A New Technique for Precedence-Ordering Chemical Process Equation Sets

An algorithm that combines structural equation analysis with algebraic rearrangement and substitution has been developed to precedence-order algebraic chemical process equations. The SWS (Structural Analysis with Substitutions) algorithm selects an output variable for each equation, rearranges the equation to solve for that variable, and reduces the number of equations by substitution. The algorithm has been used for computer generation of Fortran programs to perform material and energy balances in the design of chemical processes.

SEZER SOYLEMEZ
and
WARREN D. SEIDER
The School of Chemical Engineering
and
The Moore School of Electrical Engineering
The University of Pennsylvania
Philadelphia, Pennsylvania 19104

SCOPE

New mathematical models are regularly created by chemical engineers in the design or simulation of chemical processes. But to use an equation model the engineer must prepare an algorithm for solution of the equations before process performance can be predicted as it varies with the design and operating parameters.

The overall objective of our research has been to reduce the engineer's time necessary to solve the equations that comprise a model and thereby to encourage the creation of mathematical models for the routine evaluation of chemical process alternatives. To accomplish this goal, we have undertaken to study the rules of thumb, or heuristics, applied daily by engineers in creating algorithms for solution of model equations.

The need to systematize procedures for preparation of algorithms to solve an equation model is gaining in importance as digital computers play a more significant role in chemical process design and simulation. In fact, this work was stimulated by the desire to automate the procedures for preparation of algorithms to solve an equation model; hence, the need to identify heuristics and systematize the procedures (Soylemez and Seider, 1972).

In this paper we concentrate solely upon nonlinear algebraic equation sets such as those in material and energy balances for steady state processes. We have extended the previously published methods for algorithm preparation, or precedence-ordering, to account for nonlinearities in the terms of the equations. Furthermore, we have introduced an heuristic that reduces the dimensionality of an equation set by rearrangement and substitution to decrease the computation time in the solution of simultaneous algebraic equations.

CONCLUSIONS AND SIGNIFICANCE

The following heuristics have been selected to create algorithms that solve nonlinear algebraic equations:

1. Wherever possible, equations are rearranged to isolate an output variable which is deleted from the equation set by substitution. The intent is to reduce the dimensionality of the equations and computation times for their solution.

2. Variables to be removed from the equation set are selected from equations containing the fewest unknowns and the smallest degree of nonlinearity, and selected to require the fewest substitutions possible in the remaining equations. The intent is to avoid propagation of nonlinear terms by substitution throughout the remaining equations.

These heuristics are especially effective for material and energy balance problems where the structural matrix is sparse and many of the equations are mildly nonlinear. It is usually possible to reduce the number of equations to be solved iteratively using numerical methods and, in many cases, drastically reduce computation times. In some cases, it is possible to eliminate the need for any iterative calculations.

The following limitations have been observed for the above heuristics:

1. Equations cannot be rearranged when output variables are imbedded in the terms implicitly.

2. Methods of characterizing the nonlinearities are not well defined.

3. The relative magnitude of terms containing nonlinearities is not considered when preparing a solution algorithm.

4. Inequality constraints that are often present in process models are disregarded.

5. Solution algorithms prepared using these heuristics may be less reliable and computationally efficient than algorithms based upon the special structures that characterize an equation set.

However, despite these limitations, algorithms based upon the above heuristics, when they work, are important

S. Soylemez was at the School of Chemical Engineering only. He is at Raytheon, Inc., Boston, Massachusetts.

to the engineer who is often willing to sacrifice computational speed to reduce his efforts in obtaining a solution.

The SWS Algorithm, for implementation of the above heuristics and described in this paper, has been automated and is the basis of a Fortran program that precedence orders material and energy balance equations. This Fortran program, in turn, is the heart of a program that automatically prepares Fortran programs to solve material and energy balance problems (Soylemez and Seider, 1972).

Solution methods for simultaneous nonlinear algebraic equations have received much attention in recent years due to the advent of high speed digital computers. While numerical methods such as Newton-Raphson (Carnahan et al., 1969) may be used to solve for all unknowns simultaneously, in general, the computational labor required to solve n equations increases as n-squared. Furthermore, the numerical procedures for the solution of simultaneous nonlinear equations are iterative and successful convergence of these iterative methods cannot be guaranteed a priori.

This work is based upon the premise that the fewer equations to be solved simultaneously, the smaller the computation time, and in many instances, the greater the chances for successful solution. The heuristics presented are intended to reduce the dimensionality of the problem, that is, the number of equations to be solved simultaneously, before resorting to numerical methods.

PRIOR WORK

Steward (1965) and Lee et al. (1966) are among the early investigators that described algorithms for precedence-ordering nonlinear algebraic equation sets. These algorithms consider the information structure of the equation set in its broadest terms; that is, the identity of the variables occurring in each equation. The algebraic properties of the equation, such as the nonlinearities, are disregarded. Steward's algorithm isolates those equations with the fewest variables that can be solved independently. (Equations having just one unknown are solved first using the Steward algorithm.) The algorithm described by Lee et al. (hereafter referred to as the Rudd algorithm) locates those variables that occur in the fewest equations. Equations containing these variables are reserved to be solved after all other variables have been determined. (When a variable occurs in only a single equation, that equation is solved last for that variable value using the Rudd algorithm.)

Improved versions of these two algorithms were recently reported by Ledet and Himmelblau (1970) and by Christensen and Rudd (1969), respectively. The latter two algorithms are referred to as the Ledet and the Christensen algorithms, respectively. All four algorithms analyze the information structure of the equation set but disregard the algebraic properties, that is, the nonlinearities.

INTRODUCTION TO THE SWS ALGORITHM

The SWS Algorithm analyzes a set of simultaneous nonlinear algebraic equations in which each equation contains two or more unknowns. These equations remain after the Steward and Rudd algorithms have isolated equations to be solved independently. The resulting set of simultaneous equations is referred to as the initial maximal set.

Table 1 illustrates six equations, its structural matrix, and its initial maximal set. Equations (4) and (6) are precedence-ordered (using the Steward algorithm) to be solved first and second for x_3 and x_6, respectively. Equation (2) is precedence-ordered (using the Rudd algorithm) to be solved last for x_5. The remaining three equations form the initial maximal set.

The SWS algorithm orders the maximal set equations for solution by algebraic rearrangement and substitution. One maximal set equation is rearranged to express one of its variables, the output variable, in terms of its other variables. For example, Equation (5) is well-suited for rearrangement because it is linear:

$$x_2 = 3 - x_4 \qquad (5)$$

where x_2 is its output variable. All occurrences of x_2 in the remaining equations [(1) and (3)] are replaced by this

TABLE 1. AN ILLUSTRATIVE EQUATION SET

Equation number	Functional form	Actual equation
(1)	$f_1(x_1,x_6,x_4) = 0$	$x_1x_4 + x_6^2/x_4 - 4 = 0$
(2)	$f_2(x_2,x_5,x_6) = 0$	$x_2x_5 - 3x_6 = 0$
(3)	$f_3(x_1,x_2,x_3,x_4) = 0$	$x_1/x_2 + \ln(x_3/x_4) - 2 = 0$
(4)	$f_4(x_3) = 0$	$x_3^3 + 2x_3^2 - 2x_3 = 0$
(5)	$f_5(x_2,x_4) = 0$	$x_2 + x_4 - 3 = 0$
(6)	$f_6(x_3,x_6) = 0$	$x_3(x_3 + x_6) - 7 = 0$

a. Equation Set

Equations	Variables x_1	x_2	x_3	x_4	x_5	x_6
f_1	1			1		1
f_2		1			1	1
f_3	1	1	1	1		
f_4			1			
f_5		1		1		
f_6			1			1

b. Structural Matrix

Equation number	Functional form	Actual equation*
(1)	$f_1(x_1,x_4) = 0$	$x_1x_4 + \underline{x_6}^2/x_4 - 4 = 0$
(3)	$f_3(x_1,x_2,x_4) = 0$	$x_1/x_2 + \ln(\underline{x_3}/x_4) - 2 = 0$
(5)	$f_5(x_2,x_4) = 0$	$x_2 + x_4 - 3 = 0$

* Known variables are underlined.

c. Initial Maximal Set Equations

Equations	Variables x_1	x_2	x_4
f_1	1		1
f_3	1	1	1
f_5		1	1

d. Initial Maximal Set Structural Matrix expression. In this manner, the number of unknowns in the maximal set is reduced by one, while the features of the original equation are preserved in the equation set in substituted form:

$$x_1x_4 + \underline{x}_6^2/x_4 - 4 = 0 \qquad (1)$$

$$\frac{x_1}{(3 - x_4)} + \ln(\underline{x}_3/x_4) - 2 = 0 \qquad (3)$$

The substituted variable x_2 is evaluated after the unknowns remaining in the maximal set x_1 and x_4 are determined.

This rearrangement and substitution procedure is repeated to remove equation-by-equation from the maximal set until all of the nonlinearities in the original equations are concentrated in the remaining equations to make further reduction impractical. The remaining equations are solved simultaneously using iterative numerical methods.

The equation set above has been reduced to two equations [(1) and (3)] with unknowns x_1 and x_4. Equation (1) can be rearranged to isolate x_1:

$$x_1 = \frac{4 - \underline{x}_6^2/x_4}{x_4} \qquad (1)$$

where x_1 is its output variable. The occurrence of x_1 in the remaining Equation (3) is replaced by this expression which gives

$$\frac{4 - \underline{x}_6^2/x_4}{x_4(3 - x_4)} + \ln(\underline{x}_3/x_4) - 2 = 0 \qquad (3)$$

a single equation in one unknown x_4 that is solved numerically. The substituted variable x_1 is evaluated after x_4 is determined.

Table 2 illustrates the precedence-ordered equation set after application of the Steward, Rudd, and SWS Algorithms. Each equation is solved for its output variable in the order of precedence illustrated. Selection of the output variable for each equation is the role of these algorithms.

The SWS Algorithm considers equations that contain the fewest variables first as candidates for rearrangement, substitution, and removal from the maximal set because these are often simplest to rearrange and substitute. These equations are collected and referred to as the limited set. The structural matrix corresponding to the limited set is referred to as the limited structural matrix. The limited set equations and the limited structural matrix for the maximal set equations in Table 1 are illustrated in Table 3. The variables that occur in the fewest equations in the limited set are good candidates for substitution; for example, x_1 and x_2. Frequently, a variable occurs in only one equation in the limited set (that is, x_1 and x_2). In these cases, it is selected as the output variable for its equation, except when the equation contains one or more of the nonlinearities to be described. After each equation is removed from the maximal set, the remaining equations are altered. Therefore, the limited set is reconstructed after each application of the SWS Algorithm.

When all variables occur more than once in the limited set, the nonlinearity properties of the equations are important in the selection of output variables. In order to assess the relative difficulty of rearranging the equations, three algebraic properties are considered for each equation. They are the number of terms in the equation, the degree of nonlinearity of each term in the equation, and the degree of nonlinearity of the equation as defined below.

TABLE 2. PRECEDENCE ORDERED EQUATION SET

Order of solution	Equation	Output variable	Final form*
1	f_4	x_3	$x_3(x_3^2 - 2x_3 - 2) = 0$
2	f_6	x_6	$x_6 = 7/\underline{x}_3 - \underline{x}_3$
3	f_3	x_4†	$\frac{4 - \underline{x}_6^2/x_4}{x_4(3 - x_4)} + \ln(\underline{x}_3/x_4) - 2 = 0$
4	f_1	x_1	$x_1 = \frac{4 - \underline{x}_6^2/\underline{x}_4}{\underline{x}_4}$
5	f_5	x_2	$x_2 = 3 - \underline{x}_4$
6	f_2	x_5	$x_5 = 3\underline{x}_6/\underline{x}_2$

* Known variables are underlined.
† Iterative numerical solution for x_4.

TABLE 3. LIMITED SET EQUATIONS AND STRUCTURAL MATRIX FOR EQUATION SET IN TABLE 1

Equation number	Functional form	Actual equation
(1)	$f_1(x_1,x_4)$	$x_1x_4 + \underline{x}_6^2/x_4 - 4 = 0$
(5)	$f_5(x_2,x_4)$	$x_2 + x_4 - 3 = 0$

a. Limited Set Equations

Equation	Variables x_1	x_2	x_4
1	1		1
5		1	1

b. Limited Structural Matrix

The number of terms in an equation is one more than the number of plus or minus signs which are not contained within a parenthetical expression; for example; the equation $a + b + c = 0$ has three terms, but the equation

$$a + (b + c)/d = 0,$$

term 1 term 2

has only two terms.

The degree of nonlinearity is either 0, 1, 2, or 3 and is assigned to a term or an equation as a measure of its nonlinearity. A term has a degree of nonlinearity equal to zero when it contains no unknown variables. It has a degree of nonlinearity equal to one when it contains only one unknown and when it is linear with respect to that unknown. A term has a degree of nonlinearity equal to two when it is of the form either $a \cdot b$ or a/b, where a and b are linear expressions with respect to all the unknowns they contain and where the unknowns in a and b are mutually exclusive. A term has a degree of nonlinearity equal to three when it contains any other nonlinearity, for example, $\ln(x)$, y^2. The degree of nonlinearity of an equation equals the highest degree of nonlinearity encountered among its terms. Table 4 illustrates the degree of nonlinearity for typical terms in an equation.

As mentioned previously, the degree of nonlinearity of the terms in an equation is important in the selection of output variables. Unknowns occurring in nonlinear terms are often difficult to isolate during algebraic rearrangement. Furthermore, equations containing nonlinear terms are not good candidates to remove from the maximal set

TABLE 4. DEGREE OF NONLINEARITY

Terms in equation

$$\underline{1}\quad \underline{2}\quad \underline{3}\qquad \underline{4}\qquad \underline{5}$$

$$4 + 3x + \frac{x}{y} + \frac{(2x-3)}{(3z+5)} + xz^2 = 0$$

Term	Degree of nonlinearity
1	0
2	1
3	2
4	2
5	3

because substitution spreads the nonlinearities, which is counter to the objective to concentrate the nonlinearities in few equations. For this reason, in particular, the SWS Algorithm has been designed not to consider any equations having degree of nonlinearity equal to three when selecting output variables and equations to be removed from the maximal set. After each substitution, the degree of nonlinearity of all remaining equations is recomputed. When terms cancel, some of the nonlinearities may be cancelled, simplifying the equations, and reducing their degree of nonlinearity. Usually, however, substitution increases or does not alter the degree of nonlinearity of the remaining equations. Since the simplest equations are selected for removal from the maximal set, the nonlinearity of the remaining equations is usually not increased significantly. Hence, the SWS Algorithm does not record an increase in the degree of nonlinearity of any equations remaining in the maximal set. The ramifications of this feature will be explored.

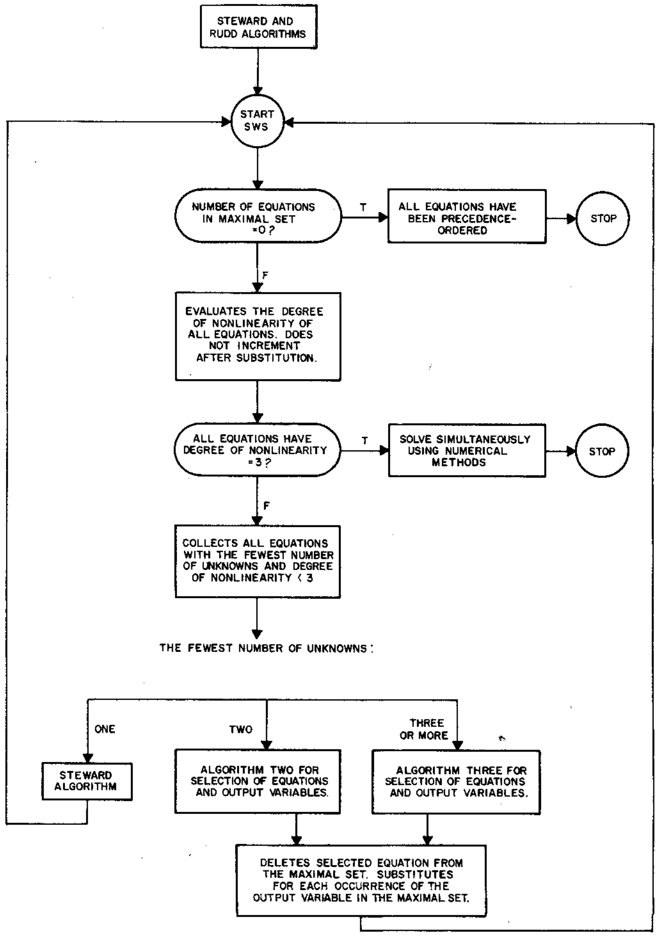

Fig. 1. An overview of the SWS Algorithm.

DETAILS OF THE SWS ALGORITHM

An overview of the SWS Algorithm is presented in Figure 1. The SWS Algorithm is applied when all equations in the maximal set contain two or more unknowns and no variable occurs in just one equation.

The SWS Algorithm is used to select each equation and its output variable to be removed from the maximal set. First, the number of equations remaining in the maximal set is checked. When equal to zero, all equations have been precedence-ordered. Next, the degree of nonlinearity of each equation is evaluated. When all equations remaining in the maximal set have degree of nonlinearity equal to three, they are solved simultaneously using numerical methods. Otherwise, the SWS Algorithm collects all equations with the fewest unknowns n_f and degree of nonlinearity less than three. n_f determines which of three algorithms is used to select the next equation to be removed from the maximal set and its output variable:

n_f	Algorithm
1	Steward Algorithm
2	Algorithm Two
$\geqq 3$	Algorithm Three

Equations having degree of nonlinearity equal tc three are not considered because substitution propagates their nonlinearities throughout the remaining equations, as described earlier.

Equations With Only Two Unknowns—Algorithm Two

An overview of Algorithm Two is presented in Figure 2. First, the limited structural matrix for equations with two unknowns is constructed. The number of incidences for each variable is counted by summing the columns of the matrix. Three possibilities may result:

1. All variables occur once in the limited structural matrix.

2. There is a single variable that has the most incidences, the maximum incidence variable.

3. There is more than one maximum incidence variable.

For material and energy balance equation sets, the second possibility occurs most often, followed by the first and third in order. We proceed to examine the algorithm for each of these cases:

1. When all variables occur once in the limited structural matrix, each of the equations is studied to determine if either of its variables has fewer incidences in the maximal set, or occurs only in linear terms in the equation, or has the lowest S, where $S = \Sigma d_i$ and d_i is the degree of nonlinearity for each equation that contains the variable.

When anyone of these conditions is satisfied, the satisfying variable is the output variable of the equation. Otherwise, the output variable is the one closest to linearity in all terms of the equation. The output variable is expressed in terms of the other unknown. This expression is substituted in the remaining maximal set equations and the SWS Algorithm is restarted for the remaining equations.

Conditions a, b and c are designed to select output variables to minimize the number of substitutions, to minimize the difficulty in rearranging the equations, and to reduce the nonlinearity of the remaining equations, respectively.

2. When there is a single maximum incidence variable,

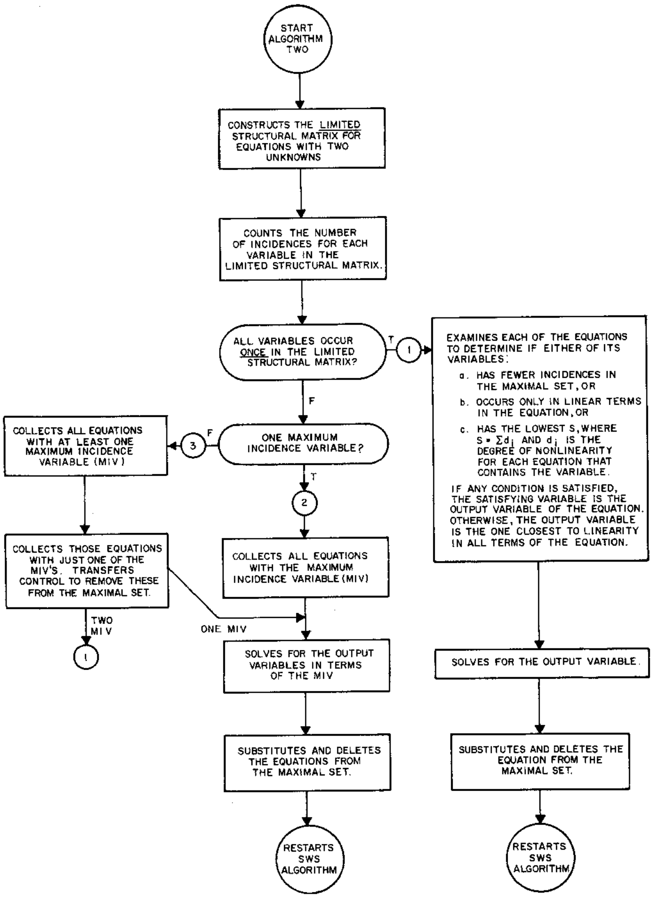

Fig. 2. An overview of Algorithm Two.

all equations containing the variable are collected. In each equation, the other variable is the output variable. The output variable is expressed in terms of the maximum incidence variable and substituted for in the remaining maximal set equations. Then, the SWS Algorithm is restarted for the remaining equations.

This step is designed to reduce the number of substitutions and to distribute variables that occur often throughout the terms of the equations while eliminating variables that occur infrequently.

3. When there is more than one maximum incidence variable, all equations with at least one maximum incidence variable are collected. Of these, those equations with just one of the maximum incidence variables are treated using the algorithm in 2 above. For the equations containing two maximum incidence variables, the algorithm in 1 above is applied to select the equation to be removed and its output variable.

Algorithm Two is used to select equations to be removed from the maximal set and their output variables as long as equations with two unknowns remain.

Equations With Three or More Unknowns—Algorithm Three

An overview of Algorithm Three is presented in Figure 3. First, the limited structural matrix for equations with the fewest unknowns is constructed. The number of incidences for each variable is counted by summing the columns of the matrix. When a variable occurs only once, it is selected as the output variable for its equation, unless more than one variable occurs only once, in which case Algorithm Two, Path 1, is used to select the output variable. The output variable is isolated and substituted for in the remaining maximal set equations.

One objective of Algorithm Three is to select output variables that have the fewest occurrences in the limited structural matrix so as to minimize the number of substitutions. The most desirable situation occurs when a variable has a single incidence in the limited structural matrix.

When the minimum number of incidences is greater than one, Algorithm Three performs the conditional tests in Figure 3 where $n =$ the number of variables in the limited set of equations. Their objective is to collect a subset of the equations whose structural matrix contains a variable that occurs once. This variable is selected as the output variable for the equation in which it occurs. The conditional tests are designed to collect the smallest subset of equations initially. When its structural matrix does not contain a variable that occurs once, the test criteria are relaxed to add equations to the subset.

The first tests collect subsets of equations that contain the fewest variables that occur most frequently in the limited structural matrix. First, equations containing the maximum incidence variable are selected. If this set does not contain a variable that occurs once, equations containing two variables that occur most frequently are collected, and so on.

When these subsets do not contain a variable that occurs only once, a subset of linear equations is assembled. One final subset of equations containing a single linear term is assembled if necessary.

Should all of these subsets fail to contain a variable that occurs only once, Algorithm Three collects the equations containing the one or more variables that have the fewest number of incidences in the limited structural matrix, that is, the minimum incidence variables. When there is only one minimum incidence variable, it is selected as the output variable for the least nonlinear equation. Otherwise, the equation containing the most minimum incidence variables and the least degree of nonlinearity is selected for removal from the maximal set. Algorithm Two, Path 1, is used to select the output variable for this equation.

An Example—Leaching Process Material Balance

An inert solid with adsorbed solute is contacted in a leaching unit by a solvent stream containing solute in small concentrations. The solvent stream leaches some of the solute from the inert solid and leaves the process unit richer in solute, while the inert solid stream loses some of the adsorbed solute and leaves the leacher leaner in solute. Given the flow rate and weight fractions of the components in the feed streams, the product stream flow rates and compositions can be computed. Table 5 illustrates the leaching unit schematic and stream variables. Table 6 contains the material balance equations and the equations that describe the equilibrium between the liquid and solid streams. Equation (2) equates the solid free concentrations of the product streams and Equation (3) is an empirical equilibrium correlation with a, b, and c known for a given solute-solvent-inert solid system. There are 19 variables and 10 equations; hence, nine design variables have been selected in accordance with the problem statement. These are $\{F_1\ x_{11},\ x_{12}$ and $F_4,\ x_{41},\ x_{42}$ and $a,\ b,\ c\}$ and are underlined in Table 6. It remains to solve the ten equations for ten unknowns. Their structural matrix is given in Table 7.

The Steward Algorithm selects x_{13}, x_{33}, and x_{43} as the output variables of Equations (7), (1), and (10), respectively, and precedence-orders these equations to be solved first. The Rudd Algorithm does not apply since none of the variables occur in only one equation.

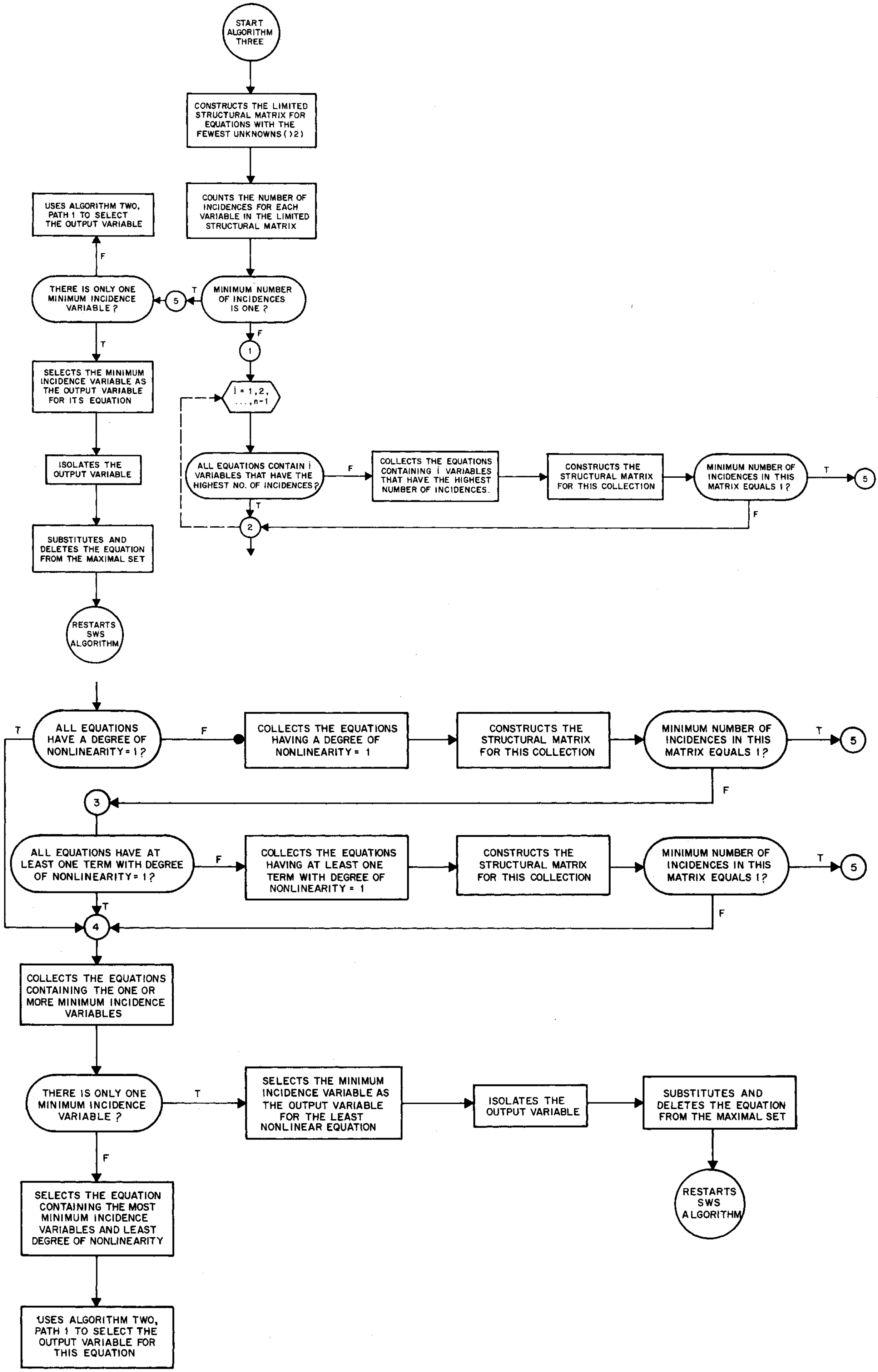

Fig. 3. An overview of Algorithm Three.

There are seven equations in the initial maximal set. According to Algorithm Two, the limited structural matrix for equations having two unknowns is constructed (see Table 8).

There is only one maximum incidence variable F_2. Hence, the variables F_3 and x_{23} in Equations (4) and (6) are selected as output variables. Since both x_{31} and x_{32} occur once in the limited structural matrix, the output variable x_{32} is selected for Equation (9) using criterion a of path 1 in Algorithm Two. The rearranged equations are

$$F_3 = \underline{F_1} + \underline{F_4} - F_2 = \underline{\alpha_1} - F_2 \quad (4)$$

$$x_{23} = (\underline{F_1}\underline{x_{13}} + \underline{F_4}\underline{x_{43}})/F_2 = \underline{\alpha_2}/F_2 \quad (6)$$

$$x_{32} = 1 - x_{31} \quad (9)$$

where

$$\underline{\alpha_1} = \underline{F_1} + \underline{F_4}$$

$$\underline{\alpha_2} = \underline{F_1}\underline{x_{13}} + \underline{F_4}\underline{x_{43}}$$

Note that all known variables are underlined. These include design variables and the output variables of equations higher in the precedence order.

The equations remaining in the maximal set after substitution of these expressions are shown in Table 9. The new limited structural matrix (see Table 10) contains all the maximal set equations except Equation (3) which has a degree of nonlinearity of three. Since the limited structural matrix contains three variables, Algorithm Three applies. The minimum number of incidences of the variables in the limited structural matrix is two. Hence, the conditional tests in Algorithm Three are used. The subset of equations with one or more linear terms contains a variable that occurs only once, x_{31}. It is selected as the output variable of Equation (2) and the rearranged equation is

$$x_{31} = x_{21}/(x_{21} + x_{22}) \quad (2)$$

The equations remaining in the maximal set after the expression for x_{31} is substituted are listed in Table 11, as well as their limited structural matrix. Note that Equation (5) remains in the limited structural matrix although its degree of nonlinearity has been increased to three (see the earlier discussion). Again, there are three variables and Algorithm Three applies. Since the minimum number of incidences is two, the conditional tests are used. The subset of equations with one or more linear terms [Equation (8)] contains three variables that occur only once, x_{21}, x_{22}, and F_2; that is, all of the variables remaining in the maximal set. x_{22} is selected as the output vari-

TABLE 5. SCHEMATIC FOR A THREE COMPONENT, SINGLE-STAGE, COUNTERCURRENT LEACHING UNIT

F_i - MASS FLOW RATE FOR STREAM i

x_{ij} - MASS FRACTION OF COMPONENT j IN STREAM i.

COMPONENT	NUMBER (j)
1	SOLUTE
2	SOLVENT
3	SOLIDS

TABLE 6. LEACHING UNIT EQUATIONS

Equation number	Equation*
Equipment equations	
(1)	$x_{33} = 0$
(2)	$x_{21}/(x_{21} + x_{22}) = x_{31}/(x_{31} + x_{32})$
(3)	$(x_{21} + x_{22})/x_{23} = \exp[\underline{a}(x_{21}/(x_{21} + x_{22}))^{\underline{b}} - \underline{c}]$
Material balance equations	
(4)	$F_2 + F_3 - \underline{F_1} - \underline{F_4} = 0$ (overall)
(5)	$F_2x_{21} + F_3x_{31} - \underline{F_1}\underline{x_{11}} - \underline{F_4}\underline{x_{41}} = 0$ (solute)
(6)	$F_2x_{23} - \underline{F_1}x_{13} - \underline{F_4}x_{43} = 0$ (inert solid)
Weight fraction equations	
(7)	$\underline{x_{11}} + \underline{x_{12}} + x_{13} = 1$
(8)	$\sum_{j=1}^{3} x_{2j} = 1$
(9)	$\sum_{j=1}^{2} x_{3j} = 1$
(10)	$\underline{x_{41}} + \underline{x_{42}} + x_{43} = 1$

* Design variables are underlined (F_1, x_{11}, x_{12}, F_4, x_{41}, x_{42}, a, b, c).

TABLE 7. THE STRUCTURAL MATRIX FOR THE LEACHING PROCESS EQUATIONS IN TABLE 6

	Variables									
Equation	x_{13}	F_2	x_{21}	x_{22}	x_{23}	F_3	x_{31}	x_{32}	x_{33}	x_{43}
(1)									1	
(2)			1	1			1	1		
(3)			1	1	1					
(4)		1				1				
(5)		1	1			1	1			
(6)	1	1			1					1
(7)	1									
(8)			1	1	1					
(9)							1	1		
(10)										1

TABLE 8. THE LIMITED STRUCTURAL MATRIX FOR THE INITIAL MAXIMAL SET

	Variables				
Equations	F_2	F_3	x_{23}	x_{31}	x_{32}
4	1	1			
6	1		1		
9				1	1
Number of incidences of the variables	2	1	1	1	1

TABLE 9. MAXIMAL SET EQUATIONS AFTER ONE APPLICATION OF THE SWS ALGORITHM

Equation number	Equation
(2)	$x_{21} - x_{31}(x_{21} + x_{22}) = 0$
(3)	$(x_{21} + x_{22})F_2/\underline{\alpha_2} - \exp[\underline{a}(x_{21}/(x_{21} + x_{22}))^{\underline{b}} - \underline{c}] = 0$
(5)	$F_2x_{21} + (\underline{\alpha_1} - F_2)x_{31} - \underline{F_1}\underline{x_{11}} - \underline{F_4}\underline{x_{41}} = 0$
(8)	$x_{21} + x_{22} + \alpha_2/F_2 = 1$

able arbitrarily according to step b of Path 1 in Algorithm Two. The rearranged equation is

$$x_{22} = 1 - \underline{\alpha}_2/F_2 - x_{21} \quad (8)$$

After the expression for x_{22} is substituted, there are only two equations with two unknowns (see Table 12). x_{21} is selected as the output variable of Equation (5) using path 1 in Algorithm Two. When Equation (5) is rearranged, the resulting expression is

$$x_{21} = \underline{\alpha}_3(1 - \underline{\alpha}_2/F_2)/(\underline{\alpha}_1 - \underline{\alpha}_2)$$

where

$$\underline{\alpha}_3 = \underline{F}_1\underline{x}_{11} + \underline{F}_4\underline{x}_{41}$$

After substituting for x_{21} in Equation (3), the remaining equation is

$$F_2/\underline{\alpha}_2 - 1 = \exp\left[\underline{a}(\underline{\alpha}_3/(\underline{\alpha}_1 - \underline{\alpha}_2))^{\underline{b}} - \underline{c}\right]$$

This equation can be solved directly for F_2. The result is

$$F_2 = \underline{\alpha}_2(\exp \underline{\beta} + 1)$$

where

$$\underline{\beta} = \underline{a}(\underline{\alpha}_3/(\underline{\alpha}_1 - \underline{\alpha}_2))^{\underline{b}} - \underline{c}$$

The final order of solution is listed in Table 13.

Table 10. The Limited Structural Matrix for the Maximal Set Equations in Table 9

Equations	Variables F_2	x_{21}	x_{31}	x_{22}
2		1	1	1
5	1	1	1	
8	1	1		1
Number of incidences of the variables	2	3	2	2

Table 11. Equations Remaining in the Maximal Set After x_{31} is Substituted, and Their Limited Structural Matrix

Equation number	Equation
(3)	$(x_{21} + x_{22})F_2/\underline{\alpha}_2 - \exp\left[\underline{a}(x_{21}/(x_{21} + x_{22}))^{\underline{b}} - \underline{c}\right] = 0$
(5)	$F_2x_{21} + (\underline{\alpha}_1 - F_2)x_{21}/(x_{21} + x_{22}) - \underline{F}_1\underline{x}_{11} - \underline{F}_4\underline{x}_{41} = 0$
(8)	$x_{21} + x_{22} + \underline{\alpha}_2/F_2 = 1$

Equations	Variables F_2	x_{21}	x_{22}
5	1	1	1
8	1	1	1
Number of incidences of the variables	2	2	2

Table 12. Equations Remaining in the Maximal Set After x_{22} is Substituted

Equation number	Equation
(3)	$(F_2/\underline{\alpha}_2 - 1) - \exp\left[\underline{a}(x_{21}/(1 - \underline{\alpha}_2/F_2))^{\underline{b}} - \underline{c}\right] = 0$
(5)	$F_2x_{21} + (\underline{\alpha}_1 - F_2)x_{21}/(1 - \underline{\alpha}_2/F_2) - \underline{F}_1\underline{x}_{11} - \underline{F}_4\underline{x}_{41} = 0$

Table 13. Precedence-Ordered Leaching Unit Equations

Order of solution	Equation number	Output variable	The Final Form of the Equation
1	(7)	x_{13}	$x_{13} = 1 - \underline{x}_{11} - \underline{x}_{12}$
2	(10)	x_{43}	$x_{43} = 1 - \underline{x}_{41} - \underline{x}_{42}$
3	(1)	x_{33}	$x_{33} = 0$
4	(3)	F_2	$F_2 = \underline{\alpha}_2(\exp \underline{\beta} + 1)$
5	(5)	x_{21}	$x_{21} = \underline{\alpha}_3(1 - \underline{\alpha}_2/F_2)/(\underline{\alpha}_1 - \underline{\alpha}_2)$
6	(8)	x_{22}	$x_{22} = 1 - \underline{\alpha}_2/F_2 - x_{21}$
7	(2)	x_{31}	$x_{31} = x_{21}/(x_{21} + x_{22})$
8	(9)	x_{32}	$x_{32} = 1 - x_{31}$
9	(6)	x_{23}	$x_{23} = \underline{\alpha}_2/F_2$
10	(4)	F_3	$F_3 = \underline{\alpha}_1 - F_2$

where

$$\underline{\alpha}_1 = \underline{F}_1 + \underline{F}_4$$
$$\underline{\alpha}_2 = \underline{F}_1\underline{x}_{13} + \underline{F}_4\underline{x}_{43}$$
$$\underline{\alpha}_3 = \underline{F}_1\underline{x}_{11} + \underline{F}_4\underline{x}_{41}$$
$$\beta = \underline{a}(\underline{\alpha}_3/(\underline{\alpha}_1 - \underline{\alpha}_2))^{\underline{b}} - \underline{c}.$$

Evaluation of the SWS Algorithm

The advantages of substitution based upon the structural properties and nonlinearities of an equation set have been demonstrated. It is usually possible to reduce the number of equations to be solved iteratively using numerical methods and drastically reduce computation times (Soylemez and Seider, 1972). This is especially true for material and energy balance problems where the structural matrix is sparse and many of the equations are mildly nonlinear (degree of nonlinearity = 2). In some cases, it is possible to eliminate the need for any iterative calculations (through cancellation of like terms, etc.). This is the principal justification for substitution.

There is room for considerable improvement in the classification of nonlinearities in an algebraic equation. The present degree of nonlinearity does not distinguish between the nonlinearities within the terms; that is, the relative nonlinearity of the variables within the terms is not given by the degree of nonlinearity. Consequently, in Path 1 of Algorithm Two the choice of a variable that occurs least nonlinearly in an equation cannot be based upon the current degree of nonlinearity. Furthermore, there is a vast difference in many of the nonlinearities that are currently assigned degree of nonlinearity equal to three. Members of our research team are exploring better classification methods. In the interim, we do not increment the degree of nonlinearity after substitutions involving expressions that have been selected to contain the fewest nonlinearities detectable.

The SWS Algorithm does not consider the numerical magnitude of the variables in the equations when selecting an output variable. While some terms may be highly nonlinear, they may be small and not worthy of examination. Good guess values of the variables would be sufficient to estimate the magnitude of the nonlinear terms relative to each other. Furthermore, consideration of the variable magnitudes during precedence-ordering should reduce the possibility for roundoff error and numerical errors during calculations to solve the equations.

Another shortcoming of the SWS Algorithm is its failure to account for the inequality constraints imposed upon the variables in a model. It is, therefore, possible for the precedence-ordered equations to compute physically unrealistic values; for example, negative mole fractions. This is a disadvantage of the SWS Algorithm when compared with constrained optimization nonlinear programming methods that minimize the summation of the residuals of a set of simultaneous equations, thereby solving the equations subject to the inequality constraints (McMillan, 1970).

It should also be noted that repeated substitutions may lead to propagation of errors during computation of the remaining complex terms. To avoid this problem, higher precision arithmetic is required and is available on most digital computers.

The SWS Algorithm represents the authors' attempt to state in algorithmic form many of the heuristics followed by mathematicians and engineers when solving sets of algebraic equations. Naturally, for process design problems characterized by equations having special structures, there are heuristics to exploit these structures. For example, when the Jacobian of an equation set is arranged in blocked tridiagonal form, the Newton-Raphson method is more efficient for countercurrent, multistaged processes. However, it is unrealistic to expect that the more general heuristics of the SWS Algorithm can lead to solution algorithms that are as reliable and efficient computationally.

A comparable situation, often overlooked, exists when Fortran compilers translate Fortran programs into machine language programs. For evaluation of lengthy arithmetic expressions, the programs that execute most rapidly are written in machine language. Yet, computational speed is often sacrificed to simplify the engineer's programming chores.

Similarly, algorithms to solve algebraic equations are more reliable and efficient computationally when they are based upon the peculiarities of the equations. Yet, when they work, algorithms based upon more general heuristics are important to the designer who is often unfamiliar with the peculiarities of the equations. Naturally when the latter algorithms do not work, the special structures of the equations must be examined closely. The objective of the SWS Algorithm is to eliminate this step, which often requires much time and skill, for the designer who is often willing to sacrifice computational speed to reduce his efforts in obtaining a solution.

Preselected Output Variables

The SWS Algorithm has been extended to allow for preselection of output variables when desirable (Soylemez, 1971). This feature is important for material and energy balance problems involving properties such as enthalpy, fugacity, and liquid-phase activity coefficients. These explicit, complex, nonlinear functions of temperature, pressure, and composition are usually impossible to rearrange to isolate temperature, pressure, or composition when the property is an output variable of another equation. In these cases, it is usually possible to simplify the precedence-order, that is, avoid iterative numerical solution of the property estimation functions, through preselection of the properties as the output variables for their estimation functions. As a consequence, the properties cannot be candidates for output variables in the other equations.

This has not been a severe limitation in our work thus far. The SWS Algorithm has precedence-ordered the material and energy balance equations for an equilibrium flash process and an equilibrium reactor for production of carbon monoxide from natural gas and steam, where the enthalpies and equilibrium constants were preselected as output variables in their respective estimation functions. The extensions to the algorithm, these examples, and tradeoffs in the preselection of output variables have been described (Soylemez, 1971).

AUTOMATIC PROGRAM GENERATION

A Fortran computer program has been written to implement the SWS Algorithm. It has several limitations that are described in the literature (Soylemez and Seider, 1972). Briefly, these limitations are:

1. After substitution, the program cannot cancel terms, factor, and cancel like expressions in the numerator and denominator of fractions because these symbolic algebraic operations are too difficult to implement in Fortran (Fortran was selected earlier in our work to allow for machine interchangeability of the program). This limitation can be avoided by using a computer language better suited to symbolic algebraic operations (Sammet, 1967).
2. In Algorithm Two, Path 1, the relative nonlinearity of variables in an equation cannot be determined. Hence, the output variable is selected arbitrarily.

The program reads cards that contain the algebraic equations, precedence-orders the equations using the SWS Algorithm, and prepares a Fortran program to solve the equations (the program follows the precedence-order). In other words, the program automatically generates a program to solve the equations. It automates the program preparation step. See Soylemez and Seider (1972) for further discussion.

ACKNOWLEDGMENTS

This research work was partially sponsored by the Exxon Education Foundation and the Ford Foundation. We wish to cite numerous discussions with the members of our Chemical Engineering Calculation System project that were very helpful. Professor Ronald L. Klaus and Mr. Michael Hanyak deserve special thanks.

NOTATION

F_i = mass flow rate of stream i, lb/hr
i = stream number
j = component (chemical) number
n_F = number of unknowns in the equation containing the fewest unknowns
x_{ij} = weight fraction of component j in stream i

LITERATURE CITED

Carnahan, B., H. A. Luther, and J. O. Wilkes, "Applied Numerical Methods," p. 319, Wiley, New York (1969).
Christensen, J. H., and D. F. Rudd, "Structuring Design Computations," *AIChE J.*, **15**, 94 (1969).
Ledet, W. P., and D. M. Himmelblau, "Decomposition Procedures for the Solving of Large Scale Systems," *Advan. Chem. Eng.*, **8**, 196 (1970).
Lee, W., J. H. Christensen, and D. F. Rudd, "Design Variable Selection to Simplify Process Calculations," *AIChE J.*, **12**, 104 (1966).
McMillan, C., Jr., "Mathematical Programming," p. 104, Wiley, New York (1970).
Sammet, J. E., "Formula Manipulation by Computer," *Advan. in Computers*, **8**, 47 (1967).
Soylemez, S., "Computer Generated FORTRAN Programs to Prepare Material and Energy Balances for Process Units," Ph.D. thesis, Univ. Pennsylvania, Philadelphia (1971).
———., and W. D. Seider, "Computer Generated FORTRAN Programs for Material and Energy Balancing Process Units," *AIChE CHEC Workshop Series*, **2**, 162 (1972).
Steward, D. V., "Partitioning and Tearing Systems of Equations," *J. SIAM*, **2B**, 345 (1965).

Manuscript received July 12, 1972; revision received March 5 and accepted April 6, 1973.