST and FACTOR. (Step 19, Fig. 3).

SORT - Subroutine to classify each phase into the simplest subcomposition that can be formed from the c components of a chemical system. (Step 2, Fig. 2).

SUBST - Subroutine to solve $\tilde{L}\,\tilde{Y} = \tilde{B}$ for $\tilde{Y}$ and then $\tilde{U}\,\tilde{X} = \tilde{Y}$ for $\tilde{X}$, given the matrices $\tilde{U}$ and $\tilde{L}$ from FACTOR and a vector B. The algorithm is modified from Conte and de Boor (37).

TNAME - Title information (I/O, C*72).

APPENDIX B: Examples of Input and Output Files

TABLE 3

Example of Input for Bounds

Copy of the input file used to generate the output shown in Figure 7. Variable names and the FORTRAN read formats are indicated parenthetically.

```
(ITITLE, unformatted)
            18
((TNAME(I),I=1,ITITLE), A72)
        THERMODYNAMIC DATA BASE FROM: HELGESON ET AL., 1978
        FLUID EQUATION OF STATE: MRK, PARAMETERS FROM CONNOLLY AND BODNAR, 1983

        COMPONENTS WITH CONSTRAINED CHEMICAL POTENTIALS:

        SIO2    O2      H2O     CO2

        CONSTRAINTS FOR THE CALCULATION:

                  ( 1) TEMPERATURE (C): 600.00
                  ( 2) PRESSURE (KBAR):  2.000
                  ( 3) H2O-CO2 FLUID SATURATED, XCO2: 0.00
                  ( 4) OXYGEN FUGACITY BUFFER: MT-HM
                  ( 5) SILICA BUFFER: A-Q

        EXCLUDED PHASES:

        CA-AL-PX  LIM       B-CR      CZO       BOE       A-QFM     MT-WU
(ICP, unformatted)
                     3
((CNAME(I),I=1,ICP), 5(A5,1X))
       CAO   A1203 FEO
(IGRF, unformatted)
                     0
(IPHCT, unformatted)
                    50
((NAMES(I),I=1,IPHCT), 8(A8,1X))
       LIME     AR       WAIRAKIT LAUM     GR       WO       CC       AN
       CO       AND      KY       SIL      GE       MA       ZO       KAO
       GIB      DIA      PYR      LAWS     A-PR     B-PR     FA       HED
       SID      A-MT     FS       AD       A-HM     EP       B-EP     FEO
       GR100    GR200    GR300    GR400    GR500    GR600    GR700    GR800
       GR900    ZO100    ZO200    ZO300    ZO400    ZO500    ZO600    ZO700
       ZO800    ZO900
((IKP(I),I=1,IPHCT), unformatted)
         0  0  0  0  1  0  0  0  0  0  0  0  0  0  2  0  0  0  0  0  0  0  0  0
         0  0  0  1  0  0  2  0  1  1  1  1  1  1  1  1  1  2  2  2  2  2  2  2
         2  2
((G(I),I=1,IPHCT), unformatted)
          -640635.          0.  -2384139.  -2341141.  -3842856.   -726306.         0.
        -2391881.  -1628731.  -1632595.  -1630384.  -1632527.  -3071454.  -4012481.
        -3943936.  -1606533.   -784505.   -805505.  -1616942.  -2341013.  -3107151.
        -3107527.   -617890.  -1042731.         0.  -3369985.   -308622.  -4450942.
        -2246657.  -4251365.  -4253476.   -303169.  -4397213.  -4340223.  -4281820.
        -4222365.  -4161995.  -4100748.  -4038586.  -3975371.  -3910745.  -4227242.
        -4198834.  -4169484.  -4139432.  -4108770.  -4077524.  -4045668.  -4013110.
        -3979610.
(CP(I,J),J=1,ICP),I=1,IPHCT), unformatted)
        1.0000  0.0     0.0     1.0000  0.0     0.0     1.0000  1.0000  0.0
        1.0000  1.0000  0.0     3.0000  1.0000  0.0     1.0000  0.0     0.0
        1.0000  0.0     0.0     1.0000  1.0000  0.0     0.0     1.0000  0.0
        0.0     1.0000  0.0     0.0     1.0000  0.0     0.0     1.0000  0.0
        2.0000  1.0000  0.0     1.0000  2.0000  0.0     2.0000  1.5000  0.0
        0.0     1.0000  0.0     0.0     0.5000  0.0     0.0     0.5000  0.0
        0.0     1.0000  0.0     1.0000  1.0000  0.0     2.0000  1.0000  0.0
        2.0000  1.0000  0.0     0.0     0.0     2.0000  1.0000  0.0     1.0000
        0.0     0.0     1.0000  0.0     0.0     3.0000  0.0     0.0     1.0000
```

```
3.0000  0.0     2.0000  0.0     0.0     2.0000  2.0000  1.0000  1.0000
2.0000  1.0000  1.0000  0.0     0.0     1.0000  3.0000  0.1000  1.8000
3.0000  0.2000  1.6000  3.0000  0.3000  1.4000  3.0000  0.4000  1.2000
3.0000  0.5000  1.0000  3.0000  0.6000  0.8000  3.0000  0.7000  0.6000
3.0000  0.8000  0.4000  3.0000  0.9000  0.2000  2.0000  1.0500  0.9000
2.0000  1.1000  0.8000  2.0000  1.1500  0.7000  2.0000  1.2000  0.6000
2.0000  1.2500  0.5000  2.0000  1.3000  0.4000  2.0000  1.3500  0.3000
2.0000  1.4000  0.2000  2.0000  1.4500  0.1000
```

TABLE 4

Example of an Output File Generated by Bounds for Graphics Applications

Copy of the data file used to generate the diagram shown in Figure 8. Variable names and FORTRAN format are indicated parenthetically. The data is for a chemical system with three inert components, in systems with differing numbers of components the variables ITFCT and IDTV are replaced by the appropriate counters and arrays (e.g., IBVCT, IDBV; IQFCT, IDQF; and ICFCT, IDCF for binary, quaternary, and quinary systems, respectively).

```
(ICP,IPHCT,ITFCT, 20(I3,1X))
          3  49  29
((NAMES(I),I=1,IPHCT), 10(A8))
        LIME    AR      WAIRAKITLAUM    GR      WO      CC      AN      CO      AND
        KY      SIL     GE      MA      ZO      KAO     GIB     DIA     PYR     LAWS
        A-PR    B-PR    FA      HED     SID     A-MT    FS      AD      A-HM    EP
        B-EP    FEO     GR100   GR200   GR300   GR400   GR500   GR600   GR700   GR800
        GR900   ZO100   ZO200   ZO300   ZO400   ZO500   ZO600   ZO700   ZO800
(((X(I,J),J=1,ICP-1),I=1,IPHCT), 11(F6.4,1X))
        1.0000 0.0000 1.0000 0.0000 0.5000 0.5000 0.5000 0.5000 0.7500 0.2500 1.0000
        0.0000 1.0000 0.0000 0.5000 0.5000 0.0000 1.0000 0.0000 1.0000 0.0000 1.0000
        0.0000 1.0000 0.6667 0.3333 0.3333 0.6667 0.5714 0.4286 0.0000 1.0000 0.0000
        1.0000 0.0000 1.0000 0.0000 1.0000 0.5000 0.5000 0.6667 0.3333 0.6667 0.3333
        0.0000 0.0000 0.5000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.6000
        0.0000 0.0000 0.0000 0.5000 0.2500 0.5000 0.2500 0.0000 0.0000 0.6122 0.0204
        0.6250 0.0417 0.6383 0.0638 0.6522 0.0870 0.6667 0.1111 0.6818 0.1364 0.6977
        0.1628 0.7143 0.1905 0.7317 0.2195 0.5063 0.2658 0.5128 0.2821 0.5195 0.2987
        0.5263 0.3158 0.5333 0.3333 0.5405 0.3514 0.5479 0.3699 0.5556 0.3889
(((IDTV(1,I,J),J=1,3),I=1,ITFCT), 20(I3,1X))
          6   8  41   8  10  29   6  28  33  28  29  33   6  41  40   8  41  40   8  29
         44   6  33  34  29  33  34   6  40  39   8  40  39   8  44  45  29  44  43   6
         34  35  29  34  35   6  39  38   8  39  46   8  45  46  44  45  36  29  43  35
         44  43  35   6  35  36   6  38  37  39  38  46  45  46  38  44  36  35  45  36
         37   6  36  37  38  37  45
((IASMBL(I),I=1,ITFCT), 20(I3,1X))
          1   1   2   2   2   2   1   2   2   2   2   2   2   2   2   2   1   2   2   1
          2   2   2   2   2   2   2   2   2
```

APPENDIX C: Program Listing

```
C                  ------------------------------
C                  *                            *
C                  *          BOUNDS            *
C                  *                            *
C                  ------------------------------
C
C BOUNDS IS A PROGRAM FOR EVALUATING A PIECEWISE LINEAR APPROXIMATION
C OF THE MINIMUM GIBBS FREE ENERGY SURFACE OF CHEMICAL SYSTEMS WITH
C UP TO FIVE COMPOSITIONAL DEGREES OF FREEDOM. THE ROUTINE BOSSOP
C AND ALL ROUTINES REFERENCED BY BOSSOP ARE COMPATIBLE WITH VERSION
C 4.84 OF VERTEX, WHICH IS A PROGRAM FOR CALCULATING SCHREINEMAKERS
C TYPE PHASE DIAGRAMS.
C
C THE BOUNDS ALGORITHM AND DETAILS OF THE PROGRAM ARE DESCRIBED IN
C A MANUSCRIPT SUBMITTED TO THE JOURNAL CALPHAD IN APRIL, 1985.
C
C THE ALGORITHMS FOR THE ROUTINES FACTOR AND SUBST ARE MODIFIED
```

```
C FROM CONTE AND DE BOOR, ELEMENTARY NUMERICAL ANALYSIS, MCGRAW-
C HILL CO., NEW YORK, 1980.
C
C THE REMAINING ALGORITHMS WERE DEVELOPED AND CODED IN FORTRAN BY
C J. A. D. CONNOLLY AT THE DEPT. OF GEOSCIENCES OF THE PENNSYLVANIA
C STATE UNIVERSITY, UNIVERSITY PARK, PA 16802.
C
C THE PROGRAM REFERENCES I/O DEVICES ASSIGNED LOGICAL UNIT NUMBERS
C N1, N2, AND N3. THE DEVICES AND THEIR ASSOCIATED FILES ARE
C DEFINED IN THE VM/SP SYSTEM SPECIFIC SUBPROGRAM FOPEN. FOR BATCH
C MODE MVS OPERATING SYSTEMS THE REFERENCE TO FOPEN SHOULD BE
C DELETED. LOGICAL UNIT NUMBERS WILL THEN BE DEFINED IN THE BLOCK DATA
C SUBPROGRAM AND PHYSICAL DEVICES MUST BE ATTACHED BY JCL.
C
C INPUT IS READ FROM DEVICE N1.
C PRINTER OUTPUT IS WRITTEN TO DEVICE N2.
C GRAPHICS APPLICATIONS OUTPUT IS WRITTEN TO DEVICE N3.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      CHARACTER*8 FNAME,BLANK,NAMES,IPNAME,IPNMS,IRNMS,EXNAME,SNAME
     *            RECORD*80,CNAME*5,TNAME*72
C
      DIMENSION ISUB(4,5)
C
      COMMON/ CST37 /JKL(4),CNAME(12)/ CST59 /TNAME(80),ITITLE
     *      / CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *      / CST60 /IKP(800),IASMBL(1273),IPOINT,IMYN/ CST12 /CP(800,5)
     *      / CST6 /ICOMP,ISTCT,IPHCT,ICP/ CST2 /G(800)/ CST7 /IFLAG
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *      / CST22 /IDT(10,800),ITCT(10)/ CST14 /IBS,ITS,I4S,I5S
     *      / CST27 /IQAC(5,4),IQACT(5),IQICT,IDQ(5,10),IDQI(5)
     *      / CST24 /IV1,IV2,IV3/ CST8 /NAMES(800)
     *      / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)/ CST18 /ICCT(800)
C
      DATA ISUB/4*0,1,3*0,3,1,2*0,6,4,1,0,2*10,5,1/
C-----------------------------------------------------------------------
C                                 ON VM/SP DEFINE FILES
      CALL FOPEN
C                                 INITIALIZATION
      DO 1 I=1,800
      IVCHK(I)=0
1     ICCT(I)=0
      DO 3 I=1,10
      ITCT(I)=0
3     IBCT(I)=0
      DO 7 I=1,5
      IQACT(I)=0
7     IUCT(I)=0
      IQICT=0
C
      IMYN=1
      ISTCT=1
C                                 READ INPUT
C
C                                 READ # OF TITLE CARDS:
      READ (N1,*) ITITLE
C                                 READ TITLE CARDS:
      READ (N1,10) (TNAME(Z),Z=1,ITITLE)
C                                 READ # OF COMPONENTS:
      READ (N1,*) ICP
C                                 SET COUNTERS FOR THE # OF SUBSYSTEMS:
      IBS=ISUB(1,ICP)
      ITS=ISUB(2,ICP)
      I4S=ISUB(3,ICP)
      I5S=ISUB(4,ICP)
C                                  READ COMPONENT NAMES:
      READ (N1,20) (CNAME(Z),Z=1,ICP)
C                                  READ FLAG FOR GRAPHICS FILE OUTPUT
C                                  TO UNIT N2, YES=0/NO=1:
      READ (N1,*) IGRF
C                                  READ # OF PHASES
      READ (N1,*) IPHCT
C                                  READ PHASE NAMES:
      READ (N1,40) (NAMES(Z),Z=1,IPHCT)
C                                  READ PHASE FLAGS, IKP IS ZERO IF THE
C                                  PHASE IS A TRUE POINT PHASE, AND > 0
C                                  IF THE PHASE IS A PSEUDO-COMPOUND,
C                                  PSEUDO-COMPOUNDS OF SPECIFIC SOLUTION
C                                  IKP IS ASSIGNED A UNIQUE VALUE
      READ (N1,*) (IKP(Z),Z=1,IPHCT)
C                                  READ MOLAR GIBBS ENERGIES:
      READ (N1,*) (G(Z),Z=1,IPHCT)
C                                  READ MOLAR STOICHIOMETRY:
      READ (N1,*) ((CP(Z,ZJ),ZJ=1,ICP),Z=1,IPHCT)
C                                  CALL SORT TO DETERMINE THE SIMPLEST
C                                  SUBSYSTEM THAT EACH PHASE MAY BE
C                                  CLASSIFIED INTO. SORT ALSO CALCULATES
C                                  THE MOLE FRACTIONS OF THE COMPONENTS
C                                  IN EACH PHASE.
      CALL SORT
C                                  CALL BOSSOP TO IDENTIFY THE PHASE
C                                  CONFIGURATIONS OF MINIMUM FREE
C                                  ENERGY. (THE STABLE PHASE ASSEMBLAGES).
50    CALL BOSSOP
C                                  OUTPUT SYSTEM SUMMARY :
C                                  TITLE INFORMATION:
      CALL OUTTIT (N2)
C                                  STABLE PHASE ASSEMBLAGES AND
C                                  GRAPHICS FILES:
      CALL OUTCHM (IGRF)
C
      STOP
C
10    FORMAT (A72)
20    FORMAT (5(A5,1X))
40    FORMAT (8(A8,1X))
C
      END
      SUBROUTINE BOSSOP
C---------------------------------------------------------------------
C BOSSOP CALLS SUBROUTINES TO DETERMINE THE STABLE PHASE
C ASSEMBLAGES OF PROGRESSIVELY MORE COMPLEX SUBSYSTEMS.
C AFTER THE ASSEMBLAGES HAVE BEEN DETERMINED BY THE SIMPLC ROUTINES
C THE ROUTINE MISCIB IS CALLED TO TEST FOR IMMISCIBILITY.
C   REFERENCED BY: MAIN
C   REFERENCES TO: SIMPL1,SIMPL2,SIMPL3,SIMPL4,SIMPL5,MISCIB
C---------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST58 /IDCF(200,5),ICFCT/ CST6 /ICOMP,ISTCT,IPHCT,ICP
     *       / CST47 /IDQF(5,200,4),IQFCT(5),IQFCTH,IGO
     *       / CST15 /IDC(5),IDBV(10,50),IBVCT(10)
     *       / CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *       / CST52 /HCP,H,HD,ID,JD,KD,LD,MD,ITC1,ITC2,ITC3,ITC4,ITC5
C---------------------------------------------------------------------
C                                  ONE COMPONENT SYSTEMS:
      CALL SIMPL1
      IF (ICP.EQ.1) RETURN
```

```
C                                  TWO COMPONENT SYSTEMS:
      CALL SIMPL2
C                                  TEST FOR IMMISCIBILITY
      IF (ICP.GT.2) GOTO 300
      ICT=IBVCT(1)
      DO 210 I=2,ICT
      HD=IDBV(1,I-1)
      ID=IDBV(1,I)
210   CALL MISCIB (I)
      RETURN
C                                  THREE COMPONENT SYSTEMS:
300   CALL SIMPL3 (IFLG4)
C                                  TEST FOR IMMISCIBILITY
      IF (ICP.GT.3) GOTO 400
      ICT=ITPCT(1)
      DO 310 I=1,ICT
      HD=IDTV(1,I,1)
      ID=IDTV(1,I,2)
      JD=IDTV(1,I,3)
310   CALL MISCIB (I)
      RETURN
C                                  FOUR COMPONENT SYSTEMS:
400   CALL SIMPL4 (IFLG4)
C                                  TEST FOR IMMISCIBILITY
      IF (ICP.GT.4) GOTO 500
      ICT=IQFCT(1)
      DO 410 I=1,ICT
      HD=IDQF(1,I,1)
      ID=IDQF(1,I,2)
      JD=IDQF(1,I,3)
      KD=IDQF(1,I,4)
410   CALL MISCIB (I)
      RETURN
C                                  FIVE COMPONENTS:
500   CALL SIMPL5
C                                  TEST FOR IMMISCIBILITY
      DO 510 I=1,ICFCT
      HD=IDCF(I,1)
      ID=IDCF(I,2)
      JD=IDCF(I,3)
      KD=IDCF(I,4)
      LD=IDCF(I,5)
510   CALL MISCIB (I)
C
      RETURN
      END
      SUBROUTINE OUTTIT (N)
C-----------------------------------------------------------------------
C OUTTIT WRITES TITLE INFORMATION AND A BRIEF DESCRIPTION OF THE
C CHEMICAL SYSTEM FOR EACH CALCULATION REQUESTED.
C
C THE SUBROUTINE ARGUMENTS ARE:
C   'N'- SPECIFIES THE LOGICAL UNIT FOR THE OUTPUT.
C ----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      CHARACTER*5 CNAME,NAMES*8,TNAME*72
C
      COMMON/ CST59 /TNAME(80),ITITLE/ CST3 /X(800,5)/ CST8 /NAMES(800)
     *      / CST6 /ICOMP,ISTCT,IPHCT,ICP/ CST14 /IBS,ITS,I4S,I5S
     *      / CST37 /JKL(4),CNAME(12)/ CST40 /IDS(3,9),ISCT(3),ICP1,ISAT
     *      / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
C------------------------------------------------------------------------
C                           OUTPUT TITLE CARDS :
      WRITE (N,1010) (TNAME(Z),Z=1,ITITLE)
```

```
C                          OUTPUT COMPONENTS :
360   WRITE (N,1080) (CNAME(Z),Z=1,ICP)
C                          OUTPUT PHASE ASSEMBLAGES :
      GOTO (390,380,370), ICP
      WRITE (N,1150) (CNAME(Z),Z=1,ICP)
      WRITE (N,1020)
      DO 400 I=ISTCT,IPHCT
      WRITE (N,1160) NAMES(I),(X(I,ZJ),ZJ=1,ICP)
400   CONTINUE
      RETURN
370   WRITE (N,1090) (CNAME(Z),Z=2,3)
      WRITE (N,1100) (NAMES(ZI),X(ZI,2),X(ZI,3),ZI=ISTCT,IPHCT)
      RETURN
380   WRITE (N,1040) CNAME(I)
      WRITE (N,1030) (NAMES(ZI),X(ZI,2),ZI=ISTCT,IPHCT)
      RETURN
390   WRITE (N,1130)
      WRITE (N,1110) (NAMES(Z),Z=ISTCT,IPHCT)
999   RETURN
1010  FORMAT (1X,A72)
1020  FORMAT (/)
1030  FORMAT (4(2X,A8,1X,F5.3))
1040  FORMAT (/,1X,'PHASES AND (PROJECTED) MOL FRACTION ',A5,' :',/)
1080  FORMAT (/,1X,'COMPONENTS WITH UNCONSTRAINED CHEMICAL POTENT'
     *        ,'IALS:',//,1X,12(A5,3X))
1090  FORMAT (/,1X,'PHASES AND (PROJECTED) MOLE FRACTION ',A5,
     *          ' AND ',A5,' :',/)
1100  FORMAT (4(1X,A4,2X,F4.2,2X,F4.2,7X))
1110  FORMAT (7(1X,A8,1X))
1130  FORMAT (/,1X,'PHASES:',/)
1150  FORMAT (/,1X,'PHASES AND (PROJECTED) MOL FRACTION :',
     *         //,15X,5(1X,A5,2X))
1160  FORMAT (3X,A8,4X,5(F5.3,3X))
      END
      SUBROUTINE OUTCHM (IGRF)
C-----------------------------------------------
C----------          OUTCHM          -------------
C
C OUTCHM WRITES NEW CHEMOGRAPHIES TO UNIT M3 AS THEY ARE GENERATED
C BY SIMPL1, SIMPL2, SIMPL3,  OR CHECK.  THE FLAG ICHEM(=1)
C DETERMINES IF OUTCHM IS CALLED, NO OUTPUT WILL BE GENERATED IF THE
C FLAG IO3=1 (SEE INTRODUCTORY COMMENTS).
C-----------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      CHARACTER*8 NAMES
      COMMON/ CST3 /X(800,5)/ CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *      / CST58 /IDCF(200,5),ICFCT/ CST6 /ICOMP,ISTCT,IPHCT,ICP
     *      / CST47 /IDQF(5,200,4),IQFCT(5),IQFCTH,IGO
     *      / CST40 /IDS(3,9),ISCT(3),ICP1,ISAT/ CST8 /NAMES(800)
     *      / CST15 /IDC(5),IDBV(10,50),IBVCT(10)
     *      / CST60 /IKP(800),IASMBL(1273),IPOINT,IMYN
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *      / CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *      / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
C
      GOTO (1000,2000,3000,4000,5000),ICP
C                          OUTPUT UNARY CHEMOGRAPHIES.
1000  WRITE (N2,1060) NAMES(IDC(1))
      GOTO 9000
C                          OUTPUT BINARY CHEMOGRAPHIES.
2000  WRITE (N2,2060)
      IB=IBVCT(1)
      WRITE (N2,2010) (NAMES(IDBV(1,ZJ)),ZJ=1,IB)
      WRITE (N2,*)
      GOTO 9000
```

```
C                                 OUTPUT TERNARY CHEMOGRAPHIES.
3000  WRITE (N2,3060)
      IB=ITPCT(1)
      WRITE (N2,3070) ((NAMES(IDTV(1,ZI,ZJ)),ZJ=1,3),IASMBL(ZI),
     *                ZI=1,IB)
      WRITE (N2,*)
      GOTO 9000
C                                 OUTPUT QUATERNARY CHEMOGRAPHY:
4000  WRITE (N2,4010)
      IB=IQFCT(1)
      DO 400 LF=1,IB
400   WRITE (N2,4020) (NAMES(IDQF(1,LF,ZJF)),ZJF=1,4),IASMBL(LF)
      GOTO 9000
C
5000  WRITE (N2,5010)
      DO 510 I=1,ICFCT
510   WRITE (N2,5020) (NAMES(IDCF(I,ZJ)),ZJ=1,5),IASMBL(I)
C
9000  WRITE (N2,*) ' '
C                                 WRITE MISCIBILITY FLAG:
      IF (IMYN.EQ.1) WRITE (N2,*) '** NO IMMISCIBILITY WAS',
     *                            'DETECTED IN STABLE SOLUTIONS'
      IF (IMYN.EQ.0) WRITE (N2,*) '** IMMISCIBILITY OCCURS IN ONE OR',
     *                            ' MORE STABLE SOLUTIONS'
      GOTO (999),IGRF
C                                 OUTPUT GRAPHICS FILE IF REQUESTED:
C                                 NUMBER OF COMPONENTS, PHASE COUNTERS,
C                                 ASSEMBLAGE COUNTER, FLUID SATURATION FLAG,
C                                 COMPONENT SATURATION FLAG, P, T, AND XCO2.
      WRITE (N3,1010) ICP,ISTCT,IPHCT,IPOINT,IB,IFYN,ISYN,P,T,XCO2
C                                 WRITE PHASE NAMES
      WRITE (N3,1015) (NAMES(ZI),ZI=ISTCT,IPHCT)
C                                 WRITE PHASE COORDINATES
      ICP1=ICP-1
      WRITE (N3,1025) ((X(ZI,ZJ),ZJ=1,ICP1),ZI=ISTCT,IPHCT)
C                                 STABLE CONFIGURATIONS, PHASES ARE
C                                 LABELLED BY THE INDEX 'ZI' IN THE
C                                 LIST OF PHASES.
      GOTO (999,950,960,970,999),ICP
C                                 BINARY
950   WRITE (N3,1020) (IDBV(1,ZJ),ZJ=1,IB)
      GOTO 980
C                                 TERNARY
960   WRITE (N3,1020) ((IDTV(1,ZI,ZJ),ZJ=1,3),ZI=1,IB)
      GOTO 980
C                                 QUATERNARY
970   WRITE (N3,1020) ((IDQF(1,ZI,ZJ),ZJ=1,4),ZI=1,IB)
C                                 WRITE ASSEMBLAGE FLAGS
980   WRITE (N3,1020) (IASMBL(ZJ),ZJ=1,IB)
C
999   RETURN
C
1010  FORMAT (7(I3,1X),F9.2,1X,F7.2,1X,F6.4)
1015  FORMAT (10(A8))
1060  FORMAT (/,1X,A8,' IS THE STABLE UNARY PHASE.',/)
1020  FORMAT (20(I3,1X))
1025  FORMAT (11(F6.4,1X))
2010  FORMAT(6(2X,A8))
2060  FORMAT (/,1X,'THE STABLE BINARY JOIN IS DEFINED BY :',/)
3060  FORMAT (/,1X,'THE STABLE TERNARY ASSEMBLAGES ARE :')
3070  FORMAT (4(1X,2(A4,'-'),A4,'(',I1,')',6X))
4010  FORMAT (/,1X,'THE STABLE QUATERNARY ASSEMBLAGES ARE :',/)
4020  FORMAT (3(1X,3(A4,'-'),A4,'(',I1,')',5X))
5010  FORMAT (/,1X,'THE STABLE QUINARY ASSEMBLAGES ARE :',/)
5020  FORMAT (2(1X,4(A4,'-'),A8,'(',I1,')',5X))
```

```
9010  FORMAT (1X,A5,2X,5(F7.5,1X))
9020  FORMAT (1X,A8,2X,F10.0,2X,5(F7.5,1X))
      END
      BLOCK DATA
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      CHARACTER*8 IPNAME
C
      COMMON/ CST5 /P,T,XCO2,TR,PR,R,PS
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)/ CST28 /ITB(10,3)
     *      / CST27 /IQAC(5,4),IQACT(5),IQICT,IDQ(5,10),IDQI(5)
     *      / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
     *      / CST32 /PTX(450),IPT3/ CST57 /ICB(5,4),ICE(5,6)
     *      / CST50 /IQTB(5,4,3),IQB(5,6)/ CST51 /IQT(5,4)
C-----------------------------------------------------------------------
C                              IQTB(I,J,K) IS AN ARRAY IDENTIFYING
C                              THE K BINARY JOINS NOT INCLUDED IN THE
C                              JTH TERNARY SUBSYSTEM OF THE ITH
C                              QUATERNARY SYSTEM.
      DATA IQTB/ 4, 7, 7, 7, 8, 2, 2, 4, 4, 5,
     *           1, 1, 1, 2, 3, 1, 1, 1, 2, 3,
     *           5, 8, 8, 9, 9, 3, 3, 5, 6, 6,
     *           3, 3, 5, 6, 6, 2, 2, 4, 4, 5,
     *           6, 9,10,10,10, 6, 9,10,10,10,
     *           5, 8, 8, 9, 9, 4, 7, 7, 7, 6/,
C                              IQB(I,J) IS AN ARRAY IDENTIFYING THE
C                              J BINARY JOINS INCLUDED IN THE ITH
C                              QUATERNARY JOIN.
     *     IQB/ 1, 1, 1, 2, 3, 2, 2, 4, 4, 5, 3, 3, 5, 6, 6,
     *          4, 7, 7, 7, 8, 5, 8, 8, 9, 9, 6, 9,10,10,10/,
C                              IQT(I,J) IS AN ARRAY IDENTIFYING THE
C                              THE J TERNARY JOINS IN THE ITH QUAT.
     *     IQT/ 1, 1, 2, 3, 4, 2, 5, 5, 6, 7,
     *          3, 6, 8, 8, 9, 4, 7, 9,10,10/
C                              ICB(I,J) IS AN ARRAY IDENTIFYING THE
C                              J BINARY JOINS NOT INCLUDED IN THE
C                              ITH QUATERNARY JOIN.
      DATA ICB/7,4,2,1,1,8,5,3,3,2,9,6,6,5,4,10,10,9,8,7/,
C                              ICE(I,J) IS AN ARRAY IDENTIFYING THE
C                              J TERNARY JOINS NOT INCLUDED IN THE
C                              ITH QUATERNARY JOIN.
     *     ICE/5,2,1,1,1,6,3,3,2,2,7,4,4,4,3,
     *         8,8,6,5,5,9,9,7,7,6,10,10,10,9,8/
      DATA IBC/1,1,2,1,2,3,1,2,3,4,2,3,3,4,4,4,5,5,5,5/
      DATA ITC/1,1,1,2,1,1,2,1,2,3,2,2,3,3,2,3,3,4,4,4,3,4,4,4,
     *         5,5,5,5,5,5/,IQAC/1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,
     *         4,5,5,5,5/
      DATA ITB/1,1,2,3,1,2,3,4,5,6,2,4,4,5,7,7,8,7,8,9,
     *         3,5,6,6,8,9,9,10,10,10/
C                              THE FOLLOWING DATA STATEMENT ASSIGNS
C                              LOGICAL UNIT NUMBERS FOR I/O.
      DATA N1,N2,N3,N4,N5,N6/5,6,6,22,24,23/
      END
      SUBROUTINE SORT
C-----------------------------------------------------------------------
C SORT DETERMINES IF A PHASE IN THE SYSTEM DEFINED IN INPUT MAY
C BE CLASSIFIED INTO A SUBSYSTEM. I INDEXES THE PHASE IN THE ARRAYS.
C
C   REFERENCED BY: MAIN
C   REFERENCES TO: NONE
C   INPUT ARRAYS:      CP,IBC,ITC,IQAC
C   OUTPUT ARRAYS:     X,ICCT,IUCT,IBCT,ITCT,IQACT,IQICT,IDU,IDB
C                      IDT,IDQ,IDQI
```

```
C-------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST3 /X(800,5)/ CST18 /ICCT(800)/ CST12 /CP(800,5)
     *      / CST6 /ICOMP,ISTCT,IPHCT,ICP
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *      / CST14 /IBS,ITS,I4S,I5S/ CST22 /IDT(10,800),ITCT(10)
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *      / CST27 /IQAC(5,4),IQACT(5),IQICT,IDQ(5,10),IDQI(5)
     *      / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
C-------------------------------------------------------------------------
      DO 160 I=1,IPHCT
      TOTMOL=0.D0
C
      DO 10 J=1,ICP
      IF (CP(I,J).EQ.0.D0) GOTO 10
      ICCT(I)=ICCT(I)+1
      TOTMOL=TOTMOL+CP(I,J)
10    CONTINUE
C                              COMPUTE MOLE FRACTIONS:
       DO 15 J=1,ICP
15     X(I,J)=CP(I,J)/TOTMOL
C                              IF ICCT<ICP PHASE IS DEGENERATE
      IGO=ICCT(I)
      IF (ICP.EQ.ICCT(I)) GOTO 5
      GOTO (70,80,90,100), IGO
5     GOTO (20,30,40,50,60), IGO
C                              UNARY
20    IUCT(1)=IUCT(1)+1
      IDU(1,IUCT(1))=I
      GOTO 160
C                              BINARY
30    IBCT(1)=IBCT(1)+1
      IDB(1,IBCT(1))=I
      GOTO 160
C                              TERNARY
40    ITCT(1)=ITCT(1)+1
      IDT(1,ITCT(1))=I
      GOTO 160
C                              QUATERNARY
50    IQACT(5)=IQACT(5)+1
      IDQ(1,IQACT(5))=I
      GOTO 160
C                              QUINARY
60    IQICT=IQICT+1
      IDQI(IQICT)=I
      GOTO 160
C
C                              DEGENERATE PHASES:
C
C                              UNARY SUBSYSTEM
70    DO 110 J=1,ICP
      IF (CP(I,J).EQ.0.D0) GOTO 110
      IUCT(J)=IUCT(J)+1
      IDU(J,IUCT(J))=I
      GOTO 160
110   CONTINUE
C                              BINARY SUBSYSTEM
80    DO 120 J=1,IBS
      IF ((CP(I,IBC(J,1)).EQ.0.D0).OR.
     *    (CP(I,IBC(J,2)).EQ.0.D0)) GOTO 120
      IBCT(J)=IBCT(J)+1
      IDB(J,IBCT(J))=I
      GOTO 160
120   CONTINUE
```

```
C                                TERNARY SUBSYSTEM
90    DO 130 J=1,ITS
      IF ((CP(I,ITC(J,1)).EQ.0.D0).OR.
     *   (CP(I,ITC(J,2)).EQ.0.D0).OR.
     *   (CP(I,ITC(J,3)).EQ.0.D0)) GOTO 130
      ITCT(J)=ITCT(J)+1
      IDT(J,ITCT(J))=I
      GOTO 160
130   CONTINUE
C                                QUATERNARY SUBSYSTEM
100   DO 140 J=1,I4S
      IF ((CP(I,IQAC(J,1)).EQ.0.D0).OR.
     *   (CP(I,IQAC(J,2)).EQ.0.D0).OR.
     *   (CP(I,IQAC(J,3)).EQ.0.D0).OR.
     *   (CP(I,IQAC(J,4)).EQ.0.D0)) GOTO 140
      IQACT(J)=IQACT(J)+1
      IDQ(J,IQACT(J))=I
      GOTO 160
140   CONTINUE
      WRITE (N3,*) '**ERROR VER014** SORT'
      STOP
160   CONTINUE
      RETURN
      END
      SUBROUTINE SIMPL1
C-----------------------------------------------------------------------
C UNARY DETERMINES THE STABLE PHASE IN UNARY (SUB)SYSTEMS.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST48 /U(5)/ CST2 /G(800)/ CST12 /CP(800,5)
     *      / CST15 /IDC(5),IDBV(10,50),IBVCT(10)
     *      / CST6 /ICOMP,ISTCT,IPHCT,ICP/ CST14 /IBS,ITS,I4S,I5S
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
C-----------------------------------------------------------------------
      DO 1 I=1,5
      U(I)=1.D67
1     CONTINUE
C                                IDENTIFY STABLE PHASE, SAVE ID
C                                IN IDC, AND CHEMICAL POTENTIAL IN U.
      DO 10 I=1,ICP
      IU=IUCT(I)
      DO 20 J=1,IU
      UPHI=G(IDU(I,J))/CP(IDU(I,J),I)
      IF (UPHI.GE.U(I)) GOTO 20
      U(I)=UPHI
      IDC(I)=IDU(I,J)
20    CONTINUE
10    CONTINUE
9999  RETURN
      END
      SUBROUTINE SIMPL2
C-----------------------------------------------------------------------
C SIMPL2 DETERMINES THE VERTICES OF THE MINIMUM FREE ENERGY
C SURFACE FOR BINARY (SUB)SYSTEMS. THE VERTICES ARE RETURNED IN THE
C ARRAY IDBV.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      DIMENSION IIDB(60)
C
      COMMON/ CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *      / CST2 /G(800)/ CST12 /CP(800,5)/ CST3 /X(800,5)
     *      / CST15 /IDC(5),IDBV(10,50),IBVCT(10)/ CST48 /U(5)
     *      / CST6 /ICOMP,ISTCT,IPHCT,ICP/ CST14 /IBS,ITS,I4S,I5S
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *      / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,I03,I04,I05
     *      / CST52 /HCP,H,HD,ID,JD,KD,LD,MD,IBC1,IBC2,IBC3,IBC4,IBC5
C-----------------------------------------------------------------------
      IFLG2=0
      HCP=2
C                            TEST FOR PHASES BELOW THE COMPOSANT G-X
C                            PLANE. SAVE METASTABLE PHASE ID'S IN IIDB
      DO 30 I=1,IBS
      IBC1=IBC(I,1)
      IBC2=IBC(I,2)
      ICT=1
      IF (IBCT(I).EQ.0) GOTO 90
C                            TRUE IF ONLY PHASES ARE COMPOSANTS
      IB=IBCT(I)
      DO 40 J=1,IB
      IF (G(IDB(I,J))-CP(IDB(I,J),IBC1)*U(IBC1)
     *              -CP(IDB(I,J),IBC2)*U(IBC2)
     *                                           .GT.0.D0) GOTO 40
      ICT=ICT+1
      IIDB(ICT)=IDB(I,J)
40    CONTINUE
C                       NOW FIND THE STABLE VERTICES FOR THE SAME BINARY
90    IDBV(I,1)=IDC(IBC1)
C                       ICT=1, COMPOSANTS ARE THE ONLY STABLE PHASES.
      IF (ICT.NE.1) GOTO 10
      IDBV(I,2)=IDC(IBC2)
      IBV=2
      GOTO 30
C                            SOME BINARY PHASES ARE STABLE:
10    ICT=ICT+1
      IIDB(1)=IDC(IBC1)
      IIDB(ICT)=IDC(IBC2)
C                            DETERMINE STABLE BINARY PHASES:
      IBV=1
      HD=IDC(IBC1)
60    ID=IDC(IBC2)
      CALL ABLOAD (*9000)
C                            BEGIN TESTING PHASES:
      DO 50 J=2,ICT
      JD=IIDB(J)
      IF (X(HD,IBC2).GE.X(JD,IBC2)) GOTO 50
      IF (G(JD)-CP(JD,IBC1)*B(1)-CP(JD,IBC2)*B(2).GT.1.D-05) GOTO 50
      ID=JD
      CALL ABLOAD (*9000)
50    CONTINUE
C                            AT END OF JOIN?
      IBV=IBV+1
      IDBV(I,IBV)=ID
      IF (X(ID,IBC2).EQ.1.D0) GOTO 30
      HD=ID
      GOTO 60
C                            DONE
30    IBVCT(I)=IBV
      RETURN
9000  WRITE (N3,*) '**ERROR BOU019** SINGULAR MATRIX IN SIMPL2'
      STOP
      END
      SUBROUTINE SIMPL3 (IFLG4)
C-----------------------------------------------------------------------
C SIMPL3 IS A SUBPROGRAM WHICH COMPUTES THE STABLE PHASE CONFIGU
C ATIONS OF TERNARY (SUB)SYSTEMS.  THE ALGORITHM IS DESCRIBED IN
C DETAIL IN PROGRAM DOCUMENTATION ELSEWHERE.  SEVERAL VERSIONS OF
C SIMPL3 ARE AVAILABLE, THIS VERSION IS DESIGNED FOR EFFICIENT
C COMPUTATION IN SYSTEMS WITH BINARY SOLUTIONS.
```

```
C
C   REFERENCED BY: BOSSOP
C   REFERENCES TO: ABLOAD,ASSD3,ASPSB,BOUND3,DGPH3,ITEST3,SID3
C                  SLOPES
C   INPUT ARRAYS:      G,CP,U,IDC,IDBV,IBVCT,ITC,ITB
C   OUTPUT ARRAYS:     IDTV,ITPCT,ITTCT,IDTT
C   TEMPORARY ARRAYS: ITBH,IVCHK,IDTPST,IDPSF,IBIN
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      DIMENSION IBIN(100)
C
      COMMON/ CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *       / CST48 /U(5)/ CST46 /IPSF,IDPSF(2000,5)
     *       / CST2 /G(800)/ CST3 /X(800,5)/ CST12 /CP(800,5)
     *       / CST15 /IDC(5),IDBV(10,50),IBVCT(10)
     *       / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *       / CST22 /IDT(10,800),ITCT(10)/ CST6 /ICOMP,ISTCT,IPHCT,ICP
     *       / CST14 /IBS,ITS,I4S,I5S/ CST18 /ICCT(800)
     *       / CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *       / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)/ CST28 /ITB(10,3)
     *       / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
     *       / CST45 /ITBH(3),ITT,ITTV,ITTP
     *       / CST52 /HCP,H,HD,ID,JD,KD,LD,MD,ITC1,ITC2,ITC3,ITC4,ITC5
C-----------------------------------------------------------------------
      DO 1 I=3,IPHCT
1     IVCHK(I)=0
C
      HCP=3
C
      DO 500 H=1,ITS
C                                   FOR EACH TERNARY SUBSYSTEM
C                                   INITIALIZE SIMPLE VARIABLES
      ITC1=ITC(H,1)
      ITC2=ITC(H,2)
      ITC3=ITC(H,3)
      ITTP=0
      ITT=ITCT(H)
      ITTV=0
      IPSF=0
      ITPSF=1
      IFLG4=0
      HGO=0
C                                   DETERMINE NUMBER OF BINARY PHASES
C                                   IN THE APPROPRIATE TERNARY:
      IFLG2=0
      IBT=0
      DO 2 I=1,3
      ITBH(I)=ITB(H,I)
      ITBD=ITBH(I)
      IB=IBVCT(ITBD)-1
      GOTO (3),IB
      IFLG2=IFLG2+1
      DO 4 K=2,IB
      IBT=IBT+1
4     IBIN(IBT)=IDBV(ITBD,K)
3     IBT=IBT+1
2     IBIN(IBT)=IDC(ITC(H,I))
C                                   IFLG2 = NUMBER OF BINARIES WITH
C                                   BINARY VERTICES
      IF (IFLG2.NE.0) GOTO 250
      IF (ITT.EQ.0) GOTO 270
      DO 280 I=1,ITT
280   IF (G(IDT(H,I))-CP(IDT(H,I),ITC1)*U(ITC1)
     *               -CP(IDT(H,I),ITC2)*U(ITC2)
     *              -CP(IDT(H,I),ITC3)*U(ITC3)
     *      .LT.0.D0) GOTO 250
270   HD=IDC(ITC1)
      ID=IDC(ITC2)
      JD=IDC(ITC3)
      CALL ASSD3 (ITPSF ,HGO)
      GOTO 271
C                             IDENTIFY TERNARY FACETS WITH 2
C                             VERTICES ON A BINARY:
250   DO 10 I=1,3
      IDBT=ITBH(I)
      II=4-I
      IB=IBVCT(IDBT)-1
      DO 20 J=1,IB
C                             SET THE BINARY VERTICES:
      HD=IDBV(IDBT,J)
      ID=IDBV(IDBT,J+1)
C                             1ST TEST THE OPPOSITE COMPOSANT:
      JD=IDC(ITC(H,II))
C                             TEST IF A VERTEX IS CONTAINED:
      IF (ITTV.EQ.0) GOTO 25
      CALL BOUND3
      DO 35 K=1,ITTV
      IF (ITEST3(IDTT(H,K)).NE.0) GOTO 35
      JD=IDTT(H,K)
      CALL BOUND3
35    CONTINUE
C                             DETERMINE U'S:
25    CALL ABLOAD (*9000)
C                             BEGIN TESTING AGAINST BINARIES:
      DO 30 K=1,3
      IF (K.EQ.I) GOTO 30
      ITBD=ITBH(K)
      JB=IBVCT(ITBD)-1
      GOTO (30),JB
      DO 40 L=2,JB
      KD=IDBV(ITBD,L)
      IF (DGPH3(KD).GT.0.0D0) GOTO 40
C                             JD METASTABLE CHANGE WITH KD:
      JD=KD
      CALL ABLOAD (*9000)
40    CONTINUE
30    CONTINUE
C                             BEGIN TESTING TERNARY PHASES:
      IF (ITT.EQ.0) GOTO 50
      DO 60 K=1,ITT
      KD=IDT(H,K)
      IF (DGPH3(KD).GT.0.0D0) GOTO 60
C                             JD METASTABLE CHANGE WITH KD:
      JD=KD
      CALL ABLOAD (*9000)
60    CONTINUE
C                             ASSIGN NEW FACET:
50    CALL ASSD3 (ITPSF ,HGO)
20    CONTINUE
10    CONTINUE
C                             BEGIN SECOND SEARCH SEGMENT.
      HGO=1
100   IF (ITPSF.EQ.IPSF) GOTO 271
      ISTART=ITPSF
      IEND=IPSF
      ITPSF=IPSF
C                             GENERATE NEW FACETS FROM UNMATCHED
C                             TIELINES:
      DO 90 I=ISTART,IEND
      IF (IDPSF(I,3).EQ.0) GOTO 90
C                                 SET VERTICES:
      HD=IDPSF(I,1)
      ID=IDPSF(I,2)
      CALL SLOP3 (I)
C                                 SET 1ST TEST VERTEX:
      DO 110 K=1,3
C                                 TRY UNARY VERTICES:
      JD=IDC(ITC(H,K))
110   IF (SID3(JD).LT.-1.D-08) GOTO 150
      CALL OUTCHM (IGRF)
      GOTO 9000
C                                 TEST FOR BOUND VERTICES:
150   IF (ITTV.EQ.0) GOTO 135
      CALL BOUND3
      DO 130 K=1,ITTV
      KD=IDTT(H,K)
      IF ((ID.EQ.KD).OR.(ITEST3(KD).NE.0)) GOTO 130
      JD=IDTT(H,K)
      CALL BOUND3
130   CONTINUE
135   CALL ABLOAD (*9000)
C                                 TEST AGAINST BINARIES:
      DO 160 K=1,IBT
      KD=IBIN(K)
      IF ((SID3(KD).GT.-1.D-08).OR.
     *   (DGPH3(KD).GT.0.0D0)) GOTO 160
      JD=KD
      CALL ABLOAD (*9000)
160   CONTINUE
C                                 TEST TERNARY PHASES:
      IF (ITT.EQ.0) GOTO 190
      DO 180 K=1,ITT
      KD=IDT(H,K)
      IF ((DGPH3(KD).GT.-1.D-08).OR.
     *    (SID3(KD).GT.-1.D-08)) GOTO 180
      JD=KD
      CALL ABLOAD (*9000)
180   CONTINUE
C                                 ASSIGN NEW FACETS AND TIELINES:
190   CALL ASSD3 (I      ,HGO)
90    CONTINUE
      GOTO 100
C                                 END OF SEARCH SEGMENT.
271   ITPCT(H)=ITTP
      ITTCT(H)=ITTV
      IF (ITTV.GT.0) IFLG4=IFLG4+1
500   CONTINUE
      H=1
C                                 OUTPUT CHEMOGRAPHY IF REQUESTED.
999   RETURN
9000  WRITE (N3,*) '**ERROR BOU002** SINGULAR MATRIX IN SIMPL3'
      STOP
      END
      SUBROUTINE SIMPL4 (IFLG4)
C-----------------------------------------------------------------------
C SIMPL4 LOCATES THE QUATERNARY FACETS OF A FREE ENERGY SURFACE.
C
C   REFERENCED BY: BOSSOP
C   REFERENCES TO: AB4,ASPST,ASSD4,DGPH4
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *      / CST2 /G(800)/ CST12 /CP(800,5)/ CST46 /IPSF,IDPSF(2000,5)
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *      / CST15 /IDC(5),IDBV(10,50),IBVCT(10)/CST14/IBS,ITS,I4S,I5S
     *      / CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *      / CST52 /HCP,H,HD,ID,JD,KD,LD,MD,IQC1,IQC2,IQC3,IQC4,IQC5
     *      / CST47 /IDQF(5,200,4),IQFCT(5),IQFCTH,IGO
     *      / CST56 /IDTQ(5,99),ITQCT(5),IQVH/ CST18 /ICCT(800)
     *      / CST27 /IQAC(5,4),IQACT(5),IQICT,IDQ(5,10),IDQI(5)
     *      / CST48 /U(5)/ CST51 /IQT(5,4)/ CST50 /IQTB(5,4,3),IQB(5,6)
C-----------------------------------------------------------------------
      HCP=4
C
      DO 500 H=1,I4S
C                               INITIALIZATION:
C
C                               VERTEX COUNTERS
      ITQCT(H)=0
C                               PUT COMPONENT ID'S INTO SIMPLE
C                               VARIABLES:
      IQC1=IQAC(H,1)
      IQC2=IQAC(H,2)
      IQC3=IQAC(H,3)
      IQC4=IQAC(H,4)
C                               IDENTITIES OF THE FOUR TERNARIES IN
C                               THE HTH QUATERNARY:
      DO 5 I=1,4
5     IQTH(I)=IQT(H,I)
C                               COUNTER FOR THE NUMBER OF STABLE
C                               QUATERNARY PHASES:
      IQQS=0
C                               COUNTER OF TOTAL NUMBER OF QUATERNARY
C                               PHASES:
      IQQ=IQACT(H)
C                               COUNTER OF QUATERNARY FACETS:
      IQFCTH=0
C                               COUNTER OF QUATERNARY VERTICES:
      IQVH=0
      JQ=0
C                               COUNTER OF PSEUDO-TERNARY PLANES:
      IPSF=0
      ITPSF=1
C                               FLAG AND COUNTER FOR ASPST IN THE 1ST
C                               SEARCH SEGMENT.
      HGO=0
      IONE=1
C
      IF (IQQ.EQ.0) GOTO 30
C                               ELIMINATE QUATERNARY PHASES
C                               METASTABLE WITH RESPECT TO COMPOSANTS:
      DO 10 I=1,IQQ
      IF (G(IDQ(H,I))-CP(IDQ(H,I),IQC1)*U(IQC1)
     *               -CP(IDQ(H,I),IQC2)*U(IQC2)
     *               -CP(IDQ(H,I),IQC3)*U(IQC3)
     *               -CP(IDQ(H,I),IQC4)*U(IQC4).GT.0.0D0) GOTO 20
      IQQS=IQQS+1
      GOTO 10
20    G(IDQ(H,I))=0.0D0
10    CONTINUE
      IF (IQQS.NE.0) GOTO 30
C                               NO QUATERNARY PHASES STABLE, IF NO
C                               TERNARY PHASES ARE STABLE (IFLG4=0)
C                               THEN THE ONLY FACET IS DEFINED BY
C                               THE COMPOSANTS.
      IF (IFLG4.NE.0) GOTO 30
      IQFCTH=1
      DO 25 I=1,4
```

```
25      IDQF(H,1,I)=IDC(IQAC(H,I))
        GOTO 490
30      DO 100 I=1,4
C                                 IDENTIFY FACETS WITH 3 VERTICES IN A
C                                 TERNARY SUBSYSTEM:
        HI=IQT(H,I)
        JJ=5-I
        II=ITPCT(HI)
        DO 40 J=1,II
C                                 SET TERNARY VERTICES:
        HD=IDTV(HI,J,1)
        ID=IDTV(HI,J,2)
        JD=IDTV(HI,J,3)
C                                 SET 4TH VERTEX AS THE OPPOSITE CO
C                                 COMPOSANT:
        KD=IDC(IQAC(H,JJ))
C                                 DETERMINE U'S:
        CALL ABLOAD (*9000)
C                                 BEGIN TRYING BINARY VERTICES
        DO 41 K=1,3
C                                 IDENTIFY THE BINARY:
        IDB=IQTB(H,I,K)
        IB=IBVCT(IDB)-1
        IF (IB.LT.2) GOTO 41
C                                 START TESTING:
        DO 42 L=2,IB
        LD=IDBV(IDB,L)
        IF (DGPH4(LD).GT.-1.D-08) GOTO 42
C                                 KD IS METASTABLE, EXCHANGE WITH LD:
        KD=LD
        CALL ABLOAD (*9000)
42      CONTINUE
41      CONTINUE
C                                 BEGIN TRYING TERNARY VERTICES:
        DO 50 K=1,3
        ICTT=IQT(H,ITC(JJ,K))
        LL=ITTCT(ICTT)
        IF (LL.EQ.0) GOTO 50
C
        DO 60 L=1,LL
        LD=IDTT(ICTT,L)
        IF (DGPH4(LD).GT.-1.D-08) GOTO 60
C                                 IF DGPH4 IS LT 0 THEN THE FACET HD-
C                                 ID-JD-KD IS METASTABLE WITH RESPECT
C                                 TO LD, INTERCHANGE KD AND LD AND
C                                 CONTINUE:
        KD=LD
        CALL ABLOAD (*9000)
60      CONTINUE
50      CONTINUE
C                                 IF NO QUATERNARY PHASES THEN THE
C                                 ESTABLISHED FACET IS STABLE, ELSE TEST
C                                 QUATERNARY PHASES:
        IF (IQQS.EQ.0) GOTO 90
        DO 70 K=1,IQQ
        LD=IDQ(H,K)
        IF (DGPH4(LD).GT.-1.D-08) GOTO 70
C                                 FACET IS METASTABLE WITH RESPECT TO LD:
C                                 TEST HERE IF KD IS BOUND !!!!!
        KD=LD
        CALL ABLOAD (*9000)
C                                 TEST HERE IF LD IS BOUND !!!!!
70      CONTINUE
C                                 SAVE THE NEW FACET:
90      CALL ASSD4 (IONE,HGO)
```

```
C                                SAVE THE NEW PSEUDO-TERNARY PLANES:
40     CONTINUE
100    CONTINUE
C                                BEGIN 2ND SEARCH SEGMENT.
       HGO=1
       IF (IPSF.EQ.0) GOTO 490
C
160    IF (IPSF.EQ.ITPSF) GOTO 490
       ISTART=ITPSF
       IEND=IPSF
       ITPSF=IEND
C                                IDENTIFY NEW FACETS GENERATED FROM
C                                THE PSTP'S:
       DO 150 I=ISTART,IEND
       IF (IDPSF(I,1).EQ.0) GOTO 150
C                                SET VERTICES:
       HD=IDPSF(I,1)
       ID=IDPSF(I,2)
       JD=IDPSF(I,3)
C                                DETERMINE THE EQUATION OF THE PSEUDO-
C                                TERNARY PLANE HD-ID-JD:
       CALL SLOP4 (I)
C                                SET FIRST TEST VERTEX:
       JJ=1
190    KD=IDC(IQAC(H,JJ))
C                                TEST IF KD IS GEOMETRICALLY FEASIBLE:
       IF (SID4(KD).GT.-1.D-08) GOTO 170
       CALL ABLOAD (*9000)
       GOTO 180
170    JJ=JJ+1
       GOTO 190
C                                CALL BOUNDC HERE TO ESTABLISH
C                                COMPOSITIONAL LIMITS OF THE TRIAL
C                                CONFIGURATION. THEN USE ITESTC TO
C                                TEST FOR BOUNDED STABLE PHASES. THIS
C                                TEST LOOP IS ONLY USEFUL IF THE
C                                SYSTEM CONTAINS QUATERNARY COMPOUNDS.
C                                BEGIN TESTING:
C                                TRY BINARY VERTICES:
180    DO 81 K=1,6
       IDB=IQB(H,K)
       IB=IBVCT(IDB)
       DO 82 L=2,IB
       LD=IDBV(IDB,L)
       IF ((SID4(LD).GT.-1.D-08).OR.(DGPH4(LD).GT.-1.D-08)) GOTO 82
C                                KD IS METASTABLE, EXCHANGE WITH LD:
       KD=LD
       CALL ABLOAD (*9000)
82     CONTINUE
81     CONTINUE
C                                TRY TERNARY VERTICES:
       DO 200 J=1,4
       ICTT=IQT(H,J)
       LL=ITTCT(ICTT)
       IF (LL.EQ.0) GOTO 200
       DO 210 K=1,LL
       LD=IDTT(ICTT,K)
       IF ((SID4(LD).GT.-1.D-08).OR.(DGPH4(LD).GT.-1.D-08)) GOTO 210
C                                FACET IS METASTABLE WITH RESPECT TO
C                                LD:
       KD=LD
       CALL ABLOAD (*9000)
210    CONTINUE
200    CONTINUE
C                                NOW TEST AGAINST QUATERNARY PHASES:
      IF (IQQS.EQ.0) GOTO 230
      DO 220 J=1,IQQ
      LD=IDQ(H,J)
      IF ((SID4(LD).GT.-1.D-08).OR.(DGPH4(LD).GT.-1.D-08)) GOTO 220
      KD=LD
      CALL ABLOAD (*9000)
220   CONTINUE
C                                 ASSIGN NEW FACET:
230   CALL ASSD4 (I,HGO)
C                                 ASSIGN PSTP TO A TEMPORARY STORAGE
C                                 ARRAY.
150   CONTINUE
C
C                                 SAVE PERMANENT COUNTERS:
C
      GOTO 160
490   ITQCT(H)=IQVH
      IQFCT(H)=IQFCTH
      GOTO 500
9000  WRITE (N3,*) '**ERROR BOU004** SINGULAR MATRIX IN SIMPL4'
      STOP
500   CONTINUE
99    RETURN
      END
      SUBROUTINE SIMPL5
C-----------------------------------------------------------------------
C SIMPL5 LOCATES THE QUINARY FACETS OF A FREE ENERGY SURFACE.
C
C   REFERENCED BY: BOSSOP
C   REFERENCES TO: ABLOAD,ASPSQ,ASSD5,DGPH5,SID5,SLOP5
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *       / CST2 /G(800)/ CST12 /CP(800,5)/ CST46 /IPSF,IDPSF(2000,5)
     *       / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *       / CST15 /IDC(5),IDBV(10,50),IBVCT(10)
     *       / CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *       / CST52 /HCP,H,HD,ID,JD,KD,LD,MD,IC(5)
     *       / CST47 /IDQF(5,200,4),IQFCT(5),IQFCTH,IGO
     *       / CST56 /IDTQ(5,99),ITQCT(5),IQVH/ CST58 /IDCF(200,5),ICFCT
     *       / CST27 /IQAC(5,4),IQACT(5),IQICT,IDQ(5,10),IDQI(5)
     *       / CST48 /U(5)/ CST57 /ICB(5,4),ICE(5,6)
C-----------------------------------------------------------------------
C                                 INITIALIZATION:
C
C                                 NUMBER OF COMPONENTS:
      HCP=5
C                                 COMPONENTS:
      DO 1 I=1,5
1     IC(I)=I
C                                 PSEUDO-QUATERNARY PLANE COUNTERS:
      IPSF=0
      ITPSF=1
C                                 FACET COUNTER
      ICFCT=0
C                                 FLAGS
      HGO=0
C                                 BEGIN FIRST SEARCH SEGMENT.  THIS
C                                 SEGMENT LOCATES FACETS WHICH HAVE
C                                 4 VERTICES IN A COMMON QUATERNARY
C                                 SUBSYSTEM.
      DO 10 H=1,5
      JJ=6-H
      HH=IQFCT(H)
      DO 20 I=1,HH
C                                   SET QUATERNARY VERTICES:
      HD=IDQF(H,I,1)
      ID=IDQF(H,I,2)
      JD=IDQF(H,I,3)
      KD=IDQF(H,I,4)
C                                   SET FIFTH VERTEX AS THE OPPOSITE
C                                   COMPOSANT:
      LD=IDC(JJ)
C                                   DETERMINE CHEMICAL POTENTIALS:
      CALL ABLOAD (*9000)
C                                  TEST STABILITY RELATIVE TO BINARY
C                                   VERTICES:
      DO 30 J=1,4
      IDB=ICB(H,J)
      IB=IBVCT(IDB)-1
      GOTO (30),IB
C                                   START TEST LOOP:
      DO 40 K=2,IB
      IF (DGPH5(IDBV(IDB,K)).GT.-1.D-08) GOTO 40
      LD=IDBV(IDB,K)
      CALL ABLOAD (*9000)
40    CONTINUE
C                                   END TEST LOOP.
30    CONTINUE
C                                   TEST STABILITY RELATIVE TO TERNARY
C                                   VERTICES:
      DO 50 J=1,6
      IDB=ICE(H,J)
      IB=ITTCT(IDB)
      IF (IB.EQ.0) GOTO 50
C                                   START TEST LOOP:
      DO 60 K=1,IB
      IF (DGPH5(IDTT(IDB,K)).GT.-1.D-08) GOTO 60
      LD=IDTT(IDB,K)
      CALL ABLOAD (*9000)
60    CONTINUE
C                                   END TEST LOOP.
50    CONTINUE
C                                   TEST STABILITY RELATIVE TO QUATERNARY
C                                   VERTICES:
      DO 70 J=1,5
      IF (J.EQ.H) GOTO 70
      IB=ITQCT(J)
      IF (IB.EQ.0) GOTO 70
C                                   START TEST LOOP:
      DO 80 K=1,IB
      IF (DGPH5(IDTQ(J,K)).GT.-1.D-08) GOTO 80
      LD=IDTQ(J,K)
      CALL ABLOAD (*9000)
80    CONTINUE
C                                   END TEST LOOP.
C                                   TEST STABILITY RELATIVE TO QUINARY
C                                   PHASES:
70    CONTINUE
      IF (IQICT.EQ.0) GOTO 100
C                                   START TEST LOOP:
      DO 90 J=1,IQICT
      IF (DGPH5(IDQI(J)).GT.-1.D-08) GOTO 90
      LD=IDQI(J)
      CALL ABLOAD (*9000)
90    CONTINUE
C                                   END TEST LOOP.
100   CALL ASSD5 (ITPSF,HGO)
20    CONTINUE
```

```
10    CONTINUE
C                                END FIRST SEGMENT
C
C                                BEGIN SECOND SEARCH SEGMENT:
      HGO=1
110   IF (IPSF.EQ.ITPSF) GOTO 250
      ISTART=ITPSF
      IEND=IPSF
      ITPSF=IPSF
C                                GENERATE NEW FACETS FROM THE PSQ'S:
      DO 120 I=ISTART,IEND
      IF (IDPSF(I,1).EQ.0) GOTO 110
C                                SET VERTICES:
      HD=IDPSF(I,1)
      ID=IDPSF(I,2)
      JD=IDPSF(I,3)
      KD=IDPSF(I,4)
C                                DETERMINE THE PLANE IN SIMPLEX
C                                COMPOSITION SPACE INCLUDING HD-
C                                ID-JD-KD.
      CALL SLOP5(I)
C
C                                SET FIRST TEST VERTEX:
      JJ=1
130   KD=IDC(JJ)
      JJ=JJ+1
      IF (SID5(KD).LT.-1.D-08) GOTO 140
      GOTO 130
C                                CALL BOUNDC HERE TO ESTABLISH
C                                COMPOSITIONAL LIMITS OF THE TRIAL
C                                CONFIGURATION. THEN USE ITESTC TO
C                                TEST FOR BOUNDED STABLE PHASES. THIS
C                                TEST LOOP IS ONLY USEFUL IF THE
C                                SYSTEM CONTAINS QUINARY COMPOUNDS.
140   CALL ABLOAD (*9000)
C                                TEST AGAINST COMPOSANTS:
      DO 150 J=JJ,5
      LD=IDC(J)
      IF ((DGPH5(LD).GT.-1.D-08).OR.(SID5(LD).GT.-1.D-08)) GOTO 150
      KD=LD
      CALL ABLOAD (*9000)
150   CONTINUE
C                                TEST AGAINST BINARIES:
      DO 160 J=1,10
      IB=IBVCT(J)-1
      GOTO (160),IB
      DO 170 K=2,IB
      LD=IDBV(J,K)
      IF ((SID5(LD).GT.-1.D-08).OR.(DGPH5(LD).GT.-1.D-08)) GOTO 170
      KD=LD
      CALL ABLOAD (*9000)
170   CONTINUE
160   CONTINUE
C                                TEST AGAINST TERNARIES:
      DO 180 J=1,10
      IB=ITTCT(J)
      IF (IB.EQ.0) GOTO 180
      DO 190 K=1,IB
      LD=IDTT(J,K)
      IF ((SID5(LD).GT.-1.D-08).OR.(DGPH5(LD).GT.-1.D-08)) GOTO 190
      KD=LD
      CALL ABLOAD (*9000)
190   CONTINUE
180   CONTINUE
C                                TEST AGAINST QUATERNARIES:
      DO 200 J=1,5
      IB=ITQCT(J)
      IF (IB.EQ.0) GOTO 200
      DO 210 K=1,IB
      LD=IDTQ(J,K)
      IF ((SID5(LD).GT.-1.D-08).OR.(DGPH5(LD).GT.-1.D-08)) GOTO 210
      KD=LD
      CALL ABLOAD (*9000)
210   CONTINUE
200   CONTINUE
C                             TEST AGAINST QUINARIES:
      IF (IQICT.EQ.0) GOTO 120
      DO 230 J=1,IQICT
      LD=IDQI(J)
      IF ((SID5(LD).GT.-1.D-08).OR.(DGPH5(LD).GT.-1.D-08)) GOTO 230
      KD=LD
      CALL ABLOAD (*9000)
230   CONTINUE
120   CALL ASSD5 (I,HGO)
C
C                             END SECOND SEARCH SEGMENT.
250   RETURN
9000  WRITE (6,1000)
1000  FORMAT (' **ERROR BOU001** SINGULARITY')
      STOP
      END
      SUBROUTINE ASSD3 (ISTART,HGO)
C-----------------------------------------------------------------------
C ASSD3 COUNTS (ITTP) AND ASSIGNS (IDTV) TERNARY FACETS. THE
C VERTICES ARE IDENTIFIED BY THE ARRAY ID(3).
C
C   REFERENCED BY: SIMPL4
C   REFERENCES TO: NONE
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      DIMENSION ILK(2)
C
      COMMON/ CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *      / CST45 /ITBH(3),ITT,ITTV,ITTP
     *      / CST13 /IDU(5,10),IDB(10,60),IBC(10,2),IUCT(5),IBCT(10)
     *      / CST52 /HCP,H,ID(6),ITC1,ITC2,ITC3,ITC4,ITC5
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *      / CST18 /ICCT(800)/ CST46 /IPSF,IDPSF(2000,5)
     *      / CST15 /IDC(5),IDBV(10,50),IBVCT(10)
C
      DATA ILK/2,1/
C-----------------------------------------------------------------------
C
      IF (ITTP.EQ.0) GOTO 40
C                             TEST FOR EQUIVALENCE WITH PREVIOUSLY
C                             IDENTIFIED FACETS:
      I1=ID(1)
      I2=ID(2)
      I3=ID(3)
      DO 10 I=1,ITTP
      IF (
     * ((IDTV(H,I,1).NE.I1).AND.(IDTV(H,I,2).NE.I1).AND.
     *  (IDTV(H,I,3).NE.I1)).OR.
     * ((IDTV(H,I,1).NE.I2).AND.(IDTV(H,I,2).NE.I2).AND.
     *  (IDTV(H,I,3).NE.I2)).OR.
     * ((IDTV(H,I,1).NE.I3).AND.(IDTV(H,I,2).NE.I3).AND.
     *  (IDTV(H,I,3).NE.I3))) GOTO 10
C                            MATCH FOUND:
      RETURN
```

```
10    CONTINUE
C                              UNIQUE:
40    ITTP=ITTP+1
      DO 50 I=1,3
      IF (IVCHK(ID(I)).EQ.1) GOTO 50
      IVCHK(ID(I))=1
      IF (ICCT(ID(I)).LT.3) GOTO 50
      ITTV=ITTV+1
      IDTT(H,ITTV)=ID(I)
50    IDTV(H,ITTP,I)=ID(I)
C
C                              TEST TIELINES FOR EQUIVALENCE:
      DO 80 I=1,2
      I1=ID(IBC(I+1,1))
      I2=ID(IBC(I+1,2))
      IF (IPSF.EQ.0) GOTO 25
      DO 20 J=ISTART,IPSF
      IF (IDPSF(J,3).EQ.0) GOTO 20
      IF (((IDPSF(J,1).NE.I1) .AND.
     *     (IDPSF(J,2).NE.I1)).OR.
     *    ((IDPSF(J,1).NE.I2) .AND.
     *     (IDPSF(J,2).NE.I2))) GOTO 20
      IDPSF(J,3)=0
      GOTO 80
20    CONTINUE
C                              TEST WITH BINARY JOINS
25    GOTO (60),HGO
      DO 30 J=1,3
      L=ITBH(J)
      IB=IBVCT(L)-1
      DO 70 K=1,IB
70    IF (((IDBV(L,K)  .EQ.I1) .OR.
     *     (IDBV(L,K+1).EQ.I1)).AND.
     *    ((IDBV(L,K)  .EQ.I2) .OR.
     *     (IDBV(L,K+1).EQ.I2))) GOTO 80
30    CONTINUE
C                              ASSIGN NEW TIELINE:
60    IPSF=IPSF+1
      IDPSF(IPSF,1)=I1
      IDPSF(IPSF,2)=I2
      IDPSF(IPSF,3)=ID(ILK(I))
80    CONTINUE
      RETURN
      END
      SUBROUTINE ASSD4 (ISTART,HGO)
C-----------------------------------------------------------------------
C ASSD4 COUNTS (IQFCTH) AND ASSIGNS (IDQF) QUATERNARY FACETS. THE
C VERTICES ARE IDENTIFIED BY THE ARRAY ID(4).
C
C   REFERENCED BY: SIMPL4
C   REFERENCES TO: NONE
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      DIMENSION ILK(4)
C
      COMMON/ CST47 /IDQF(5,200,4),IQFCT(5),IQFCTH,IGO
     *      / CST56 /IDTQ(5,99),ITQCT(5),IQVH/ CST18 /ICCT(800)
     *      / CST52 /HCP,H,ID(6),IQC1,IQC2,IQC3,IQC4,IQC5
     *      / CST16 /IDTV(10,1296,3),IDTT(10,800),ITPCT(10),ITTCT(10)
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *      / CST46 /IPSF,IDPSF(2000,5)
C
      DATA ILK/0,3,2,1/
C-----------------------------------------------------------------------
```

```
      IF (IQFCTH.LT.1) GOTO 40
C                              TEST FOR EQUIVALENCE WITH EARLIER
C                              FACETS:
      DO 10 I=1,IQFCTH
      DO 20 J=1,4
      IDI=IDQF(H,I,J)
      DO 30 K=1,4
30    IF (IDI.EQ.ID(K)) GOTO 20
      GOTO 10
20    CONTINUE
      IGO=1
      RETURN
10    CONTINUE
C                              UNIQUE FACET:
40    IGO=0
      IQFCTH=IQFCTH+1
      DO 50 I=1,4
      IDI=ID(I)
C                              SET FLAG IVCHK FOR NEW VERTICES:
      IF ((ICCT(IDI).LT.4).OR.(IVCHK(IDI).EQ.1)) GOTO 50
      IQVH=IQVH+1
      IVCHK(IDI)=1
      IDTQ(H,IQVH)=IDI
50    IDQF(H,IQFCTH,I)=IDI
C                              EACH NEW FACET MAY GENERATE 3 NEW PST'S
C                              DEFINED BY THE VERTICES (124),(134),
C                              AND (234).
      DO 60 I=2,4
C                              SET VERTICES IDENTITIES:
      I1=ID(ITC(I,1))
      I2=ID(ITC(I,2))
      I3=ID(ITC(I,3))
C                              BEGIN BY TESTING EACH POTENTIAL PST
C                              FOR A MATCH WITH ONE ALREADY DEFINED.
      IF (IPSF.EQ.0) GOTO 70
      DO 80 J=ISTART,IPSF
      IF ((IDPSF(J,1).EQ.0).OR.
     *   ((IDPSF(J,1).NE.I1).AND.(IDPSF(J,2).NE.I1).AND.
     *    (IDPSF(J,3).NE.I1)).OR.
     *   ((IDPSF(J,1).NE.I2).AND.(IDPSF(J,2).NE.I2).AND.
     *    (IDPSF(J,3).NE.I2)).OR.
     *   ((IDPSF(J,1).NE.I3).AND.(IDPSF(J,2).NE.I3).AND.
     *    (IDPSF(J,3).NE.I3))) GOTO 80
C                              MATCHED PST.
      IDPSF(J,1)=0
      GOTO 60
80    CONTINUE
C                              TEST WITH TERNARIES IF SIMPL4 IS
C                              EXECUTING THE FIRST SEARCH SEG-
C                              MENT (HGO=0).
70    GOTO (90),HGO
      DO 100 J=1,4
      IQTHJ=IQTH(J)
      IF (IQTHJ.EQ.HI) GOTO 100
      HH=ITPCT(IQTHJ)
      DO 110 K=1,HH
      IF (
     * ((IDTV(IQTHJ,K,1).NE.I1).AND.(IDTV(IQTHJ,K,2).NE.I1).AND.
     *  (IDTV(IQTHJ,K,3).NE.I1)).OR.
     * ((IDTV(IQTHJ,K,1).NE.I2).AND.(IDTV(IQTHJ,K,2).NE.I2).AND.
     *  (IDTV(IQTHJ,K,3).NE.I2)).OR.
     * ((IDTV(IQTHJ,K,1).NE.I3).AND.(IDTV(IQTHJ,K,2).NE.I3).AND.
     *  (IDTV(IQTHJ,K,3).NE.I3))) GOTO 110
      GOTO 60
110    CONTINUE
```

```
100   CONTINUE
C                               UNIQUE PST:
90    IPSF=IPSF+1
      IDPSF(IPSF,1)=I1
      IDPSF(IPSF,2)=I2
      IDPSF(IPSF,3)=I3
      IDPSF(IPSF,4)=ID(ILK(I))
C
60    CONTINUE
      RETURN
      END
      SUBROUTINE ASSD5 (ISTART,HGO)
C-----------------------------------------------------------------
C ASSD5 COUNTS (ICFCT) AND ASSIGNS (IDCF) QUINARY FACETS. THE
C VERTICES ARE IDENTIFIED BY THE ARRAY ID(5).
C
C   REFERENCED BY: SIMPL5
C   REFERENCES TO: NONE
C-----------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      DIMENSION ILK(5)
C
      COMMON/ CST46 /IPSF,IDPSF(2000,5)/ CST58 /IDCF(200,5),ICFCT
     *      / CST52 /HCP,H,ID(6),IQC1,IQC2,IQC3,IQC4,IQC5
     *      / CST47 /IDQF(5,200,4),IQFCT(5),IQFCTH,IGO
     *      / CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *      / CST27 /IQAC(5,4),IQACT(5),IQICT,IDQ(5,10),IDQI(5)
C
      DATA ILK/0,4,3,2,1/
C-----------------------------------------------------------------
      MI=H
      IF (ICFCT.LT.1) GOTO 40
C                               TEST FOR EQUIVALENCE WITH EARLIER
C                               FACETS:
      DO 10 I=1,ICFCT
      DO 20 J=1,5
      IDI=IDCF(I,J)
      DO 30 K=1,5
30    IF (IDI.EQ.ID(K)) GOTO 20
      GOTO 10
20    CONTINUE
C                               MATCHES EARLIER FACET, RETURN.
      RETURN
10    CONTINUE
C                               UNIQUE FACET:
40    ICFCT=ICFCT+1
      DO 50 I=1,5
50    IDCF(ICFCT,I)=ID(I)
C                               EACH NEW FACET MAY GENERATE 3 NEW PST'S
C                               DEFINED BY THE VERTICES (124),(134),
C                               AND (234).
      DO 60 I=2,5
C                               SET VERTICES IDENTITIES:
C                               BEGIN BY TESTING EACH POTENTIAL PST
C                               FOR A MATCH WITH ONE ALREADY DEFINED.
      IF (IPSF.EQ.0) GOTO 70
      DO 80 J=ISTART,IPSF
      DO 90 K=1,4
      DO 100 L=1,4
100   IF (ID(IQAC(I,L)).EQ.IDPSF(J,K)) GOTO 90
      GOTO 80
90    CONTINUE
C                               MATCHED AN EARLIER PSQ:
      IDPSF(J,1)=0
      GOTO 60
80    CONTINUE
C                             TEST WITH TERNARIES IF SIMPL4 IS
C                             EXECUTING THE FIRST SEARCH SEG-
C                             MENT (HGO=0).
70    GOTO (110),HGO
      DO 120 J=1,5
      IF (J.EQ.MI) GOTO 120
      IB=IQFCT(J)
      DO 130 K=1,IB
      DO 140 L=1,4
      DO 150 M=1,4
150   IF (ID(IQAC(I,L)).EQ.IDQF(J,K,M)) GOTO 140
      GOTO 130
140   CONTINUE
C                            MATCH WITH A QUATERNARY FACET
      GOTO 60
130   CONTINUE
120   CONTINUE
C                             UNIQUE PST:
110   IPSF=IPSF+1
      DO 160 J=1,4
160   IDPSF(IPSF,J)=ID(IQAC(I,J))
      IDPSF(IPSF,5)=ID(ILK(I))
C
60    CONTINUE
      RETURN
      END
      SUBROUTINE BOUNDC
C-----------------------------------------------------------------------
C A SUBROUTINE WHICH CALCULATES THE PARAMETERS FOR TEST
C I.E. THE SLOPES OF THE TIELINES CONNECTING TERNARY VERTICES
C AND THE "SIGN" OF INCLUDED POINTS.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST12 /CP(800,5)/ CST45 /ITBH(3),ITT,ITTV,ITTP
     *      / CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IER
     *      / CST52 /HCP,H,ID(6),ITCC(5)
C-----------------------------------------------------------------------
C                              LOAD THE MATRIX V-TRANSPOSE
      DO 10 I=1,HCP
      DO 20 J=1,HCP
20    A(I,J)=CP(ID(J),ITCC(I))
10    CONTINUE
      CALL FACTOR (HCP)
C
      GOTO (99),IER
      RETURN
99    WRITE (6,*) ' **ERROR BOU002**'
      STOP
      END
      FUNCTION ITESTC (LD,A,IPVT)
C-----------------------------------------------------------------------
C ITESTC DETERMINES WHETHER A POINT LIES WITHIN THE PORTION
C OF A PLANE BOUNDED BY THE PLANES CONNECTING C POINTS.
C THE C POINTS.  THE IDENTITY OF THE
C POINT TO BE TESTED IS GIVEN BY LD.
C
C ITEST RETURNS 0 IF THE POINT IS WITHIN THE PLANE, 1 IF IT IS
C OUTSIDE, AND 2 IF IT LIES ON AN EDGE OF THE BOUNDED REGION.
C
C   REFERENCED BY: SIMPL4, SIMPL5
C   REFERENCES TO: FACTOR, SUBST
C-----------------------------------------------------------------------
```

```
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-N),INTEGER (Z)
C
      COMMON/ CST12 /CP(800,5)/ CST52 /IA ,H,HD,ID,JD,KD,LL,MD,ITCC(5)
C
      DIMENSION A(8,8),IPVT(8),B(8),X(8)
C------------------------------------------------------------------------
C                                 SOLVE LY=B FOR Y:
      IP=IPVT(1)
      X(1)=CP(LD,ITCC(IP))
      DO 10 I=2,N
      SUM=0.D0
      IM1=I-1
      DO 20 J=1,IM1
20    SUM=A(I,J)*X(J)+SUM
      IP=IPVT(I)
10    X(I)=CP(LD,ITCC(I))-SUM
C                                 SOLVE UX=Y FOR X:
      X(N)=X(N)/A(N,N)
      IF (X(N).LT.-1.D-05) GOTO 50
      NM1=N-1
      DO 30 II=1,NM1
      I=N-II
      IP1=I+1
      SUM=0.D0
      DO 40 J=IP1,N
40    SUM=A(I,J)*X(J)+SUM
      X(I)=(X(I)-SUM)/A(I,I)
30    IF (X(I).LT.-1.D-05) GOTO 50
      ITESTC=0
      RETURN
50    ITESTC=1
C
      RETURN
      END
      SUBROUTINE BOUND3
C------------------------------------------------------------------------
C A SUBROUTINE WHICH CALCULATES THE PARAMETERS FOR TEST
C I.E. THE SLOPES OF THE TIELINES CONNECTING TERNARY VERTICES
C AND THE "SIGN" OF INCLUDED POINTS.
C------------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST3 /X(800,5)/ CST17 /S(3),B(3),DY(3),XB(3),DX(3)
     *       / CST45 /ITBH(3),ITT,ITTV,ITTP
     *       / CST52 /HCP,H,I1,I2,I3,KD,LD,MD,ITC1,ITC2,ITC3,ITC4,ITC5
C------------------------------------------------------------------------
      DXX=(X(I2,ITC3)-X(I1,ITC3))
      IF (DXX.NE.0.D0) GOTO 5
      DX(1)=X(I1,ITC3)-X(I3,ITC3)
      XB(1)=X(I1,ITC3)
      S(1)=1.D60
      GOTO 10
5     S(1)=(X(I2,ITC2)-X(I1,ITC2))/DXX
      B(1)=X(I2,ITC2)-S(1)*X(I2,ITC3)
      DY(1)=S(1)*X(I3,ITC3)+B(1)-X(I3,ITC2)
10    DXX=(X(I3,ITC3)-X(I1,ITC3))
      IF (DXX.NE.0.D0) GOTO 15
      DX(2)=X(I1,ITC3)-X(I2,ITC3)
      XB(2)=X(I1,ITC3)
      S(2)=1.D60
      GOTO 20
15    S(2)=(X(I3,ITC2)-X(I1,ITC2))/DXX
      B(2)=X(I3,ITC2)-S(2)*X(I3,ITC3)
      DY(2)=S(2)*X(I2,ITC3)+B(2)-X(I2,ITC2)
20    DXX=(X(I3,ITC3)-X(I2,ITC3))
```

```
      IF (DXX.NE.0.D0) GOTO 25
      S(3)=1.D60
      DX(3)=X(I2,ITC3)-X(I1,ITC3)
      XB(3)=X(I2,ITC3)
      GOTO 30
25    S(3)=(X(I3,ITC2)-X(I2,ITC2))/DXX
      B(3)=X(I3,ITC2)-S(3)*X(I3,ITC3)
      DY(3)=S(3)*X(I1,ITC3)+B(3)-X(I1,ITC2)
30    RETURN
      END
      FUNCTION ITEST3 (LD)
C-----------------------------------------------------------------------
C ITEST3 DETERMINES WHETHER A POINT LIES WITHIN THE PORTION
C OF A PLANE BOUNDED BY THE LINES CONNECTING 3 POINTS.
C THE C POINTS.  THE IDENTITY OF THE POINT TO BE TESTED IS GIVEN BY LD.
C
C ITEST3 RETURNS 0 IF THE POINT IS WITHIN THE PLANE, 1 IF IT IS
C OUTSIDE, AND 2 IF IT LIES ON AN EDGE OF THE BOUNDED REGION.
C
C   REFERENCED BY: SIMPL3
C   REFERENCES TO: SUBST,FACTOR
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST3 /X(800,5)/ CST17 /S(3),B(3),DY(3),XB(3),DX(3)
     *      / CST45 /ITBH(3),ITT,ITTV,ITTP
     *      / CST52 /HCP,H,HD,ID,JD,KD,LL,MD,ITC1,ITC2,ITC3,ITC4,ITC5
C-----------------------------------------------------------------------
      LT=1
      DO 10 I=1,3
      IF (S(I).NE.1.D60) GOTO 30
      T=DX(I)*(XB(I)-X(LD,ITC3))
      GOTO 20
30    T=DY(I)*((S(I)*X(LD,ITC3)+B(I))-X(LD,ITC2))
20    IF (DABS(T).LT.1.D-08) LT=2
      IF (T.GE.0.D0) GOTO 10
      ITEST3=1
      RETURN
10    CONTINUE
      ITEST3=0
      GOTO (99),LT
      ITEST3=2
99    RETURN
      END
      SUBROUTINE ABLOAD (*)
C-----------------------------------------------------------------------
C ABLOAD ASSEMBLES THE MATRIX 'A' AND VECTOR 'B' FOR HCP COMPONENT
C SYSTEMS AND THEN SOLVES THE EQUATION AX=B, THE VECTOR X IS
C RETURNED IN 'B'.
C
C   REFERENCED BY: SIMPL2,SIMPL3,SIMPL4,SIMPL5
C   REFERENCES TO: SUBST,FACTOR
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *      / CST2 /G(800)/ CST12 /CP(800,5)/ CST52 /HCP,H,ID(6),IC(5)
C-----------------------------------------------------------------------
      DO 10 I=1,HCP
      DO 20 J=1,HCP
20    A(I,J)=CP(ID(I),IC(J))
10    CONTINUE
      CALL FACTOR (A,HCP,IPVT,IER)
      GOTO (99),IER
      DO 30 I=1,HCP
```

```
30    B(I)=G(ID(I))
      CALL SUBST (A,IPVT,HCP,B)
      RETURN
99    RETURN 1
      END
      SUBROUTINE FACTOR (A,N,IPVT,IER)
C-----------------------------------------------------------------------
C FACTOR IS A SUBROUTINE WHICH CALCULATES THE TRIANGULAR
C DECOMPOSITIONS OF THE MATRIX 'A'. FACTOR IS MODIFIED FROM
C THE SUBROUTINE OF THE SAME NAME GIVEN BY CONTE AND DE BOOR
C IN 'ELEMENTARY NUMERICAL ANALYSIS', MCGRAW-HILL, 1980.
C FACTOR USES SCALED PARTIAL PIVOTING.
C
C INPUT      A- AN N BY N ARRAY CONTAINING THE ELEMENTS OF MATRIX A.
C            N- THE DIMENSION OF THE MATRIX A.
C OUTPUT     A- AN N BY N ARRAY CONTAINING THE UPPER, U, AND LOWER, L,
C               TRIANGULAR DECOMPOSITIONS OF INPUT MATRIX A.
C         IPVT- A VECTOR INDICATING THAT ROW IPVT(K) WAS USED TO
C               ELIMINATE THE A(N,K).
C          IER- A FLAG, ZERO IF A IS OF RANK = N, AND 1 IF A IS OF
C               LOWER RANK.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-N),INTEGER (Z)
C
      DIMENSION A(8,8),D(8),IPVT(8)
C-----------------------------------------------------------------------
      IER=0
C                                 INITIALIZE IPVT,D
      DO 10 I=1,N
      IPVT(I)=I
      RMAX=0.D0
      DO 20 J=1,N
20    RMAX=DMAX1(RMAX,DABS(A(I,J)))
C                                 AX=B IS SINGULAR IF RMAX=0
      IF (RMAX.EQ.0.D0) GOTO 9000
10    D(I)=RMAX
C                                 BEGIN DECOMPOSITION:
      NM1=N-1
      DO 30 I=1,NM1
      IP1=I+1
C                                 DETERMINE PIVOT ROW (ISTR).
      RMAX=DABS(A(I,I))/D(I)
      ISTR=I
      DO 40 J=IP1,N
      TMAX=DABS(A(J,I))/D(J)
      IF (TMAX.LE.RMAX) GOTO 40
      RMAX=TMAX
      ISTR=J
40    CONTINUE
      IF (RMAX.EQ.0.D0) GOTO 9000
C                                 IF ISTR GT I, MAKE I THE PIVOT ROW
C                                 BY INTERCHANGING IT WITH ROW ISTR.
      IF (ISTR.LE.I) GOTO 50
      J=IPVT(ISTR)
      IPVT(ISTR)=IPVT(I)
      IPVT(I)=J
      TEMP=D(ISTR)
      D(ISTR)=D(I)
      D(I)=TEMP
      DO 60 J=1,N
      TEMP=A(ISTR,J)
      A(ISTR,J)=A(I,J)
60    A(I,J)=TEMP
C                                 ELIMINATE X(K) FROM ROWS K+1,...,N.
50    DO 70 J=IP1,N
```

```
      A(J,I)=A(J,I)/A(I,I)
      RATIO=A(J,I)
      DO 80 K=IP1,N
80    A(J,K)=A(J,K)-RATIO*A(I,K)
70    CONTINUE
30    CONTINUE
      IF (A(N,N).EQ.0.D0) IER=1
      RETURN
C                                     ALGORITHMIC SINGULARITY.
9000  IER=1
      RETURN
      END
      SUBROUTINE SUBST (A,IPVT,N,B)
C-----------------------------------------------------------------------
C SUBST USES THE LU DECOMPOSITION OF THE MATRIX 'A' CONTAINED
C IN THE ARRAY 'A' TO SOLVE AX=B FOR X. SUBST IS MODIFIED FROM THE
C THE SUBROUTINE OF THE SAME NAME LISTED BY CONTE AND DE BOOR
C IN 'ELEMENTARY NUMERICAL ANALYSIS', MCGRAW-HILL, 1980.
C
C INPUT      A- AN N BY N ARRAY CONTAINING THE NON-ZERO ELEMENTS OF
C               THE U AND L DECOMPOSITIONS OF A, AS OUTPUT BY FACTOR.
C            N- THE DIMENSION OF THE MATRIX A.
C         IPVT- A VECTOR INDICATING THAT ROW IPVT(K) WAS USED TO
C               ELIMINATE THE COEFFICIENT A(N,K).
C            B- THE VECTOR B.
C OUTPUT     B- THE SOLUTION VECTOR X.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-N),INTEGER (Z)
C
      DIMENSION A(8,8),IPVT(8),B(8),X(8)
C-----------------------------------------------------------------------
C                                      SOLVE LY=B FOR Y:
      IP=IPVT(1)
      X(1)=B(IP)
      DO 10 I=2,N
      SUM=0.D0
      IM1=I-1
      DO 20 J=1,IM1
20    SUM=A(I,J)*X(J)+SUM
      IP=IPVT(I)
10    X(I)=B(IP)-SUM
C                                      SOLVE UX=Y FOR X:
      X(N)=X(N)/A(N,N)
      NM1=N-1
      DO 30 II=1,NM1
      I=N-II
      IP1=I+1
      SUM=0.D0
      DO 40 J=IP1,N
40    SUM=A(I,J)*X(J)+SUM
      X(I)=(X(I)-SUM)/A(I,I)
30    B(I)=X(I)
      B(N)=X(N)
C
      RETURN
      END
C-----------------------------------------------------------------------
C THE DGPHC FUNCTIONS DETERMINE THE G DIFFERENCE BETWEEN A PHASE
C IDENTIFIED BY 'LD', AND THE PLANE DEFINED BY 'U'.
C   REFERENCED BY: SIMPL3,SIMPL4,SIMPL5
C-----------------------------------------------------------------------
      FUNCTION DGPH3 (LD)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST23 /A(64),U(9),I(20)/ CST2 /G(800)/ CST12 /CP(800,5)
     *      / CST52 /L(8),ITC1,ITC2,ITC3,ITC4,ITC5
```

```
C
      DGPH3=G(LD)-CP(LD,ITC1)*U(1)
     *             -CP(LD,ITC2)*U(2)
     *             -CP(LD,ITC3)*U(3)
      RETURN
      END
C
      FUNCTION DGPH4 (LD)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST23 /A(64),U(9),I(20)/ CST2 /G(800)/ CST12 /CP(800,5)
     *       / CST52 /L(8),ITC1,ITC2,ITC3,ITC4,ITC5
C
      DGPH4=G(LD)-CP(LD,ITC1)*U(1)
     *             -CP(LD,ITC2)*U(2)
     *             -CP(LD,ITC3)*U(3)
     *             -CP(LD,ITC4)*U(4)
      RETURN
      END
C
      FUNCTION DGPH5 (LD)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST23 /A(64),U(9),I(20)/ CST2 /G(800)/ CST12 /CP(800,5)
C
      DGPH5=G(LD)-CP(LD,1)*U(1)
     *             -CP(LD,2)*U(2)
     *             -CP(LD,3)*U(3)
     *             -CP(LD,4)*U(4)
     *             -CP(LD,5)*U(5)
      RETURN
      END
      FUNCTION SID3 (KD)
C---------------------------------------------------------------------
C GIVEN THE PARAMETERS OF A PLANE SPANNING THE JOIN OF A C-1
C DIMENSIONAL SIMPLEX, E.G. ID1,...,IDC-1, SIDC DETERMINES IF
C A POINT 'KD' LIES ON THE SAME SIDE OF THE JOIN AS THE VERTEX
C IDC. PARAMETERS FOR THE SIMPLEX ID1,...,IDC ARE INITIALIZED
C BY SLOPC.
C SIDC IS 0 IF KD LIES IN THE PLANE, LT 0 IF KD LIES ON THE OPPOSITE
C SIDE OF THE JOIN FROM IDC, AND GT 0 IF KD IS ON THE SAME SIDE AS
C IDC.
C---------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
C
      COMMON/ CST3 /X(800,5)/ CST55 /SLOPE,SS(2),B,SIGN,LGO
     *       / CST52 /HCP,H,ID(6),ITC1,ITC2,ITC3,ITC4,ITC5
C---------------------------------------------------------------------
      GOTO (10),LGO
      SID3=(SLOPE*X(KD,ITC1)+B-X(KD,ITC2))/SIGN
      RETURN
C
10    SID3=(B-X(KD,ITC1))/SIGN
      RETURN
      END
C
      FUNCTION SID4 (KD)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST55 /S1,S2,S3,S4,SIGN,LGO/ CST3 /X(800,5)
     *       / CST52 /HCP,H,ID(6),IQC1,IQC2,IQC3,IQC4,IQC5
C                                 LGO IS A FLAG WHICH INDICATES
C                                 WHICH COMPONENTS WERE USED TO
C                                 DEFINE THE PLANE IN THE SIMPLEX.
      SID4=(S1*X(KD,IQC1)+S2*X(KD,LGO)+S4-X(KD,IQC3))/SIGN
C
      RETURN
      END
```

```
C
      FUNCTION SID5 (KD)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST55 /S1,S2,S3,S4,SIGN,LGO/ CST3 /X(800,5)
C                             LGO IS A FLAG WHICH INDICATES
C                             WHICH COMPONENTS WERE USED TO
C                             DEFINE THE PLANE IN THE SIMPLEX.
      SID5=(S1*X(KD,1)+S2*X(KD,2)+S3*X(KD,LGO)+S4-X(KD,5))/SIGN
      RETURN
      END
C-----------------------------------------------------------------------
C GIVEN THE JOIN OF A SIMPLEX OF C-1 DIMENSIONS, E.G. ID1,...,IDC-1,
C SLOPC FINDS THE EQUATION OF THE PLANE SPANNING THE JOIN. THE
C DISPLACEMENT OF THE VERTEX NOT ON THE JOIN, IDC, IS DETERMINED
C RELATIVE TO THIS PLANE (I.E. THE PARAMETER 'SIGN').
C-----------------------------------------------------------------------
      SUBROUTINE SLOP3 (J)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST3 /X(800,5)/ CST55 /SLOPE,SS(2),B,SIGN,LGO
     *      / CST52 /HCP,H,HD,ID,JD,KD,LD,MD,ITC1,ITC2,ITC3,ITC4,ITC5
     *      / CST46 /IPSF,IDPSF(2000,5)
C
      I3=IDPSF(J,3)
      DX=X(HD,ITC1)-X(ID,ITC1)
      IF (DX.EQ.0.D0) GOTO 10
      LGO=0
      SLOPE=(X(HD,ITC2)-X(ID,ITC2))/DX
      B=-SLOPE*X(HD,ITC1)+X(HD,ITC2)
      SIGN=SLOPE*X(I3,ITC1)+B-X(I3,ITC2)
      RETURN
C
10    LGO=1
C                             THE LINE IS PARALLEL TO THE ITC1
C                             COMPONENT:
      SIGN=X(HD,ITC1)-X(I3,ITC1)
      B=X(HD,ITC1)
      RETURN
      END
C
      SUBROUTINE SLOP4 (J)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST3 /X(800,5)/ CST55 /S1,S2,S3,S4,SIGN,LGO
     *      / CST52 /HCP,H,ID(6),IQC1,IQC2,IQC3,IQC4,IQC5
     *      / CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *      / CST46 /IPSF,IDPSF(2000,5)
      DATA ICT/3/
C                             LOAD MATRIX AND VECTORS
      I4=IDPSF(J,4)
      LGO=IQC2
40    DO 10 I=1,3
      A(I,3)=1
      A(I,1)=X(IDPSF(J,I),IQC1)
      A(I,2)=X(IDPSF(J,I),LGO)
10    B(I)  =X(IDPSF(J,I),IQC3)
C                             SOLVE FOR THE PLANE:
      CALL FACTOR (A,ICT,IPVT,IER)
C                             IF PARALLEL TO A COMPONENT THEN
C                             GO TO 30 (SWITCH COMPONENTS).
      GOTO (30),IER
C
      CALL SUBST (A,IPVT,ICT,B)
C                             SAVE PLANE PARAMETERS:
      S1=B(1)
      S2=B(2)
      S4=B(3)
```

```
C                                COMPUTE SIGN:
      SIGN=S1*X(I4,IQC1)+S2*X(I4,LGO)+S4-X(I4,IQC3)
      LGO=0
      RETURN
C                                SWITCH COMPONENTS:
30    LGO=IQC4
      GOTO 40
      END
C
      SUBROUTINE SLOP5 (J)
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      COMMON/ CST3 /X(800,5)/ CST55 /S1,S2,S3,S4,SIGN,LGO
     *       / CST52 /HCP,H,ID(6),IQC1,IQC2,IQC3,IQC4,IQC5
     *       / CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *       / CST46 /IPSF,IDPSF(2000,5)
      DATA ICT/4/
C                                LOAD MATRIX AND VECTORS
      LGO=3
      I5=IDPSF(J,5)
20    DO 10 I=1,4
      A(I,1)=X(IDPSF(J,I),1)
      A(I,2)=X(IDPSF(J,I),2)
      A(I,3)=X(IDPSF(J,I),LGO)
      A(I,4)=1.D0
10    B(I)  =X(IDPSF(J,I),5)
C                                SOLVE FOR THE PLANE:
      CALL FACTOR (A,ICT,IPVT,IER)
C                                IF A IS SINGULAR (IER=1) THEN
C                                SWITCH THE COMPONENTS AND RELOAD.
      GOTO (30),IER
      CALL SUBST (A,IPVT,ICT,B)
C                                SAVE PARAMETERS:
      S1=B(1)
      S2=B(2)
      S3=B(3)
      S4=B(4)
C                                COMPUTE SIGN:
      SIGN=S1*X(I5,1)+S2*X(I5,2)+S3*X(I5,LGO)+S4-X(I5,5)
      RETURN
30    LGO=4
      GOTO 20
      END
      SUBROUTINE MISCIB (ICT)
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER (Z),INTEGER*2 (H-M)
C
      DIMENSION IOK(2)
C
      COMMON/ CST19 /HI,IQTH(4),IVCHK(800),ITC(10,3)
     *       / CST41 /N1,N2,N3,N4,N5,N6,N7,N8,IO3,IO4,IO5
     *       / CST23 /A(8,8),B(9),IPVT(8),IDV(8),IOPHI,IDPHI,IIPHI,IFLG1
     *       / CST12 /CP(800,5)/ CST52 /HCP,H,ID(6),IC(5)
     *       / CST60 /IKP(800),IASMBL(1273),IPOINT,IMYN
     *       / CST6 /ICOMP,ISTCT,IPHCT,ICP
C-----------------------------------------------------------------------
C                                INITIALIZE FLAGS AND COUNTERS
      IOK(1)=0
      IHOM=1
      IOK(2)=0
      IM=0
      IASMBL(ICT)=1
      II=1
C                                TEST IF THE PHASE IDV(II) IS A
C                                A PSEUDO-COMPOUND NOT PREVIOUSLY
C                                MATCHED
```

```
50    IKT=IKP(ID(II))
      IF ((IKT.EQ.0).OR.(IKT.EQ.IOK(1)).OR.(IKT.EQ.IOK(2))) GOTO 20
C                                LOOK FOR A MATCH
      JJ=II+1
      DO 30 J=JJ,HCP
      IF (IKP(ID(J)).NE.IKT) GOTO 30
C                                TEST FOR POSSIBILITY OF A HOMOGENEOUS
C                                PHASE
      IF (JJ.NE.2) GOTO 40
      IHOM=IHOM+1
30    CONTINUE
      IF (IHOM.EQ.1) GOTO 20
      GOTO 40
C                                NO MATCH OR PREVIOUSLY FOUND
20    II=II+1
      IF (II.EQ.HCP) GOTO 99
      GOTO 50
C                                MATCH FOUND, SET FLAGS. NOTE THAT THE
C                                MAXIMUM NUMBER OF MATCHES IN A FIVE
C                                COMPONENT SYSTEM IS TWO, INCREASE THE
C                                DIMENSION OF IOK FOR LARGER SYSTEMS.
40    IM=IM+1
      IOK(IM)=IKT
      IASMBL(ICT)=2
C                                LOAD AND FACTOR THE TRANSPOSE OF THE
C                                CONCENTRATION MATRIX OF THE PSEUDO-
C                                INVARIANT ASSEMBLAGE.
      DO 70 I=1,HCP
      DO 80 J=1,HCP
80    A(J,I)=CP(ID(I),J)
70    CONTINUE
      CALL FACTOR (A,HCP,IPVT,IER)
      GOTO (9000),IER
C                                TEST FOR BOUNDED METASTABLE PSEUDO-CPDS
C                                OF THE SOLUTION IKT
      DO 60 J=ISTCT,IPHCT
      IF ((IVCHK(J).EQ.1).OR.(IKP(J).NE.IKT)) GOTO 60
      DO 90 I=1,HCP
C                                LOAD COMPOSITION VECTOR, THE ALPHA
C                                VECTOR IS RETURNED IN THE SAME ARRAY
90    B(I)=CP(J,I)
C                                SOLVE FOR THE ALPHA VECTOR
      CALL SUBST (A,IPVT,HCP,B)
C                                A NEGATIVE COEFFICIENT INDICATES
C                                THE PHASE IS NOT BOUNDED
      DO 100 I=1,HCP
100   IF (B(I).LT.-1.D-05) GOTO 60
C                                THE PHASE IS BOUNDED, EXSOLUTION HAVE
C                                OCCURRED, TEST TO DETERMINE IF DEGEN-
C                                ERATE SOLVUS HAS BEEN FOUND
      IF (IHOM.EQ.HCP) GOTO 110
      DO 120 I=1,HCP
120   IF ((IKP(ID(I)).NE.IKT).AND.(B(I).GT.1.D-05)) GOTO 60
C
110   IASMBL(ICT)=4
      IMYN=0
      GOTO 99
C
60    CONTINUE
C                                 IF NO IMMISCIBILITY AND THE ASSEMBLAGE
C                                 IS A HOMOGENEOUS PHASE SET ASMBL FLAG
      IF (IHOM.EQ.HCP) IASMBL(ICT)=3
C
99    RETURN
C                                 ON ERROR
```

```
9000  WRITE (N8,*) '*** ERROR *** ROUTINE MISCIB'
      STOP
      END
      SUBROUTINE FOPEN
C-----------------------------------------------------------------------
C   FOPEN- A SUBROUTINE TO OPEN VM/SP FILES
C     VERTEX CONVENTIONS:
C          N1-CARD DECK INPUT (DISK)
C          N2-PRINTER OR TERMINAL OUTPUT
C          N3-GRAPHICS FILE (I/O, DISK)
C          N4-
C          N5-
C          N6-
C          N7-TERMINAL INPUT
C          N8-TERMINAL OUTPUT
C-----------------------------------------------------------------------
      IMPLICIT REAL*8 (A-G,O-Y),INTEGER*2 (H-M),INTEGER (Z)
      CHARACTER*8 FNAME,RECORD*80
      COMMON/ CST41 /N1,N2,N3,N4,N5,N6,N7,N8,I03,I04,I05
C                               ON VM/SP DEFINE FILES
C                               GRPHIN USES THREE UNITS:
C                               N2-TERMINAL OUTPUT
      N2=20
      CALL CMSF ('FI 20 TERM (LRECL 132',Z1)
C                               N1-DISK INPUT
      FNAME='
      N1=21
      CALL WRTRDF ('ENTER NAME OF INPUT DATA FILE',FNAME,
     *             Z1,Z2)
      WRITE (RECORD,250) N1,FNAME
      WRITE (N2,*) RECORD
      CALL CMSF (RECORD,Z1)
      FNAME='
C                               N3-DISK OUTPUT
      N3=22
      CALL WRTRDF ('ENTER NAME OF GRAPHICS OUTPUT DATA FILE',FNAME,
     *             Z1,Z2)
      WRITE (RECORD,250) N3,FNAME
      WRITE (N2,*) RECORD
      CALL CMSF (RECORD,Z1)
250   FORMAT ('FILEDEF ',I2,' DISK ',A8,
     *        ' DATA * (PERM RECFM FB LRECL 80 BLKSIZE 80')
C
      RETURN
      END
```