SOLVING THE CHEMICAL EQUILIBRIUM PROBLEM USING THE DECOMPOSITION PRINCIPLE

G. B. Dantzig
Marvin Shapiro

Mathematics Division
The RAND Corporation

P-2056

August 10, 1960

SUMMARY

The determination of the equilibrium composition of a gaseous mixture is equivalent to the determination of the number of moles of each molecular species present that minimize the total free energy of the mixture. The convex function for free energy as given by Gibbs is minimized subject to the constraints of the mass-balance equations for the mixture. A solution to the problem is given using a version of the decomposition procedure for linear programming. A mathematical description of the method of solution as well as the computational algorithm as programmed for the IBM 704 with some results are given.

SOLVING THE CHEMICAL EQUILIBRIUM PROBLEM USING THE DECOMPOSITION PRINCIPLE

George B. Dantzig and Marvin Shapiro

An earlier paper by one of the authors on Convex Objectives [5] provides the theoretical foundation for this report. It gave a proof of convergence for the case when an infinite master program is generated by the decomposition process [2]. In this paper the method is applied to the convex function of Gibbs which represents the free energy of a chemical system under constant temperature and pressure. This procedure, which is similar in some respects to that discussed in [4], was compared with the "quadratic fit" procedure [3] further developed by Selmer Johnson and Herschel Kanter and now in wide use. This procedure was extended by S. Johnson and one of the authors to the Multiphased Chemical Systems in [6] and the code described in Section 4 covers this case also. In each test problem tried the quadratic fit appeared to be more efficient. This was true even after correction for the use of single precision and the use of internal memory for the quadratic fit technique instead of the use of double precision and tape for the decomposition approach. Empirical experience at this point is too limited to say more than that on larger problems the two methods seemed more comparable. For example a problem involving 12 mass balance constraints and 30 molecular species, see [6], took 17 iterations which is equivalent to 17 x 12 = 204 iterations of the decomposition principle; actually the latter required 230.

However, the present method has certain advantages. It is more flexible because, with minor code changes, it can be extended (as will be developed in a future report) to solve a very broad class of problems; namely to find

$$\text{Min } G_0(x) \qquad x = (x_1, x_2, \ldots, x_n) \ ,$$

subject to m inequality constraints $G_i(x) \leq 0$ where $G_i(x)$ for $i = 0, 1, 2, \ldots, m$ are general convex functions in n variables $x = (x_1, x_2, \ldots, x_n)$ over a closed bounded convex domain $x \in R$.

1. GENERAL THEORY

We consider an equilibrium mixture containing m different atom types. While in theory these will combine into all chemically possible molecular species, in practice only standard types are considered, including the monotonic types, which are known to occur in measurable amounts.

Let

b_i = the number of atomic weights of species i present in the mixture,

x_j = the number of moles of molecular species j present in the mixture where

$$x_j \geq 0, \qquad j = 1, 2, \ldots, n, \tag{1}$$

$\bar{x}$ = the total number of moles of gas in the mixture, i.e.,

$$\bar{x} = \sum x_j, \tag{2}$$

a_{ij} = the number of atoms of species i in a molecule of species j.

Then the mass balance equations are

$$\sum_{j=1}^{n} a_{ij}x_j = b_i \quad \text{for} \quad i = 1, 2, \ldots, m. \tag{3}$$

The determination of the equilibrium composition of a gaseous mixture is equivalent to the determination of the values of the mole numbers x_j that obey constraints and minimize the total free energy of the mixture given by

$$(4)\quad G(x_1, \ldots, x_n) = \sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} x_j \log (x_j/\bar{x})$$

$$= \bar{x}\left[\sum_{j=1}^{n} c_j (x_j/\bar{x}) + \sum_{j=1}^{n} (x_j/\bar{x}) \log (x_j/\bar{x}) \right]$$

$$= \bar{x} \sum (c_j u_j + u_j \log u_j), \quad u_j = x_j/\bar{x} \geq 0,$$

which can be shown to be a convex function. The values c_j are modified Gibbs free energy function F^o/RT of the atomic species at a given temperature plus the natural logarithm of the pressure in atmospheres. Our problem is to minimize (4) subject to the linear equality and inequality constraints (1), (2), (3).

Homogeneous objectives of degree 1: In the chemical equilibrium problem $G(x)$ appears in the special form

$$(5)\quad G(x) = \bar{x} G\left(\frac{x_1}{\bar{x}}, \frac{x_2}{\bar{x}}, \ldots, \frac{x_n}{\bar{x}}\right)$$

where

$$(6)\quad \bar{x} = x_1 + x_2 + \ldots + x_n;$$

that is to say, G is a homogeneous function of degree 1.

Here

$$G(u) = \sum (c_j u_j + u_j \log u_j)$$

is easily seen to be a convex function.

In this case, we may rewrite our system (3) in the form, find $\bar{x} \geq 0$, and ratios $u_i = x_i/\bar{x} \geq 0$, Min z satisfying

$$(7)\quad \bar{x}\left[a_{11}u_1 + a_{12}u_2 + \ldots + a_{1n}u_n\right] = b_1$$

$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

$$\bar{x}\left[a_{m1}u_1 + a_{m2}u_2 + \ldots + a_{mn}u_m\right] = b_m$$

$$\bar{x}G(u) = z \text{ (Min)}$$

where

$$(8)\qquad u_1 + u_2 + \ldots + u_n = 1, \quad u_j \geq 0.$$

We regard system (7) as a single variable ($\bar{x}$) m-equation linear programming problem with variable coefficients

$$(9)\quad \alpha_i = \sum a_{ij}u_j, \qquad i = 1, 2, \ldots, m,$$

$$\gamma \geq G(u),$$

where u satisfies (8). It is easy to prove that if $G(u)$ is a convex function, the set of coefficients $[\alpha_1, \alpha_2, \ldots, \alpha_m; \gamma]$ are selected from a convex set. Hence we may apply the generalized coefficient method discussed in The RAND Corporation Paper P-1544 on the Decomposition Principle [2] and the

special application discussed in P-1664 on General Convex Objective Forms [5].

Iterative procedure: We assume the algorithm can be initiated with some m sets of vectors u^p satisfying (8), which generate

$$(10)\quad \alpha_{ip} = \sum_j a_{ij}u_j^p \qquad p = 1, 2, \ldots, m$$

$$\gamma_p = G(u^p).$$

such that $[\alpha_{ip}]$ forms a feasible basis for representing b, i.e. there exist $\lambda_p = \lambda_p^o \geq 0$, $\bar{z} = \bar{z}_0$ satisfying

$$(11)\quad \sum_p \alpha_{ip}\lambda_p = b_i, \qquad p = 1, 2, \ldots, m$$

$$\sum \gamma_p\lambda_p = \bar{z}.$$

The simplex multipliers $\pi_i = \pi_i^o$ satisfy

$$(12)\quad \sum_i \pi_i\alpha_{ip} = \gamma_p \qquad p = 1, 2, \ldots, m\ .$$

To test optimality, the expression

$$(13)\quad G(u) - \sum \pi_i^o \sum a_{ij}u_j = \Delta$$

is minimized over all $u_j \geq 0$ satisfying $\sum u_j = 1$. If Min $\Delta \geq 0$, the solution

$$(14)\quad x = \sum u^p\lambda_p^o$$

is optimum. If not, choose $u = u^{m+1}$ such that

$$(15)\quad G(u^{m+1}) - \sum \pi_i^o \sum a_{ij} u_j^{m+1} = \text{Min}\,\Delta.$$

Augment system (11) and find $\lambda_p = \lambda_p^1 \geq 0$, Min $\bar{z} = \bar{z}_1$ satisfying

$$(16)\quad \sum \alpha_{ip}\lambda_p + \alpha_{im+1}\lambda_{m+1} = b_i$$

$$\sum \gamma_p \lambda_p + \gamma_{m+1}\lambda_{m+1} = \bar{z}\ (\text{Min}),$$

where α_{im+1} and γ_{m+1} are found by substituting u^{m+1} in (10). The new simplex multipliers π_i^1 are substituted for π_i^o in expression (13) and Δ is minimized over all $\sum u_i = 1$, $u_i \geq 0$. The procedure is then iterated introducing more and more columns m+1, m+2, ... into (16).

Theorem:* Given a homogeneous convex objective form $G(x)$ of degree unity such that Inf $G(x)$ exists, then the modified algorithm converges to an optimal solution under ϵ-perturbation if there exists at least one nondegenerate basic solution to the "master" program (16).

For this application, then,

$$(17)\quad \Delta = G(u) - \sum \pi_i \sum a_{ij} u_j$$

$$= \sum (u_j \log u_j + \bar{c}_j u_j)$$

*A proof of this theorem will be found in The RAND Corporation P-1664. It is assumed Inf $G(x)$ exists, where x is in a convex set defined by (1) and (3).

where $\pi_i = \pi_i^k$ are the simplex multipliers of some iteration and

$$(18)\quad \bar{c}_j = c_j - \sum_{i=1}^{m} \pi_i a_{ij},$$

To find the Min Δ subject to

$$(19)\quad \sum u_j = 1, \qquad u_j \geq 0,$$

we ignore the relations $u_j \geq 0$ and find the unconditional minimum of the function

$$(20)\quad \bar{\Delta} = \sum (u_j \log u_j + \bar{c}_j u_j) - \theta \left(\sum u_j - 1\right)$$

where θ is a Lagrange multiplier. We set the partial derivatives of $\bar{\Delta}$ with respect to u_j equal to zero; thus

$$(21)\quad \frac{\partial \bar{\Delta}}{\partial u_j} = 0 = 1 - \theta + \log u_j + \bar{c}_j,$$

whence u_j may be written in the form

$$(22)\quad u_j = Ae^{-\bar{c}_j}$$

where $A = e^{\theta - 1} > 0$. Substituting into (19) determines $A > 0$ so that the conditions $u_j \geq 0$ hold at the minimum; hence

$$(23)\quad u_j = e^{-\bar{c}_j} \Big/ \sum_{1}^{n} e^{-\bar{c}_j} > 0.$$

These are the values $u_j = u_j^{k+1}$ to initiate the next iteration.

The value of Min Δ is obtained by substituting (23) into (17):

$$\text{Min } \Delta = \log A = -\log \sum e^{-\bar{c}_j}.$$

Letting $x \geq 0$ be any vector such that $Ax = b$, then

$$G(x) - \pi b = G(x) - \pi Ax = \bar{x}\left[G(u) - \pi Au\right] \geq \bar{x}\ \text{Min}\ \Delta$$

so that πb because of convexity and definition of π provides an upper bound for $G(\hat{x})$ and $\pi b - \hat{\bar{x}}$ Min Δ provides a lower bound where $\hat{x}$ is the optimum x and $\hat{\bar{x}}$ is the sum of components of $\hat{x}$. Hence Min Δ may be used as a measure of the maximum possible decrease in free energy per mole.

2. COMPUTATIONAL ALGORITHM

Find $x_j \geq 0$ and Min $G(x)$ satisfying

$$\begin{aligned} x_1 &+ a_{1m+1}x_{m+1} + \ldots + a_{1n}x_n = b_1 \\ &\ddots \\ &\ddots \\ &\ddots \\ &\qquad x_m + a_{mm+1}x_{m+1} + \ldots + a_{mn}x_n = b_m \end{aligned}$$

$$\sum x_j\,(c_j + \ell n\, x_j/\bar{x}) = G(X)$$

where $\bar{x} = x_1 + x_2 + \ldots + x_n$ and $b_i \geq 0$

We define $A = [a_{ij}]$, $b = (b_1, b_2, \ldots, b_m)$, column vector;

$x = (x_1, x_2, \ldots, x_n)$, column vector;

$Ax = \alpha$, column vector.

Each iteration p begins by computing (1) a "candidate" $x = x^p$; (2) $Ax^p = \alpha^p$; (3) $G(x^p) = \gamma^p$. A test is then made to see how close Min Δ is to zero. If Min Δ, which is a measure of the non-optimality of the solution, is within specified limits, say $h^* < 0$, the "solution" is printed out and the computation terminates. The information on hand at the start of the p^{th} iteration consists of m+3 columns of information of which m are associated with some subset $k = k_1, k_2, \ldots, k_m$ of m previous x^k including a special initiating set. For notational convenience below we will assume $k = 1, 2, \ldots, m$ and $p > m$ but will understand that k actually takes on some set of values k_i. Let $\alpha^k = (\alpha_{1k}, \ldots, \alpha_{mk})$. Then the data at hand at the start of an iteration can be obtained from the "information matrix."

$$\left[\begin{array}{cc|ccccc|c} 1 & 0 & \gamma_1 & \gamma_2 & \cdots & \gamma_m & 0 \\ 0 & 1 & 1 & 1 & \ldots & 1 & 0 \\ \hline 0 & 0 & \alpha_{11} & \alpha_{12} & \cdots & \alpha_{1m} & b_1 \\ 0 & 0 & \alpha_{21} & \alpha_{22} & \cdots & \alpha_{2m} & b_2 \\ & & \vdots & & & & \vdots \\ 0 & 0 & \alpha_{m1} & \alpha_{m2} & \cdots & \alpha_{mm} & b_m \end{array}\right]$$

The square array $B = [a_{ik}]$ is called the basis associated with the p^{th} iteration. It has a special property, namely, if we determine a column vector such that $B\lambda = b$, its components $\lambda_1 \geq 0$. Instead of recording explicitly the above matrix the set of "inverse" values are recorded

1	0	$-\pi_1$	$-\pi_2$	$-\pi_m$	$-g$
0	1	β_{01}	β_{02}	β_{0m}	$-d$
0	0	β_{11}	β_{12}	β_{1m}	λ_1
0	0	β_{21}	β_{22}	β_{2m}	λ_2
			⋮		⋮
0	0	β_{m1}	β_{m2}	β_{mm}	λ_m

← These values known at start of Cycle p:

where

$B^{-1} = [\beta_{ij}]$	$\lambda = B^{-1}b$
$\beta_o = \bar{x}^* [\beta_{ij}]$	$d = \beta_o b$ or $\bar{x}^* \lambda$
$\pi = \gamma [\beta_{ij}]$	$g = \pi b$ or $\gamma \lambda$

and $\bar{x}^* = (1, \ldots, 1)$

$\gamma = (\gamma_1, \ldots, _m)$

The above relations are not used in the computation; they are shown to give the relation between the "information" matrix and its "inverse."

To initiate the algorithm the special initating set is

$$x^{01} = (1, 0, 0, \ldots 0)$$

$x^{o2} = (0, 1, 0, \ldots 0)$

$\vdots$

$x^{om} = (0 \ldots 1, 0 \ldots 0)$

Thus the starting basis is the identity in the "information matrix" below

$$\left[\begin{array}{cc|ccccc|c} 1 & 0 & c_1 & c_2 & \cdots & c_m & 0 \\ 0 & 1 & 1 & 1 & \cdots & 1 & 0 \\ \hline 0 & 0 & 1 & 0 & \cdots & 0 & b_1 \\ 0 & 0 & 0 & 1 & \cdots & 0 & b_2 \\ & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & b_m \end{array}\right]$$

In inverse form, to start cycle 1 we have:

$$\begin{array}{cc|ccccc|c} 1 & 0 & -c_1 & -c_2 & \cdots & -c_m & -\sum c_i\, b_i \\ 0 & 1 & -1 & -1 & \cdots & -1 & -\sum b_i \\ \hline 0 & 0 & 1 & 0 & \cdots & 0 & b_1 \\ 0 & 0 & 0 & 1 & \cdots & 0 & b_2 \\ & & & \vdots & & & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 1 & b_m \end{array}$$

ITERATIVE PROCEDURE

Step I: Compute $\bar{c}_j = c_j - \sum_1^m \pi_i a_{ij}$ $(j=1,2,\ldots,n)$

$y_j = e^{-\bar{c}_j}$; $\bar{y} = \sum_1^n e^{-\bar{c}_j}$

$\alpha^p = \frac{1}{\bar{y}} Ay$ $y = (y_1, y_2, \ldots, y_n)$

$\gamma^p = \pi \alpha^p - \ell n \bar{y}$

Step II: Adjoin a new column to the set of inverse values

1	0	$-\pi_1$	$-\pi_2 \cdots -\pi_m$	$-g$	h
0	1	β_{01}	$\beta_{02} \cdots \beta_{0m}$	$-d$	e
0	0	β_{11}	$\beta_{12} \cdots \beta_{1m}$	λ_1	μ_1
0	0	β_{r1}	$\beta_{r2} \cdots \beta_{rm}$	λ_r	μ_p
0	0	β_{m1}	$\beta_{m2} \cdots \beta_{mm}$	λ_m	μ_m

where

$h = -\pi\alpha^p + \gamma^p = -\ell n\bar{y}$

$e = \beta_o \alpha^p + 1$

$\mu = [\beta_{ij}]\alpha^p$

Choose r so that

$$\frac{\lambda_r}{\mu_r} = \text{Min} \frac{\lambda_i}{\mu_i}$$

where μ_i, $\mu_r > 0$ and r, i=1,2,...,m

Step III: If $\bar{y} = 1$ (within an agreed upon error range) print out solution

$$x = \sum \lambda_i^p x_i \qquad i = 1, \ldots, m$$

where x_i is the solution x which generated i^{th} column $a^k{}_i$ in the basis. If $\bar{y} \neq 1$ go to step IV.

Step IV: Pivot on μ_r

Step V: Initiate new iteration with transformed tableau dropping adjoined column.

3. ALTERNATIVE COMPUTATIONAL PROCEDURE

The "information matrix" at the start of the p^{th} iteration is the same as before except that it is arranged in the form

$$\left[\begin{array}{ccc|cccc|c}
1 & 0 & 0 & 1 & 1 & \cdots\cdots & 1 & 0 \\
0 & 1 & 0 & \gamma_1 & \gamma_2 & \cdots\cdots & \gamma_m & 0 \\
0 & 0 & 1 & 1 & 1 & \cdots\cdots & 1 & 0 \\
\hline
0 & 0 & 0 & \alpha_{11}-b_1 & \alpha_{12}-b_1 & \cdots\cdots & \alpha_{1m}-b_1 & b_1 \\
0 & 0 & 0 & \alpha_{21}-b_2 & \alpha_{22}-b_2 & \cdots\cdots & \alpha_{2m}-b_2 & b_2 \\
 & \vdots & & \vdots & & & & \vdots \\
0 & 0 & 0 & \alpha_{m1}-b_m & \alpha_{m2}-b_m & \cdots\cdots & \alpha_{mm}-b_m & b_m
\end{array}\right]$$

For a basis, the square array

$$B = [\alpha_{ik}-b_i]$$

is used instead. The information is stored in "inverse" form (in the tableau below the adjoined column is also shown):

1	0	0	$\bar{\beta}_{-21}$	$\bar{\beta}_{-22}$	$\bar{\beta}_{-2m}$	$\bar{d}^1$	$\bar{e}^1$
0	1	0	$-\bar{\pi}_1$	$-\bar{\pi}_2$	$-\bar{\pi}_m$	$\bar{g}$	h
0	0	1	$\bar{\beta}_{01}$	$\bar{\beta}_{02}$	$\bar{\beta}_{0m}$	$\bar{d}$	$\bar{e}$
0	0	0	$\bar{\beta}_{11}$	$\bar{\beta}_{12}$	$\bar{\beta}_{1m}$	$\bar{\lambda}_1$	$\bar{\mu}_1$
0	0	0	$\bar{\beta}_{21}$	$\bar{\beta}_{22}$	$\bar{\beta}_{2m}$	$\bar{\lambda}_2$	$\bar{\mu}_2$
	⋮		⋮			⋮	⋮
0	0	0	$\bar{\beta}_{m1}$	$\bar{\beta}_{m2}$	$\bar{\beta}_{mm}$	$\bar{\lambda}_m$	$\bar{\mu}_m$

These values known at start of cycle, except last column.

where

$[\beta_{ij}] = \bar{B}^{-1}$	$\bar{\lambda} = \bar{B}^{-1} b$	$\bar{\mu} = \bar{B}^{-1} \alpha^p$
$\bar{B}_0 = \bar{x}^* \bar{B}^{-1}$	$\bar{d} = \bar{B}_0 b$	$\bar{e} = \beta_0 \alpha^p - \bar{x}^p$
$\bar{\beta}_{-2} = (1, 1, \ldots 1)B^{-1}$	$\bar{d}^1 = \beta_{-2} b$	$\bar{e}^1 = \beta_{-2} \alpha^p - 1$
$\bar{\pi} = \gamma B^{-1}$	$\bar{g} = \bar{\pi} b$	$\bar{h} = \bar{\pi} \alpha^p - \gamma_p$

The procedure is identical with that given earlier except add in an extra Step II. 5 as follows:

Step II. 5: Recover essential information of original tableau

$$\lambda = \bar{\lambda} \Big/ _{1- \bar{d}^1}$$

$$\mu = \bar{\mu} + \bar{e}^1 \lambda$$

$$\pi = \bar{\pi} - \frac{\bar{g}}{1- \bar{d}^1} \beta_{-2}$$

$$d = -\bar{d} \Big/ _{(1- \bar{d}^1)}$$

To initiate cycle 1 the "information matrix" is now:

$$\left[\begin{array}{ccc|cccc|c} 1 & 0 & 0 & 1 & 1 & \cdots\cdots & 1 & 0 \\ 0 & 1 & 0 & c_1 & c_2 & \cdots\cdots & c_m & 0 \\ 0 & 0 & 1 & 1 & 1 & \cdots\cdots & 1 & 0 \\ \hline 0 & 0 & 0 & 1- b_1 & -b_1 & \cdots\cdots & -b_1 & b_1 \\ 0 & 0 & 0 & -b_2 & 1- b_2 & \cdots\cdots & -b_2 & b_2 \\ \vdots & & & & \vdots & & & \vdots \\ 0 & 0 & 0 & -b_m & -b_m & \cdots\cdots & 1- b_m & b_m \end{array}\right]$$

In inverse form this becomes:

Cycle 1:

$$
\left[\begin{array}{ccc|ccccc|c}
1 & 0 & 0 & -\frac{1}{1-\bar{b}} & -\frac{1}{1-\bar{b}} & \cdots & -\frac{1}{1-\bar{b}} & -\frac{\bar{b}}{1-\bar{b}} \\
0 & 1 & 0 & -(c_1+\frac{cb}{1-\bar{b}}) & -(c_2+\frac{cb}{1-\bar{b}}) & \cdots & -(c_m+\frac{cb}{1-\bar{b}}) & -\frac{cb}{1-\bar{b}} \\
0 & 0 & 1 & -\frac{1}{1-\bar{b}} & -\frac{1}{1-\bar{b}} & \cdots & -\frac{1}{1-\bar{b}} & -\frac{\bar{b}}{1-\bar{b}} \\
\hline
0 & 0 & 0 & 1+\frac{b_1}{1-\bar{b}} & \frac{b_1}{1-\bar{b}} & \cdots & \frac{b_1}{1-\bar{b}} & \frac{b_1}{1-\bar{b}} \\
0 & 0 & 0 & \frac{b_2}{1-\bar{b}} & 1+\frac{b_2}{1-\bar{b}} & \cdots & \frac{b_2}{1-\bar{b}} & \frac{b_2}{1-\bar{b}} \\
 & \vdots & & & & & & \\
0 & 0 & 0 & \frac{b_m}{1-\bar{b}} & \frac{b_m}{1-\bar{b}} & \cdots\ 1+ & \frac{b_m}{1-\bar{b}} & \frac{b_m}{1-\bar{b}}
\end{array}\right]
$$

4. IBM 704 Program

The algorithm has been coded for IBM 704. A brief description of the program and results will be given in this section. The code was written in the FORTRAN language. It consists of the SHARE FORTRAN code, RSM1, for linear programming with some changes and deletions and the addition of the following eight subroutines:

A1 – compute c^{-p}, y^p

A2 – compute solution $x = \sum \gamma_1^p x_1$

A4 – Price out α s that dropped out of the basis

AXXJ – compute $\alpha^p = Ax^p$

BOS – main routine

CYC – do one iteration

NEW – start phase one

UNL – unload data or restart

RSM1 performs the revised simplex method with the explicit form of the inverse. Since the algorithm for the chemical equilibrium code requires the computation of n exponential functions for each α^p generated, the inverse is kept in double precision form and the pivoting operation is done in double precision to preserve accuracy.

One technique used in step I of the iterative procedure for preserving accuracy is to compute quantities y_j', $\bar{y}'$ instead of y_j, $\bar{y}$. Letting $j = *$ be defined by $y_* = \text{Max } y_j$, we set $y_j' = y_j/y_*$, $\bar{y}' = \sum y_j'$, $j = 1,\ldots,n$. Thus

$y'_* = 1$, y'_j is between 0 and 1, $1 < \bar{y}' \leq n$ and $\ell n\, \bar{y} = \ell n\, \bar{y}' - \bar{c}_*$. Letting $\bar{c}_* = -\ell n\, y_*$, then $\ell n\, y_j = \ell n\, y'_j - \bar{c}_*$ and $\ell n\, \bar{y} = \ell n\, \bar{y}' - \bar{c}_*$.

Instead of creating, at each iteration, a new column α^p as outlined in section 2, it is possible to enter into the basis an α^k that dropped out of the basis of some previous iteration k. This is the case if

$$G(u^k) - \pi_1^o \alpha^k < 0. \tag{24}$$

Therefore, each generated α is saved and, before deciding to generate a new α on a given iteration, (24) is computed for all old α s. If none price out negatively, step I is used for generating a new α. The X vector corresponding to each α is also saved to be used, if necessary, in the computation of the final solution in step III.

As shown in [5] the phase one procedure, where $\sum_{i=1}^{m} x_{n+i}$ is minimized, reduces to the standard simplex method. Thus, at each iteration in phase one, the new α column brought into the basis is the original column a_{ij}, i=1,...,m, which has the most negative reduced cost, $\bar{c}_j$. The X vector corresponding to this α is (0, 0,...,1,...,0), where the one appears in column j.

The program was written to handle chemical equilibrium problems with partitions (chemical phases). The twelve equation system representing the external respiratory system is such a problem (see [6]). The thirty columns are divided into three partitions and each iteration involves generating, from

one of the partitions, an α column for the basis. In general, a generated α comes from one of the K partitions, where the partitions are selected in cyclic order. The problem terminates when, for K consecutive iterations, $\bar{y}_i = 1$ in each partition i. It may be that it is not possible, using the columns of a selected partition, to generate an α for the basis with negative γ. In such a case the partition is never looked at again.

If each original column j is made a partition, then $x_j = \bar{x}$ in each partition, and the logarithm term $(\ln x_j/\bar{x})$ drops out of the objective function. Thus the problem becomes a standard linear programming one.

The program allows for problems up to dimensions 50 x 200. The 12 x 30 problem took 230 iterations (about 30 minutes on the 704) to reach a final solution. The number of iterations depends, of course, on the various tolerances used in the code. The most critical tolerance is that used for testing $\bar{y} = 1$. In the above example it was 10^{-5}.

The code is available from The RAND Corporation.

REFERENCES

1. Frank, M., and P. Wolfe, "An Algorithm for Quadratic Programming," Naval Logistics Quarterly, vol. 3, Nos. 1 and 2, March and June 1956.

2. Dantzig, G. B. and P. Wolfe, "A Decomposition Principle for Linear Programs," The RAND Corporation, P-1544, November 10, 1958. Revised December 10, 1959.

3. White, W. B., S. M. Johnson and G. B. Dantzig, "Chemical Equilibrium in Complex Mixtures," The Journal of Chemical Physics, vol. 28, No. 5, pp. 751-755, May 1958.

4. Dantzig, G. B., S. Johnson, and W. B. White, "A Linear Programming Approach to the Chemical Equilibrium Problem," Management Science, vol. 5, No. 1, October 1958.

5. Dantzig, G. B., "General Convex Objective Forms," The RAND Corporation, P-1664, April 9, 1959.

6. Dantzig, G. B., and DeHaven, J. C., "Contributions Toward a Mathematical Model of the Human System — I: The Respiratory Subsystem," The RAND Corporation, RM-2519, September 28, 1959.