An Optimization Algorithm for Structured Design Systems

Structured design systems are systems which may be represented by generally nonlinear equality and inequality constraints, each of which contains few of the variables in the system. Assuming the existence of an automatic capability to derive and modify as needed effective solution procedures for sets of structured equality constraints, a two-step strategy for optimizing design systems is presented in the form of two algorithms. An earlier paper presented an algorithm for locating a first feasible point, and this paper presents a companion algorithm based on the strategy known as restriction to optimize the system once feasible.

This companion algorithm divides the set of inequality constraints into three sets: the set of those currently being held as equality constraints, the set currently released but being used as search coordinates, and the set of all remaining constraints which are not currently part of the problem. Solution procedures are modified as inequality constraints and are moved from one set to another. Added constraints in the set being held tend to aid the optimization process by reducing the dimension of search space for what is usually a marginal added burden in solving an enlarged set of (structured) equations.

The strategy has proved remarkably effective on three relatively simple examples, including a nonlinear 31 constraint, 10 variable alkylation problem.

ARTHUR W. WESTERBERG and CHARLES J. DEBROSSE

Department of Chemical Engineering
University of Florida
Gainesville, Florida 32601

SCOPE

The optimization literature abounds with methods for solving constrained optimization problems. For linear systems with a linear objective function, the simplex algorithm of linear programming is remarkable in its effectiveness at reaching the optimum in very few steps. Typically the number of steps, or point evaluations, required is only one to two times the number of constraints for the problem.

The solution of nonlinear constrained optimization problems is clearly another story however. In the mathematical programming literature, many algorithms use local linearization and the simplex algorithm to approach the optimum of a nonlinear function. Another general approach is to recast the problem into an unconstrained problem by appending penalty functions which penalize the objective function whenever a constraint is about to be or is violated. A third approach is for the optimization algorithm itself to detect constraint violations and to project the search direction back into the feasible region. Several pattern search algorithms exist which simply exclude pattern points that violate constraints.

The algorithm proposed here belongs to a class known as restriction and uses a quite different approach from any of these methods. This particular algorithm is distinct in that it assumes the existence of an automatic capability to derive solution procedures for structured sets of equations such as occur in design problems. One such system is GENDER, a set of computer aids for design being developed by one of the authors. This capability is assumed to be available as the problem solution is being found so the equation sets can be automatically modified by additions and deletions of constraints.

Given this capability, the equality constraints for an optimization problem are always satisfied. Any inequality constraints which would be violated while approaching the optimum are included with the equality constraints when found and the solution procedure modified. Inequality constraints thus reduce the dimension of the search space when included as equalities. The effectiveness of such an approach is significantly influenced by the nature and structure of the equality and inequality constraints defining the problem. The essence of the approach is to speed the solution of a problem by automatically using its particular structure in the development of the solution strategy.

CONCLUSIONS AND SIGNIFICANCE

The paper presents and illustrates a minimum seeking algorithm for nonlinear, constrained problems of the type encountered in process calculations. The approach proves extremely effective as shown for three relatively simple examples. In each case, the number of problem point evaluations or steps to start from an arbitrary infeasible initial point and reach the problem optimum was on the order of the number of inequality constraints in the problem. In particular, a 2 variable, 4 inequality constraint problem was optimized in six total steps; a four variable, seven constraint problem was optimized in seven total steps; and a ten variable, 28 inequality constraint, three equality con-

straint problem was optimized in a total of 23 steps.

The algorithm stresses repeated modifications to the solution procedure with a strategic choice made for the problem decision variables as the steps are taken. These modifications are equivalent here to reprogramming the problem solution.

The paper also presents the two commonly used methods of evaluating Kuhn-Tucker multipliers, stresses their physical meaning when one is at an arbitrary point away from the problem optimum, and shows how to use them to decide which inequality constraints should be binding and which released at each step as one proceeds to the optimum. The two methods are labeled the constrained derivative method and the least squares or projection method. The advantages of the former method when one uses numerical estimates are made obvious as the multipliers are readily evaluated and the problem of partitioning the constraints becomes trivial. The paper corrects deficiencies which have appeared in earlier algorithms to solve the partitioning problem using the least squares method.

In general, the optimal design of a process system involves minimizing some objective function subject to both equality and inequality constraints. The equality constraints consist of the mathematical representation of the processes occurring in the various units of the system, the interconnections between units, and the design specifications. The inequality constraints represent limitations imposed on such variables as temperature and pressure.

A mathematical statement of the generalized design problem is

$$\underset{x \in E^n}{\text{Min}} \quad F(x) \tag{P1}$$

subject to

$$g_i(x) \leq 0, \quad i = 1, 2, \ldots, n_I$$

$$g_j(x) = 0, \quad j = n_I + 1, \ldots, n_I + n_J$$

where the functions F and g are in general nonlinear.

In a previous paper by the authors (1973), we developed an algorithm which automatically finds a feasible point, if one exists, for Problem (P1). The algorithm assumes that the equations are highly structured, that is, the equations are sparse and highly precedence orderable.

Once a feasible point has been obtained, the major task of minimizing the objective function F remains. Many different approaches to solving a constrained optimization problem are possible. We shall present a minimum-seeking optimization algorithm designed specifically for structured design systems. As we shall see later, the algorithm looks promising in the examples investigated so far. However, additional computational experience with the algorithm and variations thereof is needed to gauge the effectiveness of the algorithm vis-a-vis its competitors.

The algorithm employs a type of solution strategy which Geoffrion (1970) calls restriction. If at an optimal solution the number of tight inequality constraints is nearly as large as the degrees of freedom in the problem, as is the case for most design systems, restriction is a very useful strategy.

In the paper we shall first introduce a vector notation useful for the presentation of this material. Then, after introducing slack variables, we shall present the essential features of the algorithm adopted for solving (P1). Following the algorithm will be a more detailed explanation of each step in it, and finally we present some examples.

NOTATION AND THE INTRODUCTION OF SLACK VARIABLES

To facilitate the presentation of the material in this paper, we shall adopt a special convention using index sets for referring to vector variables. Often we shall need to refer to a vector whose elements are a subset of the elements of another vector. Illustrating with an example, the vector w_M will be a vector of n_M components of the vector w. M is an index set defined as

$$M = \{i | w_i \text{ is a component of } w_M\}$$

After defining the index sets

$$I = \{i \mid i = 1, 2, \ldots, n_I\}$$

and

$$J = \{j \mid j = n_I + 1, \ldots, n_I + n_J\}$$

and also introducing slack variables s_I, problem (P1) can be restated in an equivalent form.

$$\underset{x,\, s_I \in E^{n+n_I}}{\text{Min}} \quad F(x) \tag{P2}$$

subject to

$$g_I(x) + s_I = 0$$

$$g_J(x) = 0$$

$$s_I \geq 0$$

THE ALGORITHM

In this section we state the essential features of the algorithm, leaving the detailed description to the sections following. The algorithm has four major steps.

I. Locate a first feasible point, $x(0)$ to (P1) or (P2) using the algorithm in the companion paper to this (DeBrosse and Westerberg, 1973).

II. Partition index set I into 3 index sets at the current point $x(0)$

$$A = \{i \mid g_i(x(0)) < 0, \quad i \in I\}$$

$$B = \{i \mid g_i(x(0)) = 0, \quad i \in I, \text{ and } g_i(x(0)) \text{ should be released}\}$$

$$C = \{i \mid g_i(x(0)) = 0, \quad i \in I, \text{ and } g_i(x(0)) \text{ should remain tight}\}$$

Thus A is the set of n_A indices for all inequality constraints for which $x(0)$ is an interior point. B is the set n_B indices for all inequality constraints which are tight but should be released, and C is the remaining set of n_C indices for those constraints which are tight and should remain tight. For convenience we shall also define set $D = B \cup C$, the set of indices for all the tight constraints at $x(0)$.

III. Derive a solution procedure for the restricted problem

$$\underset{x,\, s_I}{\text{Min}} \quad F(x)$$

subject to

$$\begin{aligned} g_I(x) + s_I &= 0 \\ g_J(x) &= 0 \\ s_C &= 0 \end{aligned} \qquad \text{(P3)}$$

IV. Execute the solution procedure derived in step III, obtaining the sequence of points $x(0), x(1), \ldots x(k)$ with the property

$$F(x(p+1)) < F(x(p)), \quad p = 0, 1, \ldots, k-1$$

until one of the following occurs.

a. A variable s_j, $j \notin C$, is driven less than zero.
b. Progress toward a minimum becomes too slow. Progress can be measured by a test of the form

$$F(x(k-1)) - F(x(k)) < \epsilon_1$$

while

$$||x(k) - x(k-1)|| > \epsilon_2(\epsilon_3 + ||x(k)||)$$

where ϵ_1, ϵ_2 and ϵ_3 are appropriately chosen constants.
c. An apparent minimum has been located for the restricted problem.

In case a, transfer index j from either set A or B to set C and repeat from step III.

In case b, attempt to repartition the index set I by repeating step II with $x(0) = x(k)$. If C is unchanged, accept the slow progress and continue to try to locate the minimum.

In case c, repeat step II with $x(0) = x(k)$. If C is unchanged, an apparent minimum has been found to (P1); exit.

PARTITIONING INDEX SET I

To partition the index set I, one can use results suggested by the Kuhn-Tucker (1951) conditions for a local optimum, which can be stated briefly as follows.

Kuhn-Tucker Conditions: Write the Lagrange function for problem (P1):

$$L(x, \lambda) = F(x) - \lambda^T g(x) \qquad (1)$$

Then the Kuhn-Tucker conditions for a minimum are

$$\nabla_x L = \nabla_x F - \nabla_x g\, \lambda = 0 \qquad (2)$$

For $i \in I$

$$g_i(x)\lambda_i = 0 \qquad (3)$$

$$\lambda_i \leq 0 \qquad (4)$$

$$g_i(x) \leq 0 \qquad (5)$$

For $i \in J$

$$g_i(x) = 0 \qquad (6)$$

(It is possible for a point to be a local solution to problem (P1), and yet it will not satisfy the Kuhn-Tucker conditions. Usually one assumes, as we shall do here, that the candidate solution points satisfy a constraint qualification and, under this qualification, the Kuhn-Tucker conditions hold if the point is a local solution. Gould and Tolle (1972) discuss the geometry of constraint qualifications and present weaker criteria than the Kuhn-Tucker conditions for ascertaining if a point is a local optimum.)

When one is away from the minimum, two common and geometrically meaningful options can be used to evaluate the multipliers λ, which can then be used for partitioning set I. In both cases only conditions (2) and (3) are used. Condition (3) requires that

$$\lambda_A = 0 \qquad (7)$$

thus Equations (2) involve only the n_K constraints $g_K(x)$, where here $K = B \cup C \cup J$. (This is not a definition of K as the index sets in K will change throughout the remainder of the paper.)

Constrained Derivatives (CD): Partition the x variables into a set of n_K dependent variables y and a set of $n - n_K$ independent or decision variables z. Choose $\lambda_K{}^{CD}$ to satisfy the following linear equations

$$\nabla_y L = \nabla_y F - \nabla_y g_K \lambda_K{}^{CD} = 0 \quad \text{at} \quad x = x(0) \qquad (8)$$

Using the resulting values for $\lambda_K{}^{CD}$, one can easily show that $\nabla_z L$ are the constrained derivatives (Wilde and Beightler, 1967) of F with respect to z, where the perturbations in y are dependent on the perturbations in z through the relationships

$$dg_K = \nabla_y g_K{}^T dy + \nabla_z g_K{}^T dz = 0 \qquad (9)$$

Note the Jacobian $\nabla_y g_K \equiv \dfrac{\partial g_K{}^T}{\partial y}$ must be nonsingular.

If no matrix of rank n_K exists for any choice of dependent variables and provided we are not plagued with a constraint qualification problem, one (or more) of the constraints in g_K is redundant at the point $x(0)$ and must be deleted.

Least Squares (LS) *or Projection:* Find λ to minimize the sum of squares norm of $\nabla_x L$, $\nabla_x L^T \nabla_x L$. This value of λ satisfies the linear equations [at $x = x(0)$]

$$\nabla_x g_K{}^T(\nabla_x F - \nabla_x g_K \lambda_K{}^{LS}) = 0 \qquad (10)$$

Because $\lambda_K{}^{LS}$ is a least squares estimate, the resulting vector $\nabla_x L$ is the orthogonal projection of $\nabla_x F$ onto the tangent hyperplane of the constraints $g_K = 0$ at the point $x(0)$ (Rosen (1960)). Again the coefficient matrix $\nabla_x g_K{}^T \nabla_x g_K$ must be nonsingular.

The partitioning Criterion. The sign of the Kuhn-Tucker multiplier provides us with the primary tool to partition the index set I. A physical interpretation of $\lambda_j{}^{CD}$, a particular component of $\lambda_K{}^{CD}$, is possible. $\lambda_j{}^{CD}$ is the ratio of the change in F, the objective function, to an infinitesimal change in the value of the constraint g_j. The specific direction of movement permitted in the x space when making the infinitesimal change in g_j is that resulting from staying in the tangent hyperplane to the line defined by holding all the other constraints at $x(0)$ at zero and all the decision variables unchanged. This surface is a line since all but one degree of freedom, to perturb g_j, is fixed.

$\lambda_j{}^{LS}$ has a similar physical interpretation. However for $\lambda_j{}^{LS}$, no decision variables are identified so a different second set of restraints provides us with the direction of movement. For $\lambda_j{}^{LS}$, movement is in the unique direction resulting from projecting ∇g_j into the tangent hyperplane to the surface defined by holding all other constraints in g_K at zero.

Thus $\lambda_j < 0$ implies that, while holding all other constraints, an infinitesimal move in the permitted direction to the interior of the feasible region of g_j will result in an increase in F; constraint g_j should be held. Conversely for $\lambda_j > 0$, g_j should be released.

Criteria for Using $\lambda_K{}^{LS}$. To partition the set I, Farris and Law (1969) suggested a trial-and-error procedure in which at each step the set D is tentatively partitioned into sets B and C. For each $j \in B$, a different index set K is defined as $K = J \cup C \cup j$, and, for each set K, $\lambda_K{}^{LS}$ is evaluated. The set D is successfully partitioned into sets B and C, if, for each set K, the resulting $\lambda_j{}^{LS} > 0$ while $\lambda_C{}^{LS} < 0$. If the test fails for any j, a new partitioning of

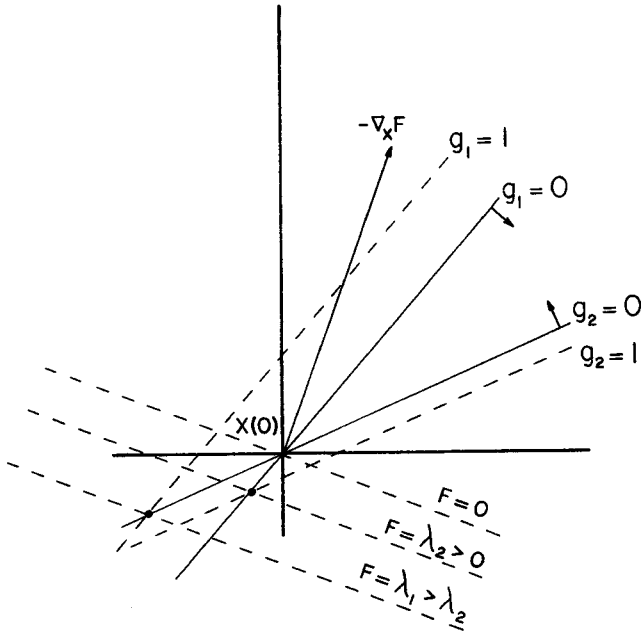

Fig. 1. Two-dimensional linear system with infeasible gradient. Illustrates physical interpretation of λ_1 and λ_2.

D is proposed and the procedure repeated.

Farris and Law's procedure proved to be disappointing when applied to a ten-dimensional system that we shall discuss later. Each time a constraint in the set g_B was added to the basis set g_K, the signs of several components of $\lambda_C{}^{LS}$ changed sign, immediately invalidating the proposed sets. In fact, there appeared to be no assurance the procedure would even converge on satisfactory sets g_B and g_C.

Murtagh and Sargent (1968) propose the following algorithm to partition the inequality constraints:

i) Normalize the gradients of all constraints.

ii) Place all tight inequality constraints initially into the set g_C; that is, let $C = D$ and $B = \phi$.

iii) Let $K = J \cup C$ and evaluate $\lambda_K{}^{LS}$.

iv) If any of the components of $\lambda_K{}^{LS}$ corresponding to the constraints in set g_C are positive, move the index of the constraint corresponding to the largest such component from the set C to B. Go to (iii). Otherwise, stop.

Murtagh and Sargent's algorithm has one defect—it is not certain that the set C determined by the algorithm will be such that the negative of the final projected gradient $-\nabla_x L_C$ will point into the feasible region of every constraint in g_B. It may happen that dropping a constraint index from the set C will have the effect that $-\nabla_x L_C$ points into the infeasible region of a constraint previously dropped from g_C. This difficulty is illustrated by the two-dimensional example in Figure 1. Letting $K = \{1, 2\}$, we evaluate $\lambda_K{}^{LS}$ for this system and find that both components $\lambda_1{}^{LS}$ and $\lambda_2{}^{LS}$ are positive. (One can use the geometric interpretation just stated above for $\lambda_j{}^{LS}$ to see this.) Suppose that $\lambda_1{}^{LS} > \lambda_2{}^{LS}$, and we therefore drop 1 from the set C, giving us the negative of the projected gradient $-\nabla_x L_2$. (If the gradients are normalized, it is obvious that $\lambda_1{}^{LS} < \lambda_2{}^{LS}$ in this two-dimensional example. We are attempting to show what might happen in a higher dimensional space by means of a two-dimensional example.) Letting $K = \{2\}$, we then compute $\lambda_K{}^{LS}$ and find that it is also positive. We therefore drop 2 from the set C, and $C = \phi$. The negative of the projected gradient is now equal to $-\nabla_x F$, which points into the infeasible region of g_1.

The criterion we shall adopt for partitioning the inequality constraints using $\lambda_K{}^{LS}$ will be similar to Murtagh and Sargent's. Each time we drop an inequality constraint g_j from the set g_C, we shall include a check to see whether the new gradient $-\nabla_x L_K$ still points into the feasible region of the constraints previously dropped from g_C. We do not have to test g_j obviously. If it does, we proceed as in Murtagh and Sargent's algorithm; if it does not, we put j back into set C and stop. We could at this point choose to drop any other constraint index $i \in C$ with a positive component in $\lambda_K{}^{LS}$ (after j has been put back) and attempt again to reduce the size of C. Either approach can in principle lead to the solution. One can always release at least one constraint if any should be released.

The following proposition states the conditions under which the negative of the projected gradient $-\nabla_x L_K$ is feasible with respect to all constraints in the set g_B.

Proposition 1. The projected gradient $-\nabla_x L_K$ is feasible with respect to all constraints in the set g_B if and only if for all $j \in B$ $\delta_j \equiv \nabla_x L_K \cdot \nabla_x g_j \geq 0$.

Proof. $-\nabla_x L_K$ is the projection of interest since we are minimizing F. Further, since we are writing the constraints as $g_j \leq 0$, $-\nabla_x L_K$ is feasible with respect to g_j if and only if $-\nabla_x L_K$ and $-\nabla_x g_j$ are less than or equal to 90 degrees apart, that is, if and only if $\delta_j \geq 0$.

In Figure 1, for example, $-\nabla_x L_2$ is feasible with respect to g_1 because $\nabla_x L_2 \cdot \nabla_x g_1 > 0$. However, $-\nabla_x F$ itself is not feasible with respect to g_1 because $\nabla_x F \cdot \nabla_x g_1 < 0$.

Criterion Using $\lambda_K{}^{CD}$. Abadie and co-workers (1969, 1970) have used a partitioning criterion based on $\lambda_K{}^{CD}$. The indices for all tight constraints are first put into D. If one then partitions the variables so that the variables s_D are included as decision variables, the components of $\lambda_K{}^{CD}$ corresponding to constraints g_D obey the relationship

$$\lambda_D{}^{CD} = -\nabla_{s_D} L \tag{11}$$

that is, these multipliers and the constrained derivatives with respect to corresponding slack variables are the negative of each other. Partitioning is straightforward. Evaluate all $\lambda_D{}^{CD}$ and put the $\lambda_i{}^{CD} < 0$, $i \in D$, in set C and the rest into B. Note one does not have to worry about the effect of releasing more than one constraint on the signs of the other $\lambda_i{}^{CD}$. This result occurs because the s_i, being decision variables, are readily manipulated independently.

DERIVING A SOLUTION PROCEDURE FOR (P3)

Problem (P3) is an equality only constrained optimization problem. The strategy proposed here for such a problem is to use a solution procedure that partitions the variables x and s_I into a set of $n_I + n_J + n_C$ dependent variables u and $n - n_J - n_C$ independent or decision variables v. An unconstrained optimization algorithm is used to adjust only v while the equality constraints are satisfied at each step to give values to u and then $F(x)$. For this approach one requires in step 3 a method to derive a solution procedure for a set of algebraic equations.

Thus a principal requirement for the implementation of both the optimization algorithm of this paper and the feasible-point algorithm of the authors' previous paper (1973) is the ability to solve efficiently a structured set of $m \leq n$ equations in n variables. Rather than discuss in detail available methods for solving a structured set of equations, we shall refer the reader to pertinent sources. A survey of the available methods for determining a solution procedure for structured systems is given by DeBrosse (1972). A partial list of pertinent references includes the

following: Sargent and Westerberg (1964), Lee and Rudd (1966), Rudd and Watson (1968), Ledet (1968), Christensen and Rudd (1969), Christensen (1970), Westerberg and Edie (1971), Edie and Westerberg (1971), Kavorkian and Snoek (1972), Barkley and Motard (1972), Westerberg (1972), Willoughby (1968), Rose and Willoughby (1972), Reid (1972), Director (1972), Erisman (1972). The references from Willoughby on describe procedures based on sparse matrix methods whereas the others stress tearing methods. Several of the references are from the NATO Institute on Decomposition held in Cambridge, England, in July, 1972. These will appear in the meeting proceedings edited by Himmelblau (1973).

In step III, a further refinement of the problem statement can reduce the number of modified solution procedures required and reduce potential problems while taking a step with the unconstrained optimization algorithm. This refinement is to drop the equations $s_C = 0$ from the set of equality constraints and then to require the solution procedure to have variables s_D (variables s_B and s_C) as decision variables. The dropped equations are readily enforced by externally setting the decision variables s_C to zero.

If one now discovers s_j, $j \epsilon B$, becoming negative in step 4 of the algorithm, a new solution procedure is not required. The index j is moved to set C, and s_j, then being a decision variable, is held at zero. It is also simple to move an index from set C to set B. If it is convenient in the solution procedure, any of the s_A variables which can be chosen as decisions would give the same advantage. Generally, however, n_A is much larger than the remaining degrees of freedom so not all the s_A variables can be chosen.

By choosing s_B to be among the decision variables, movement by the unconstrained optimization algorithm into the feasible region ($s_B \geq 0$) is easily controlled. Otherwise the unconstrained optimization technique can cause an immediate difficulty. Many optimization algorithms initiate a search for the minimum by perturbing each coordinate direction in turn. Figure 2 illustrates what can happen as a result. If we wish to move in a steepest descent direction, it is obvious that both g_1 and g_2 should be released. Then if we perturb x_1 to the point 'a' while holding x_2 constant, we immediately enter the infeasible region of g_2. Furthermore, if, as in the Powell algorithm (1964), we continue in the same direction until a minimum is encountered, we will stay infeasible for several more points. If we perturb x_2 first, we will have the same problem with g_1. Using s_1 and s_2 as coordinates overcomes this difficulty.

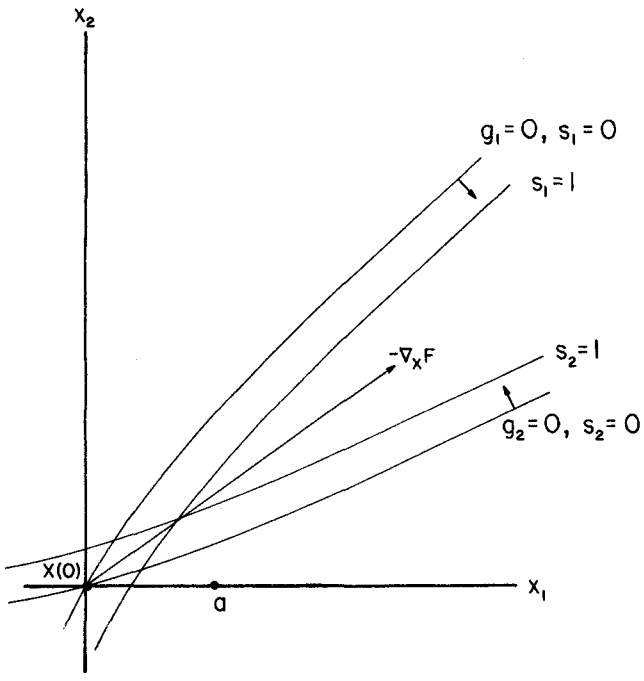

Fig. 2. Two-dimensional inequality constrained system which illustrates advantage of using slack variables as decision variables.

The last and very useful advantage to choosing variables s_D to be among the decision variables is that a numerical estimation of $\lambda_{s_D}{}^{CD}$ is straightforward because of relationship (11). By perturbing each s_i, $i \epsilon D$, one-at-a-time, one can approximate the $\lambda_{s_i}{}^{CD}$ as

$$\lambda_{s_i}{}^{CD} \approx -\frac{\Delta F}{\Delta s_i}, \qquad i \epsilon D$$

A final comment is appropriate regarding the ability to derive and execute a solution procedure for a set of equality constraints. If the constraints g_J are always satisfied, n_J variables of the original problem are effectively removed from the problem leaving a set we could call $\hat{x}$. If numerical estimates for the needed gradients are used, only approximations to $\nabla_x g_I$ and $\nabla_x F$ are then needed to estimate the Kuhn-Tucker multipliers $\lambda_K{}^{LS}$. The penalty will be the requirement that $g_J = 0$ must be satisfied for each point used in the approximations.

SELECTED APPLICATIONS OF THE OPTIMIZATION ALGORITHM

In this section we shall apply the optimization algorithm to three examples. The first two examples hopefully will enable the reader to grasp the essentials of the algorithm. The last example will furnish an indication of the effectiveness of the algorithm for structured design systems.

Example 1

Minimize

$$F \equiv 10x_1 - 25x_2 + 10x_1{}^2 + x_2{}^2 + 4x_1x_2$$

subject to

1. $-x_1 + s_1 = 0$
2. $-x_2 + s_2 = 0$
3. $x_1 + x_2 - 9 + s_3 = 0$
4. $-x_1 - 2x_2 + 10 + s_4 = 0$

$$s_1, s_2, s_3, s_4 \geq 0$$

By applying the feasible-point algorithm for structured design systems, a feasible point was obtained in two steps at $x_1 = 0$, $x_2 = 5$. We can now find the minimum of F subject to the four linear inequality constraints listed above by applying the optimization algorithm.

The figures in parentheses refer to the algorithm step being implemented.

(I) $x(0)^T = [0, 5]$, the initial feasible point.

(II) At $x(0)$, s_1 and s_4 are zero and s_2 and s_3 are greater than zero. If we use $\lambda_K{}^{LS}$ to partition, then the initially proposed index sets are $A = \{2, 3\}$, $B = \phi$, and $C = \{1, 4\}$. Index set $K = C$ since $J = \phi$ (no equality constraints).

We evaluate the Kuhn-Tucker multiplier $\lambda_C{}^{LS}$ as

$$\lambda_C{}^{LS} = (\nabla_x g_C{}^T \nabla_x g_C)^{-1} \nabla_x g_C{}^T \nabla F$$

$$= \left(\begin{bmatrix} -1 & 0 \\ -1 & -2 \end{bmatrix} \begin{bmatrix} -1 & -1 \\ 0 & -2 \end{bmatrix} \right)^{-1}$$

$$\begin{bmatrix} -1 & 0 \\ -1 & -2 \end{bmatrix}\begin{bmatrix} 10 \\ 15 \end{bmatrix} = \begin{bmatrix} -17.5 \\ 7.5 \end{bmatrix}$$

$\lambda_4^{LS} = 7.5$ is the largest nonnegative component of λ_C^{LS} so we move index 4 from C to B. Thus we have $A = \{2, 3\}$, $B = \{4\}$, and $C = \{1\}$.

We again evaluate λ_C^{LS} to get

$$\lambda_C^{LS} = \left(\begin{bmatrix} -1 & 0 \end{bmatrix}\begin{bmatrix} -1 \\ 0 \end{bmatrix}\right)^{-1}\begin{bmatrix} -1 & 0 \end{bmatrix}\begin{bmatrix} 10 \\ -15 \end{bmatrix}$$

$$= -10$$

Since $\lambda_C^{LS} < 0$ and B contains only the constraint just dropped, we have successfully partitioned index set I.

(III) The solution procedure we derive must solve the following restricted problem.

$$\text{Min} \quad 10x_1 - 25x_2 + 10x_1^2 + x_2^2 + 4x_1x_2$$

subject to

$$\begin{aligned} -x_1 + s_1 &= 0 \\ -x_2 + s_2 &= 0 \\ x_1 + x_2 - 9 + s_3 &= 0 \\ -x_1 - 2x_2 + 10 + s_4 &= 0 \\ s_1 &= 0 \end{aligned}$$

We require s_1 and s_4 to be decision variables and could evolve the following solution procedure for the equality constraints.

Decision variables $s_1 = 0$, s_4

$$\begin{aligned} x_1 &= s_1 \\ x_2 &= \frac{1}{2}(-x_1 + 10 + s_4) \qquad \text{(SP1)} \\ s_2 &= x_2 \\ s_3 &= 9 - (x_1 + x_2) \end{aligned}$$

We augment this solution procedure with

$$F = 10x_1 - 25x_2 + 10x_1^2 + x_2^2 + 4x_1x_2$$

and use these equations as input to an unconstrained optimization problem which is solving, in effect, the one dimensional problem

$$\text{Min} \quad F(s_4)$$

$$s_4$$

(IV) In solving we take the following steps

Step	s_1	s_4	x_1	x_2	s_2	s_3	F
0	0	0	0	5	5	5	−100
1	0	1	0	5.5	5.5	3.5	−107.25
2	0	4	0	7	7	2	−126
3	0	10	0	10	10	−1	−150

Case a now occurs; we find $s_3 < 0$. We redefine the index sets

$$A = \{2\}, \quad B = \{4\}, \quad C = \{1, 3\}$$

The new solution procedure found is

Decision variables $s_1 = s_3 = 0$

$$\begin{aligned} x_1 &= s_1 \\ x_2 &= 9 - x_1 - s_3 \\ s_2 &= x_2 \\ s_4 &= x_1 + 2x_2 - 10 \end{aligned}$$

(IV) In solving we find

Step	s_1	s_3	x_1	x_2	s_2	s_4	F
4	0	0	0	9	9	8	−144

No further steps can be taken since s_1 and s_3 are both required to be zero. Case c applies.

(I) Recalculating λ_C^{LS} gives $(x(0)^T = [0, 9])$

$$\lambda_C^{LS} = \begin{bmatrix} \lambda_1^{LS} \\ \lambda_3^{LS} \end{bmatrix} = \begin{bmatrix} -33 \\ -7 \end{bmatrix}$$

Both multipliers are negative; an apparent minimum is found for the original problem. (We note the projected gradient is

$$\nabla_x L_C = \begin{bmatrix} 26 \\ -7 \end{bmatrix} - \begin{bmatrix} -1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} -33 \\ -7 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

at this solution point.) Exit with $x_1^* = 0$, $x_2^* = 9$.

If we choose to use λ_C^{CD} rather than λ_C^{LS}, we would modify the steps taken somewhat. The more interesting case is to use numerical estimates for λ_C^{CD}. The algorithm steps would occur as follows.

(I) $x^T(0) = [0, 5]$, the initial feasible point.

(II) Since $D = \{1, 4\}$, derive the solution procedure where these are the decisions. This procedure probably exists as the final one used to find the initial feasible point. We already have it as (SP1). Perturb s_1 and then s_4 in turn to get

Step	s_1	s_4	x_1	x_2	s_2	s_3	F
0	0	0	0	5	5	5	−100
1	1	0	1	4.5	4.5	4.5	− 54.25
2	0	1	0	5.5	5.5	3.5	−107.25

Then

$$\lambda_{s_D}^{CD} \approx \begin{bmatrix} -45.75 \\ 7.25 \end{bmatrix}$$

indicating s_1 should be held and s_4 released. (The choice of step size here is obviously too large.) The remainder of the algorithm proceeds similarly.

Example 2

Minimize $F \equiv 24.55x_1 + 26.75x_2 + 39.00x_3 + 40.50x_4$ subject to

1. $-12.0x_1 - 11.9x_2 - 41.8x_3 - 52.1x_4 + 1.645\,[0.28x_1^2 + 0.19x_2^2 + 20.5x_3^2 + 0.62x_4^2]^{1/2} + 21 + s_1 = 0$
2. $-2.3x_1 - 5.6x_2 - 11.1x_3 - 1.3x_4 + 5 + s_2 = 0$
3. $-x_1 - x_2 - x_3 - x_4 + 1 + s_3 = 0$
4. $-x_1 + s_4 = 0$
5. $-x_2 + s_5 = 0$
6. $-x_3 + s_6 = 0$
7. $-x_4 + s_7 = 0$

$$s_1, s_2, s_3, s_4, s_5, s_6 \geqq 0\,.$$

Arbitrarily starting with $x = 0$, a feasible point was found at $x_1 = 0.595$, $x_2 = 0$, $x_3 = 0.504$, $x_4 = 0$ via the

feasible point algorithm for structured design systems in three steps. The optimal solution of $x^* = [0.635, 0, 0.313, 0.052]$ was located in four additional steps using the algorithm in this paper. Inequality constraints 1, 2, 3, and 5 are tight at the final solution point. This problem appears on page 100 of Bracken and McCormick (1968) and was previously solved via different methods by Fiacco and McCormick (1964) and by van de Panne and Popp (1963).

Example 3

Minimize

$F \equiv -0.063x_4x_7 + 5.04x_1 + 0.035x_2 + 10.00x_3 + 3.36x_5$

subject to

1. $-x_1 + s_1 = 0$
2. $-x_2 + s_2 = 0$
3. $-x_3 + s_3 = 0$
4. $-x_4 + s_4 = 0$
5. $-x_5 + s_5 = 0$
6. $-x_6 + 85 + s_6 = 0$
7. $-x_7 + 90 + s_7 = 0$
8. $-x_8 + 3 + s_8 = 0$
9. $-x_9 + 1.2 + s_9 = 0$
10. $-x_{10} + 145 + s_{10} = 0$
11. $x_1 - 2000 + s_{11} = 0$
12. $x_2 - 16000 + s_{12} = 0$
13. $x_3 - 120 + s_{13} = 0$
14. $x_4 - 5000 + s_{14} = 0$
15. $x_5 - 2000 + s_{15} = 0$
16. $x_6 - 93 + s_{16} = 0$
17. $x_7 - 95 + s_{17} = 0$
18. $x_8 - 12 + s_{18} = 0$
19. $x_9 - 4 + s_{19} = 0$
20. $x_{10} - 162 + s_{20} = 0$
21. $-[x_1(1.12 + 0.13167x_8 - 0.00667x_8^2)] + 0.99x_4 + s_{21} = 0$
22. $[x_1(1.12 + 0.13167x_8 - 0.00667x_8^2)] - 1.01x_4 + s_{22} = 0$
23. $-[86.35 + 1.098x_8 - 0.038x_2^2 + 0.325(x_6 - 89)] + 0.99x_7 + s_{23} = 0$
24. $[86.35 + 1.098x_8 - 0.038x_8^2 + 0.325(x_6 - 89)] - 1.01x_7 + s_{24} = 0$
25. $-[35.82 - 0.222x_{10}] + 0.99x_9 + s_{25} = 0$
26. $[35.82 - 0.222x_{10}] - 1.01x_9 + s_{26} = 0$
27. $-[-133 + 3x_7] + 0.99x_{10} + s_{27} = 0$
28. $[-133 + 3x_7] - 1.01x_{10} + s_{28} = 0$
29. $(x_2 + x_5)/x_1 - x_8 = 0$
30. $98000x_3/(x_4x_9 + 1000x_3) - x_6 = 0$
31. $1.22x_4 - x_1 - x_5 = 0$

s_1 through $s_{28} \geq 0$.

Applying the feasible point algorithm to an arbitrary starting point, a feasible point was obtained at [1468.0, 16000, 65.48, 2528.2, 1616.4, 86.07, 92.18, 12, 3.59, 145] in eight steps.

We summarize the results obtained via application of the optimization algorithm in Table 1. The minimum of F subject to the 31 constraints given above was reached in 15 steps from the first feasible point. Note the constraint sets g_B and g_C were altered eight times. Each time sets B and C are fixed the search is potentially infinite and the number of steps is a function of the unconstrained search technique used. No attempt was made here to be clever during these searches as it was found that new constraints were quickly encountered.

The problem solved in Example 3 appears in Chapter 4 of Bracken and McCormick (1968). It represents an actual alkylation process as described by Sauer et al. (1964). Bracken and McCormick obtain a slightly lower minimum $F(x^*) = -1769$. However, their optimal solution is in violation of g_{27} by 0.46 units which indicates a probable error in their coding (Bracken, 1971), whereas the optimum calculated above satisfies $g_{27} = 0$.

NOTATION

A = index set, contains indices of constraints strictly satisfied at $x(0)$
B = index set, contains indices of tight inequality constraints which should be released at $x(0)$
C = index set, contains indices of tight inequality constraints which should be released at $x(0)$
D = index set, contains indices of tight inequality constraints at $x(0)$
E^n = n-dimensional Euclidean space
F = objective function
g = constraint functions
i = simple subscript index
I = index set of all inequality constraints for problem (P1) or (P2)
j = simple subscript index
J = index set of all equality constraints for problem (P1) or (P2)
k = simple subscript index
K = index set for the constraints forming the basis set g_K used when evaluating Kuhn-Tucker multipliers
L = Lagrange function
p = simple index
s = slack variable vector for problem (P2)
u = dependent variable vector
v = decision variable vector
x = variables in original problem (P1)
$\hat{x}$ = reduced set of variables if constraints g_J used to eliminate some of the original variables from (P1) or (P2)
y = dependent variable vector
z = decision variable vector

Greek Letters

δ_i = value of dot product of two gradients in Proposition 1
$\epsilon_1, \epsilon_2, \epsilon_3$ = appropriately chosen constants for measuring rate of progress in the optimization algorithm
λ^{CD} = Kuhn-Tucker and Lagrange multipliers evaluated using the method of Constrained Derivatives
λ^{LS} = Kuhn-Tucker and Lagrange multipliers evaluated using the method of Least Squares
$\cup$ = mathematical symbol for set union
∇_x = gradient operator, $\nabla_x g \equiv \left[\frac{\partial g_j}{\partial x_i}\right] \equiv \frac{\partial g^T}{\partial x}$, is a Jacobian matrix (with row index i and column index j)
ϕ = the null set

LITERATURE CITED

Abadie, J., "Application of the GRG Algorithm to Optimal Control Problems," in *Integer and Nonlinear Programming*, pp. 191-211, North Holland, Amsterdam (1970).

———., and J. Carpentier, "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints," in *Optimization*, pp. 37-47, Academic Press, New York (1969).

TABLE 1. SUMMARY OF RESULTS OF OPTIMIZATION ALGORITHM FOR EXAMPLE 3

Initial point	Final point	Constraint indices in Set B	Constraint indices in Set C	Comment
$x(0)$: [1,468.0, 16,000, 65.48, 2,528.2, 1,616.4, 96.07, 92.18, 12, 3.59 145]; $F(x(0)) = -636.87$	$x(2)$: [1,502.2, 16,000, 46.63, 2,596.9, 1,666, 83.97, 93.15, 11.76, 3.00, 147.93]; $F(x(2)) = -1,044.53$	{10, 18, 24, 26}	{12, 22, 27}	$x(2)$ infeasible. Multipliers evaluated at $x(0)$ for partitioning.
$x(2)$	$x(3)$: [1,518.4, 16,000, 46.24, 2,628.9, 1,688.9, 85, 93.65, 11.65, 2.69 149.44]; $F(x(3)) = -1,160.73$	{10, 18, 26}	{6, 12, 22, 27}	$x(3)$ infeasible. Algorithm step IIIa occurs at $x(2)$.
$x(3)$	$x(4)$: [1,518.4, 16,000, 46.24, 2,628.9, 1,688.9, 85, 93.62, 11.65, 2.69, 149.35]; $F(x(4)) = -1,155.76$	{18}	{6, 12, 22, 23, 25, 27}	Algorithm step IIIa occurs at $x(3)$.
$x(4)$	$x(6)$: [1,553.2, 16,000, 60.54, 2,697.0, 1,737.2, 86.72, 94.13, 11.42, 2.92, 148.1]; $F(x(6)) = -1,163.50$	{6, 18, 25, 27}	{12, 22, 23}	Algorithm step IIIc occurs at $x(4)$. (Multipliers evaluated for partitioning.)
$x(6)$	$x(9)$: [1,681.2, 16,000, 76.58, 2,939.3, 1,904.7, 89.42, 94.82, 10.65, 2.50, 149.96]; $F(x(9)) = -1,359.35$	{6, 18}	{12, 22, 23, 26, 28}	Algorithm step IIIa occurs at $x(6)$.
$x(9)$	$x(12)$: [1,684.4, 15,530, 48.33, 3,037.2, 2,021, 90.18, 95, 10.42, 1.38, 155.1]; $F(x(12)) = -1,870.94$	{12, 18, 22, 28}	{17, 23, 26}	$x(12)$ infeasible. Algorithm step IIIc occurs at $x(9)$. (Multipliers evaluated.)
$x(12)$	$x(14)$: [1,700.3, 1,607.2, 58.56, 3,033.0, 2,000, 90.0, 95, 10.63, 1.72, 153.53]; $F(x(14)) = -1,715.08$	{21}	{15, 17, 21, 23, 26, 27}	$x(14)$ infeasible. Algorithm step IIIc occurs at $x(12)$. (Multipliers evaluated.)
$x(14)$	$x(15)$: [1,699.5, 16,000, 58.85, 3,032.4, 2,000, 90.02, 95, 10.59, 1.72, 153.53]; $F(x(15)) = -1,714.93 = F(x^*)$	ϕ	{12, 15, 17, 21, 23, 26, 27}	Algorithm step IIIa occurs at $x(14)$.
$x(15)$				Algorithm step IIIc occurs at $x(15)$.

Barkley, R. W., and R. L. Motard, "Decomposition of Nets," NATO Inst. on Decomp., Cambridge, England (July, 1972).

Bracken, J., private communication (1971).

———., and G. P. McCormick, *Selected Applications of Nonlinear Programming*, Wiley, New York (1968).

Christensen, J. H., "The Structuring of Process Optimization," *AIChE J.*, **16**, 177 (1970).

———., and D. F. Rudd, "Structuring Design Computations," *ibid.*, **15**, 94 (1969).

DeBrosse, C. J., "Feasible-point and Optimization Algorithms for Structured Design Systems in Chemical Engineering," Ph.D. dissertation, pp. 67-84, Univ. Florida (1971).

———., and A. W. Westerberg, "A Feasible-point Algorithm for Structured Design Systems in Chemical Engineering," *AIChE J.*, **19**, 000 (1973).

Director, S. W., "A Survey of Decomposition Techniques for Analysis and Design of Electrical Networks," NATO Inst. on Decomp., Cambridge, England (1972).

Edie, F. C., and A. W. Westerberg, "Computer Aided Design, Part 3," *Chem. Eng. J.*, **2**, 114 (1971).

Erisman, A. M., "Decomposition Methods Using Sparse Matrix Techniques with Application to Certain Electrical Network Problems," NATO Inst. on Decomp., Combridge, England (July, 1972).

Farris, R. H., and V. J. Law, "Search Vector Strategy with Non-linear Equality and Inequality Constraints," paper presented at 64th National AIChE Meeting, New Orleans (1969).

Fiacco, A. V., and G. P. McCormick, "Computational Algorithm for Sequential Unconstrained Minimization Technique for Nonlinear Programming," *Management Sci.*, **10**, 601 (1964).

Geoffrion, A. M., "Elements of Large-scale Mathematical Programming," *Management Sci.*, **16**, 652 (1970).

Gould, F. J., and J. W. Tolle, "Geometry of Optimality Conditions and Constraint Qualifications," *Mathematical Programming*, **2**, 1 (1972).

Himmelblau, D. M. (ed.), Proc. of NATO Inst. on Decomp., Cambridge, England, 1972, to be published by North-Holland, Amsterdam (1973).

Kavorkian, A. K., and J. Snoek, "Decomposition in Large Scale Systems, Parts I and III," NATO Inst. on Decomposition, Cambridge, England (1972).

Kuhn, H. W., and A. W. Tucker, "Non-linear Programming," in *Proc. 2d Berkeley Symp. on Math. Statistics Probl.*, 481, Univ. of Calif. Press, Berkeley (1951).

Ledet, W. P., "Decomposition of Large-scale Systems of Equations," Ph.D. dissertation, Univ. Texas, Austin (1968).

Lee, W., and D. F. Rudd, "On the Ordering of Recycle Calculations," *AIChE J.*, **12**, 1184 (1966).

Murtagh, B. A., and R. W. H. Sargent, "A Constrained Minimization Method with Quadratic Convergence," paper presented at Inst. of Math. and Its Applications and Br. Computer Soc. Jt. Conf. on Optimization, Univ. Keele (1968).

Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives," *Computer J.*, **7**, 155 (1964).

Reid, J. K., "Sparse Matrices and Decomposition with Applications to the Solution of Algebraic and Differential Equa-

tions," NATO Inst. on Decomposition, Cambridge, England (1972).
Rose, D. J., and R. A. Willoughby (eds.), *Sparse Matrices and Their Applications*, Plenum Press, New York (1972).
Rosen, J. B., "The Gradient Projection Method for Non-linear Programming, Part I, Linear Constraints," *J. Soc. Ind. Appl. Math.*, 8, 181 (1960).
Rudd, D. F., and C. C. Watson, *Strategy of Process Engineering*, pp. 34-69, Wiley, New York (1968).
Sargent, R. W. H., and A. W. Westerberg, " 'SPEED-UP' in Chemical Engineering Design," *Trans. Instn. Chem. Engrs.*, 42, T190 (1964).
Sauer, R. N., A. R. Colville, Jr., and C. W. Burwick, *Hydrocarbon Process. Petrol. Refiner*, 43, 84 (1964).
Van de Panne, C., and W. Popp, *Management Sci.*, 9, 405 (1963).
Westerberg, A. W., and F. C. Edie, "Computer Aided Design, Parts 1 and 2," *Chem. Eng. J.*, 2, 9, 17 (1971).
Westerberg, A. W., "Decomposition Methods for Solving Steady State Process Design Problems," NATO Inst. on Decomp., Cambridge, England (1972).
Wilde, D. J., and C. S. Beightler, *Foundations of Optimization*, p. 35, Prentice-Hall, Englewood Cliffs, N. J. (1967).

Manuscript received March 9, 1972; revision received October 11, 1972; paper accepted October 16, 1972.

Restricted Diffusion in Binary Solutions

Diffusion coefficients of binary solutions can be measured accurately by observation at long times of isothermal diffusion in a vertical cell closed at the ends (restricted diffusion). The present analysis, which accounts for the effects of solvent flux and variable solution properties, demonstrates that the experiment yields a well-defined, differential diffusion coefficient, even in concentrated solutions.

Observation is effected by Rayleigh interference optics, and results for aqueous potassium chloride solutions confirm the accuracy of the method.

JOHN NEWMAN and THOMAS W. CHAPMAN

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

SCOPE

Accurate values of diffusion coefficients of binary mixtures, including their dependence on composition and temperature, are required to provide complete transport-property data (somewhat analogous to thermodynamic data like activity coefficients), to permit quantitative design of separation processes like distillation and electrochemical deposition, and to permit detailed comparison with theoretical results. Among theoretical results we have in mind, on the one hand, the treatment of simple systems where variable physical properties have been included in the analysis and, on the other hand, the theoretical prediction or correlation of transport properties on a molecular basis such as the Debye-Hückel theory.

For binary solutions of electrolytes, three methods are used to measure the diffusion coefficient: the magnetically stirred diaphragm cell, optical observation of diffusion from an initially sharp boundary, and (restricted) diffusion in a box of finite height with observation of the concentration distribution by measurement of the electrical conductivity at two positions in the box. The last method, developed by Harned beginning in 1949, appeals to the intuition because the maximum concentration difference in the system vanishes as the experiment progresses, and an absolute determination of the diffusion coefficient requires only the height of the box.

One question we raise is just what diffusion coefficient is measured and is this complicated by the reference velocity for the diffusion flux, the precise definition of transport properties for concentrated solutions, or the variation of physical properties with composition?

Harned's method is restricted to very dilute solutions of electrolytes because adequate resistance measurement with solutions of higher conductivity requires larger cell dimensions where free convection becomes a problem. The second question is how can the method be extended to concentrated solutions?

CONCLUSIONS AND SIGNIFICANCE

An analysis of the restricted diffusion process shows that the measurement yields the differential diffusion coefficient D at the final average concentration. This is the diffusion coefficient which appears in the flux Equations (13) (where Q can be set equal to zero) and is related to the fundamental diffusion coefficient $\mathcal{D}$ by Equation (15). D is equivalent to the binary diffusion coefficient $\mathcal{D}_{AB}$ used by Bird, Stewart, and Lightfoot.

The analysis with variable physical properties gives a new form, different from previous results, for the correction to the dominant behavior of the cell. This permits quantitative determination of the time after which this perturbation would become negligible. The detailed analysis also permits one to estimate how much the height of the liquid column changes during the course of the experiment due to volume changes on mixing (or diffusion).

T. W. Chapman is with the Department of Chemical Engineering, University of Wisconsin, Madison, Wisconsin 53706.