Hubbard, M. G., and A. E. Dukler, "The Characterization of Flow Regimes for Horizontal Two-Phase Flow," *Proceeding of the 1966 Heat Transfer and Fluid Mechanics Institute*, M. A. Saad and J. A. Moller, ed., pp. 100-121, Stanford University Press, Calif. (1966).
Jeffreys, H., "On the Formation of Water Waves by Wind," *Proc. Royal Soc.*, A **107**, 189 (1925).
———., "On the Formation of Water Waves by Wind (second paper), *ibid., Proc. Royal Soc.*, **A110**, 241 (1926).
Kordyban, E. S., and T. Ranov., "Mechanism of Slug Formation in Horizontal Two-Phase Flow," *Trans. ASME, J. Basic Engineering*, **92**, Series D, No. 4, 857-864 (1970).
Kosterin, S. I., *Izvestiya Akademii Nauk., SSSR., O. T. N.*, **12**, 1824, USSR (1949).
Levich, V. G., *Physicochemical Hydrodynamics*, Prentice-Hall, Englewood Cliffs, N. J. (1962).
Lockhart, R. W., and R. C. Martinelli, "Proposed Correlation of Data for Isothermal Two-Phase, Two-Component Flow in Pipes," *Chem. Eng. Progr.*, **45**, 39-48 (1949).
Mandhane, J. M., G. A. Gregory, and K. Aziz, "A Flow Pattern Map for Gas-Liquid Flow in Horizontal Pipes," *Intern. J. Multiphase Flow*, **1**, 537-553 (1974).
Milne-Thomson, L. M., *Theoretical Hydrodynamics*, The MacMillan Co., New York (1960).
Stewart, R. W., "Mechanics of the Air Sea Interface, Boundary Layers and Turbulence," *Phys. Fluids*, **10**, S47 (1967).
Wallis, G. B., and J. E. Dobson, "The Onset of Slugging in Horizontal Stratified Air-Water Flow," *Intern. J. Multiphase Flow*, **1**, 173-193 (1973).
White, P. D., and R. L. Huntington, "Horizontal Co-Current Two-Phase Flow of Fluids in Pipe Lines," *Petrol. Eng.*, **27**, No. 9, D40 (Aug., 1955).

Manuscript received May 30, 1975; revision received and accepted July 25, 1975.

The Selection of Design Variables in Systems of Algebraic Equations

Present algorithms for the selection of design variables produce a single combination of design variables which results in a solution sequence of minimum difficulty. A new theory entitled *solution mapping* results in information about all optimal combinations of design variables. This theory gives detailed information about the structure of all underspecified systems of algebraic equations which do not contain persistent iteration.

NEIL L. BOOK
and
W. FRED RAMIREZ

Department of Chemical Engineering
University of Colorado
Boulder, Colorado 80302

SCOPE

Typical systems of algebraic equations in steady state, macroscopic design problems have a few characteristics which are dominating factors in the difficulty of obtaining solutions. Usually the equations contain few variables (are sparse) and are nonlinear. Solution strategies for sparse systems of equations are, in general, easily obtained. Conversely, the nonlinearity of the equations increases the difficulty of obtaining a solution.

A general characteristic of a system of algebraic design equations is that the number of variables is equal to or greater than the number of equations. When the number of variables is greater than the number of equations, a set of design variables must be assigned numerical values in order that the system be reduced to a determinant system with an equal number of equations and variables. Judicious selection of design variables can lead to a set of equations whose solution sequence encounters a minimum of difficulty.

Solution sequences for systems of algebraic equations can be separated into three classes: one-to-one or acyclic, simultaneous, and iterative solution sequences. A one-to-one, as opposed to simultaneous or iterative solution sequences, solves each equation, whether linear or nonlinear, in a one at a time technique and does not require any assumed solution points. Systems of equations which can be reduced to a determinant system with a one-to-one solution sequence are said to be without persistent iteration. Any system which cannot be reduced such that it has a one-to-one solution sequence is said to contain persistent iteration.

For systems of algebraic equations without persistent iteration, all one-to-one solution sequences are considered as equal minimum difficulty solution strategies. This is a somewhat arbitrary objective function, since it might be less difficult to solve a set of linear simultaneous equations than a series of acyclic equations for nonlinear elements.

Previous algorithms developed for the selection of design variables in systems of equations without persistent iteration are based on the pioneering work of Steward (1962) which introduced the concept of admissable output sets. An admissable output set has two properties: each equation contains exactly one output variable, and each variable appears as the output element of exactly one equation.

The work by Lee et al. (1966) was one of the earliest attempts to base design variable selection on a mathematiical basis. This algorithm operates on a bipartite graph representation for a system of equations and operates only on systems of equations without persistent iteration. The Lee et al. (1966) algorithm was adapted to operate on occurrence matrix representations as presented in Rudd and Watson (1968). This algorithm assigns an admissable output set to a system of equations. The result of the algorithm is a single, optimal combination of design variables and a one-to-one solution sequence for the system of equations.

A system of equations can have many combinations of design variables which will result in one-to-one solution

Correspondence concerning this paper should be addressed to W. Fred Ramirez.

sequences. Each of the optimal combinations can have varying degrees of applicability to a specific design problem. The purpose of this study was to obtain information on all optimal combinations of design variables which resulted in a minimum difficulty (one-to-one) solution sequence. This information allows the designer to obtain an optimal combination of design variables with the most applicability to a design problem.

CONCLUSIONS AND SIGNIFICANCE

Simple algorithms have been developed which will determine if a set of design variables result in a one-to-one solution sequence (Book, 1974). If these algorithms are applied for each possible design variable combination, the result is a listing of all optimal combinations of design variables. Unfortunately, typical systems of design equations contain large numbers of design equations and have several design variables associated with the system of equations. Combinatorial problems arise rapidly with the number of equations which prohibit the use of such simple algorithms.

The algorithms presented here are based on a simple theory entitled *solution mapping*. The solution mapping theory defines all combinations of design variables which result in one-to-one solution sequences and also results in specific information about the structure of a system of algebraic equations. The structural information can severely reduce combinatorial problems encountered in analyzing all combinations of design variables. This allows the designer to quickly choose among the optimal combinations for the combination most suited to a specific design problem.

Although the solution mapping theory is most applicable to systems of design equations in which the designer has a great deal of flexibility in selecting combinations of design variables, the technique is applicable to any underspecified system of algebraic equations which does not contain persistent iteration.

STATEMENT OF THE PROBLEM

A system of N equations with M unknown variables can be written as

$$\mathbf{f}(\mathbf{x}) = 0 \qquad (1)$$

where $\mathbf{x}$ is the M dimensional vector of variables, and $\mathbf{f}$ is an N dimensional set of linear and nonlinear operators.

To obtain an N by N determinate system of equations, $M - N = D$, design variables must be assigned numerical values. There are $C_N^M = C_D^M = M!/N!D!$ possible combinations of design variables. The combinatorial problems become quickly prohibitive in checking all possible combinations of design variables for a minimum difficulty solution sequence.

The combinatorial problems can be reduced by analyzing the structure of the system of equations. One of the most convenient techniques for expressing the structure of a system of equations is the occurrence matrix representation. The columns of the occurrence matrix represent the M variables, and the rows represent the N equations. If variable x_j appears in equation i, then a flag is set to true in element (i, j). If variable x_j does not appear in equation i, then a flag is set to false in element (i, j).

The number of elements set true in row i of the occurrence matrix is defined as the equation degrees of freedom f_i of equation i. The number of elements of column j which are set true is defined as the variable degrees of freedom r_j for variable j.

An example process flow sheet for a mixer-exchanger-mixer system is shown in Figure 1. The following assumptions were made in the development of the design equations which model the system: steady state operation, pressure effects were ignored, all physical properties were known and constant, perfect mixing, no energy loss, and cost information was not important.

The twenty eight equations of Figure 2 were generated which model the steady state operation of the design system. The variables were assigned numbers as indicated in Table 1. This problem was presented by Ramirez and Vestal (1972).

Variables T_5 and $x_{3,6}$ were specified as known variables in the statement of the design problem. Equations (9),

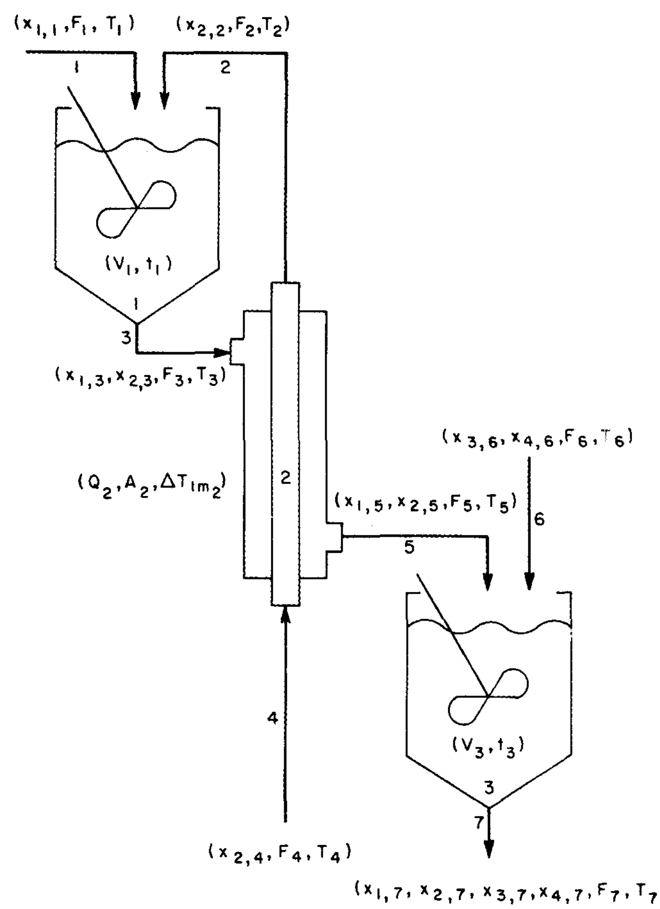

Fig. 1. Mixer—exchanger—mixer system.

(10), (12), and (16) were arbitrarily chosen to be removed to eliminate four redundant material balance equations. Further, Equations (1), (2), (4), and (6) and the variables which appeared in them were removed because the equations had a degree of freedom of 1.

The occurrence matrix representation for the system of equations is shown in Figure 3. There are 29 variables and 20 equations resulting in $C_{20}^{28} = 3\,108\,105$ possible combinations of eight design variables.

Material Balance Equations

Mole Fraction Equations

1 $x_{1,1} = 1$
2 $x_{2,2} = 1$
3 $x_{1,3} + x_{2,3} = 1$
4 $x_{2,4} = 1$
5 $x_{1,5} + x_{2,5} = 1$
6 $x_{3,6} + x_{4,6} = 1$
7 $x_{1,7} + x_{2,7} + x_{3,7} + x_{4,7} = 1$

Component Balance Equations

8 $x_{1,1} * F_1 = x_{1,3} * F_3$
9 $x_{2,2} * F_2 = x_{2,3} * F_3$
10 $x_{2,4} * F_4 = x_{2,2} * F_2$
11 $x_{1,3} * F_3 = x_{1,5} * F_5$
12 $x_{2,3} * F_3 = x_{2,5} * F_5$
13 $x_{1,5} * F_5 = x_{1,7} * F_7$
14 $x_{2,5} * F_5 = x_{2,7} * F_7$
15 $x_{3,6} * F_6 = x_{3,7} * F_7$
16 $x_{4,6} * F_6 = x_{4,7} * F_7$

Flow Balance Equations

17 $F_1 + F_2 = F_3$
18 $F_2 = F_4$
19 $F_3 = F_5$
20 $F_5 + F_6 = F_7$

Four material balance equations are redundant.

Energy Balance Equations

21 $C_1 * F_1 * T_1 + C_2 * F_2 * T_2 = C_3 * F_3 * T_3$
22 $C_4 * F_4 * T_4 = C_2 * F_2 * T_2 - Q_2$
23 $C_3 * F_3 * T_3 = C_5 * F_5 * T_5 - Q_2$
24 $C_5 * F_5 * T_5 + C_6 * F_6 * T_6 = C_7 * F_7 * T_7$

Equipment Specification Equations

25 $V_1 = F_3 * t_1 / \rho_3$
26 $Q_2 = U_2 * A_2 * \Delta T_{lm2}$
27 $\Delta T_{lm2} = ((T_2 - T_3) - (T_4 - T_5)) / \log_e ((T_2 - T_3) / (T_4 - T_5))$
28 $V_3 = F_7 * t_3 / \rho_7$

Fig. 2. Equations describing mixer—exchanger—mixer system

PROPERTIES OF OCCURRENCE MATRIX REPRESENTATIONS

The occurrence matrix representation is a representation of the functional form of the system of equations. It does not indicate the difficulty in obtaining the solution of a specific equation. An equation whose equation degree of freedom is 1 (a degree one equation) has the functional form

$$f(x_i) = 0 \tag{2}$$

where x_i is variable i and f is a set of linear or nonlinear operators. It is an equation in one unknown and, if a single, real solution exists, can be solved for x_i such that the equation is reduced to

$$x_i = C \tag{3}$$

where C is a numerical constant.

A degree one equation may have multiple real solutions, imaginary solutions, or no solution may exist. In most engineering applications, a variable usually has a single, real root with physical significance.

By solving a degree one equation, the variable which appears in the equation becomes a known variable. If the numerical value of a known variable is substituted into all equations in which the known variable appears, the occurrence matrix will not contain any entries in the column representing the known variable.

A design variable is assigned a numerical value by the designer.

$$x_i = C \tag{4}$$

It is simply a chosen known variable. Once the design variables are selected, the columns representing the design variables may be removed from the occurrence matrix.

Himmelblau (1967) defines a trianglar occurrence matrix as a square matrix in which all elements on one side of the diagonal are flagged false. A triangular occurrence matrix with all main diagonal elements flagged true represents a one-to-one solution sequence. The main diagonal defines an admissable output set for the system of equations. A system of equations is determinant if an admissable output set exists for the system. Therefore, a triangular occurrence matrix is a representation of the objective function.

TABLE 1. VARIABLE ASSIGNMENTS FOR MIXER-EXCHANGER-MIXER SYSTEM

Assigned number	Variable	Assigned number	Variable
1	$x_{1,3}$	15	F_7
2	$x_{2,3}$	16	T_1
3	$x_{1,5}$	17	T_2
4	$x_{2,5}$	18	T_3
5	$x_{1,7}$	19	T_4
6	$x_{2,7}$	20	T_6
7	$x_{3,7}$	21	T_7
8	$x_{4,7}$	22	Q_2
9	F_1	23	A_2
10	F_2	24	V_1
11	F_3	25	t_1
12	F_4	26	ΔT_{lm2}
13	F_5	27	V_3
14	F_6	28	t_3

PROPERTIES OF THE LEE-CHRISTENSEN-RUDD ALGORITHM

One of the first attempts to make design variable selections on a mathematical basis was that of Lee et al (1966). This algorithm will be termed the LCR algorithm (Appendix A). The LCR algorithm has three useful properties.
1. The LCR algorithm reaches termination only for systems of equations which do not contain persistent iteration.
2. The LCR algorithm defines an admissable output set..
3. The LCR algorithm defines a one-to-one solution sequence which is the reverse of the equation elimination sequence.

THE REARRANGED MATRIX

By using the principles of the LCR algorithm, two algorithms were devised which convert the occurrence matrix representation for a system of equations to a second occurrence matrix called the *rearranged matrix*. The equation ordering algorithm and the variable group algorithm (Appendix B) convert the occurrence matrix to a form which is a nonsquare, almost triangular matrix. The equation ordering algorithm uses the third property of the LCR algorithm to order the equations into a one-to-one solution sequence which begins with the top row and proceeds consecutively to the bottom row. The variable group algorithm packs the admissable output variables in the leftmost columns of the matrix.

The rearranged matrix of the mixer-exchanger-mixer system is shown in Figure 4.

E \ V	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	f_i
3	X	X																											2
5			X	X																									2
7					X	X	X	X																					4
8	X								X		X																		3
11	X		X								X		X																4
13			X		X								X		X														4
14				X		X							X		X														4
15							X							X	X														3
17									X	X	X																		3
18										X		X																	2
19											X		X																2
20													X	X	X														3
21									X	X	X					X	X	X											6
22										X		X					X		X			X							5
23											X		X					X				X							4
24													X	X	X					X	X								5
25											X													X	X				3
26																						X	X			X			3
27																	X	X	X							X			4
28															X												X	X	3
τ_i	3	1	3	2	2	2	2	1	3	4	7	2	7	3	6	1	3	3	2	1	1	3	1	1	1	2	1	1	

Fig. 3. The occurrence matrix.

PROPERTIES OF THE REARRANGED MATRIX

To facilitate the discussion of the properties of the rearranged matrix, some definitions will be made.

New variables. The new variables of an equation in a rearranged matrix are those variables which do not appear in any row above the row containing the specified equation.

Platform variables. Platform variables of an equation are new variables of an equation in which there are more than one new variable.

Output variable. An output variable of an equation is the new variable of an equation in which there is only one new variable.

For the rearranged matrix in Figure 4, there are seven sets of platform variables:

Equation	Platform variables
(19)	11, 13
(23)	18, 22
(18)	10, 12
(22)	17, 19
(25)	24, 25
(28)	15, 27, 28
(24)	20, 21

The variables which are not platform variables are output variables of the equations in which they are new variables.

Variable groups. Variable group i is the set composed of all variables to the left of the $(i+1)^{\text{th}}$ set of platform variables. The final variable group will consist of all variables in the occurrence matrix. Group zero will be defined as the empty set (ϕ).

Subgroups of variables. Subgroup i is defined as the complement of the intersection of variable group $(i-1)$ and variable group i.

Equations associated with a subgroup. The set of equations whose new variables form a subgroup of variables is defined as the set of equations associated with the subgroup.

For the rearranged matrix of Figure 4, the following groupings are derived from the definitions:

Subgroups of variables	Subgroup equations
SG1 = 11, 13	ESG1 = 19
SG2 = 18, 22	ESG2 = 23
SG3 = 10, 12, 9, 1, 3, 4, 2	ESG3 = 18, 17, 8, 11, 5, 3
SG4 = 17, 19, 26, 23, 16	ESG4 = 22, 27, 26, 21
SG5 = 24, 25	ESG5 = 25
SG6 = 15, 27, 28, 14, 7, 6, 5, 8	ESG6 = 28, 20, 15, 14, 13, 7
SG7 = 20, 21	ESG7 = 24

PARTITIONS OF A REARRANGED MATRIX

The algorithms which produce the rearranged matrix are constructed such that the variables in a subgroup are in consecutive columns of the matrix. The equations associated with a subgroup are also in consecutive rows.

If a rearranged matrix has n subgroups; $SG1, SG2, \cdots, SGn$, the columns will be partitioned into n sets of columns whose variables are in a subgroup. The rows will be partitioned into n sets of rows whose equations are

E\V	11	13	18	22	10	12	9	1	3	4	2	17	19	26	23	16	24	25	15	27	28	14	7	6	5	8	20	21	f_i
19	X	X																											2
23	X	X	X	X																									4
18					X	X																							2
17	X				X		X																						3
8	X						X	X																					3
11	X	X						X	X																				4
5									X	X																			2
3								X			X																		2
22				X	X	X						X	X																5
27			X									X	X	X															4
26				X										X	X														3
21	X		X		X		X					X				X													6
25	X																X	X											3
28																			X	X	X								3
20		X																	X			X							3
15																			X			X	X						3
14		X								X									X					X					4
13		X							X										X						X				4
7																							X	X	X	X			4
24		X																	X			X					X	X	5
τ_j	7	7	3	3	4	2	3	3	3	2	1	3	2	2	1	1	1	1	6	1	1	3	2	2	2	1	1	1	

Fig. 4. The rearranged matrix.

associated with the n subgroups. The partitions of the example rearranged matrix are shown in Figure 5.

PLATFORM DESIGN VARIABLE COMBINATIONS

For a rearranged matrix, the columns containing the platform variables result in the deviation of the rectangular rearranged matrix from a square triangular matrix. The platform variables are contained in the first row of each main diagonal partition. If columns containing platform variables were removed from the rearranged matrix such that each equation containing platform variables had only an output variable, the reduced matrix would be a square triangular matrix with all main diagonal elements flagged true. Such a triangular matrix has an associated one-to-one solution sequence and is a minimum difficulty solution sequence.

The platform variable columns can be removed from the occurrence matrix by designating the platform variables as design variables. The elimination of the columns of any combination of platform variables which reduces the matrix to a triangular matrix is an optimal set of design variable choices.

The partitions along the main diagonal define the number of design variables associated with a subgroup. The number of design variables associated with a subgroup is the number of platform variables in the subgroup minus one.

For the example of Figure 5, there are seven sets of platform variables. Each of these sets is statistically independent which results in $2{*}2{*}2{*}2{*}2{*}3{*}2 = 192$ possible combinations of platform variables which, if selected as design variables, will result in one-to-one solution sequences.

SOLUTION MAPPING

The rearranged matrix has generated much information about the selection of design variables. However, the platform variable selections are not necessarily all of the optimal combinations which exist for a problem. If the platform design variable selections do not satisfy the needs of the designs, then a more involved study is needed.

A detailed explanation of the solution mapping principles is beyond the scope of this presentation, but such a treatment with numerous examples is available in Book (1974). This presentation will be concerned with introducing the basic principles upon which the solution mapping theory is based and with its applications. The example problem in this presentation contains at least one of every different logic loop encountered in the development of the theory. An analysis of the example should give the reader an understanding of all possible logic steps which might be encountered.

BASIC PRINCIPLES OF THE SOLUTION MAPPING THEORY

The solution mapping theory operates on rearranged matrices. The most important property of the rearranged matrix is that any submatrix composed of the top i rows is a rearranged matrix for the i equations it contains. These submatrices are also independent of the remaining portion of the rearranged matrix. This allows a submatrix of the top i rows to be analyzed as if it were a complete rearranged matrix.

A second concept which is used in the solution mapping theory is equation and subgroup dependence. Equation

E\V	11	13	18	22	10	12	9	1	3	4	2	17	19	26	23	16	24	25	15	27	28	14	7	6	5	8	20	21	f_i
19	X	X																											2
23	X	X	X	X																									4
18					X	X																							2
17	X				X		X																						3
8	X						X	X																					3
11	X	X						X	X																				4
5									X	X																			2
3								X			X																		2
22				X	X	X						X	X																5
27			X									X	X	X															4
26				X										X	X														3
21	X		X		X		X					X				X													6
25	X																X	X											3
28																			X	X	X								3
20		X																	X			X							3
15																			X			X	X						3
14		X								X									X					X					4
13		X							X										X						X				4
7																							X	X	X	X			4
24		X																	X			X					X	X	5
τ_j	7	7	3	3	4	2	3	3	3	2	1	3	2	2	1	1	1	1	6	1	1	3	2	2	2	1	1	1	

Fig. 5. Partitions of the rearranged matrix.

j (Ej) is dependent on equation i (Ei) if it is necessary to solve equation i before equation j can be solved. Equation j is directly dependent on equation i if a new variable of equation i appears in equation j. Equation j is dependent on equation i if there exists a series of n equations $[Ek, E(k+1), \cdots, E(k+n)]$ such that Ej is directly dependent on Ek which is directly dependent on $E(k+1) \cdots$ which is directly dependent on $E(k+n)$ which is directly dependent on Ei. An equation Ej which is not dependent on Ei is said to be independent of Ei. An equation Ej is totally independent if there exists no equation Ei where Ej is directly dependent on Ei. Equations will not be considered dependent upon themselves.

The properties of the rearranged matrix are such that an equation associated with subgroup i is always dependent on the equation which contains the platform variables in subgroup i.

The algorithm EDVFIX (Appendix C) determines the rows of the rearranged matrix which contain equations which are dependent on the equation in row j.

A similar concept is used to determine the order of computation of subgroups of variables. The partition of the rearranged matrix composed of the columns containing the variables in SGj and the rows containing the equations in $ESGi$ will be called partition i, j or $P(i, j)$.

A subgroup interaction matrix is formed from the partitions of a rearranged matrix. For a rearranged matrix with n subgroups, the subgroup interaction matrix is a square matrix with n rows and n columns. If $P(i, j)$ has one or more elements flagged true, then element i, j of the subgroup interaction matrix is flagged true. If $P(i, j)$ has no elements flagged true, then element i, j of the subgroup interaction matrix is flagged false.

When row j of the rearranged matrix is analyzed, the subgroup interaction matrix is developed, considering only the equations dependent on the equation in row j. Subgroup i is dependent on subgroup j if the variables in subgroup j must be calculated before it is possible to solve for the variables in subgroup i. If subgroup i is not dependent on subgroup j, the variables of subgroup i can be solved for before the variables of subgroup j, and subgroup i is said to be independent of subgroup j.

Subgroup i is directly dependent on subgroup j if element i, j of the subgroup interaction matrix is flagged true. Subgroup i is indirectly dependent on subgroup j if there exists a set of n subgroups $[SGk, SG(k+1), \cdots, SG(k+n)]$ such that subgroup i is directly dependent on subgroup k which is directly dependent on subgroup $(k+1) \cdots$ and subgroup $(k+n)$ is directly dependent on subgroup j. A subgroup j is totally independent if there exists no subgroup i (where i does not equal j) for which subgroup j is directly dependent on subgroup i.

The subgroup interaction matrix is a triangular matrix with all main diagonal elements flagged true. All subgroups are dependent on themselves, and subgroup i is independent of all subgroups j if j is greater than i. $SG1$ is always a totally independent subgroup.

Friedman and Ramirez (1973) demonstrated a directed graph (digraph) representation for a square occurrence matrix with all main diagonal elements flagged true. Each row forms a node of the graph. The main diagonal elements form the output element of each node. All non-main diagonal elements are edges of the graph. The edges are directed to all nodes in which the output element

E\V	11	13	18	22	10	12	9	1	3	4	2	17	19	26	23	16	24	25	15	27	28	14	7	6	5	8	20	21	f_i
19	X	X																											
23	X	X	X	X																									
18					X	X																							
17	X				X		X																						
8																													
11																													
5																													
3																													
22				X	X	X						X	X																
27																													
26																													
21	X		X		X		X					X				X													
25																													
28																													
20																													
15																													
14																													
13																													
7																													
24																													
τ_j																													

Fig. 6. Equation (21) with dependent equations.

appears.

For Equation (21) in the mixer-exchanger-mixer example, the dependent equations are: (22), (17), (18), (23), and (19). The partitioned matrix containing only the dependent equations is shown in Figure 6. The subgroup interaction matrix for the dependent equations is shown in Figure 7*a*, and the digraph representation for the subgroup interaction matrix is shown in Figure 7*b*.

The subgroup interaction matrix shows the direct dependencies of the subgroups. Dependence can be determined by checking all possible directly dependent paths between the subgroups. The subgroup dependency matrix will be constructed from the subgroup interaction matrix by checking all possible directly dependent paths in the subgroup interaction matrix.

If element *i, j* of the subgroup dependency matrix is flagged true, then subgroup *j* is dependent on subgroup *i*. If element *i, j* of the subgroup dependency matrix is flagged false, then subgroup *j* is independent of subgroup *i*.

The subgroup dependency matrix for the dependent equations of Equation (21) is shown in Figure 8*a*, and the digraph representation is shown in Figure 8*b*. Algorithm SDMFIX (Appendix D) constructs a row of the subgroup dependency matrix from the subgroup interaction matrix. All main diagonal elements of the subgroup interaction matrix must be flagged true.

EQUATIONS WITH DEGREES OF FREEDOM OF TWO

An equation whose functional relationship is composed of only two variables is a special class of equations:

$$f(x_a, x_b) = 0 \quad (5)$$

There are three properties which are important to this class of equations when design variables are selected:

1. Selecting both x_a and x_b as design variables results in an equation which is overspecified.
2. If x_a is in an optimal design variable combination, x_b will also appear in at least one optimal combination.
3. If x_a does not appear in any optimal combinations of design variables, then x_b will not appear in any optimal combinations.

Properties 2 and 3 can be shown by assigning a numerical value to variable x_a. The equation is now a degree one equation and can be solved for x_b. Thus, by solving the degree two equation, both x_a and x_b have numerical values. If x_a appears in an optimal combination, x_b can be chosen instead of x_a in that optimal combination.

THE SOLUTION MAPPING TECHNIQUE

Since there can be a large number of combinations of design variables that result in one-to-one solution sequences, the result of the solution mapping should be a very compact expression of large sets of combinations. A matrix is a very compact unit for expressing large sets of combinations. The solution mapping matrix expresses the combinations of design variables that result in minimum difficulty solution sequences.

The solution mapping matrix is formed from the entries in the rearranged matrix. It has been shown that the platform variable entries in the rearranged matrix express optimal combinations of design variables. These entries

V \ E	SG1	SG2	SG3	SG4			
ESG1	X						
ESG2	X	X					
ESG3	X		X				
ESG4		X	X	X			

Fig. 7a. Subgroup interaction matrix.

V \ E	SG1	SG2	SG3	SG4			
ESG1	X						
ESG2	X	X					
ESG3	X		X				
ESG4	X	X	X	X			

Fig. 8a. Subgroup dependency matrix.

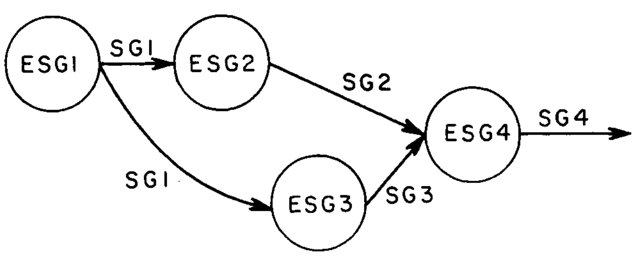

Fig. 7b. Digraph representation.

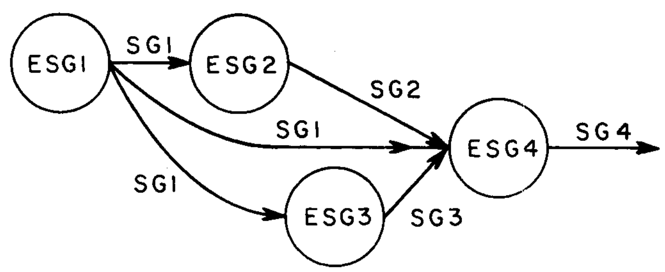

Fig. 8b. Digraph representation.

will remain in the solution mapping matrix. The criterion for determining whether or not the other entries stay in the solution mapping matrix is termed the *acceptabiliy criterion.*

THE ACCEPTABILITY CRITERION

The acceptability criterion is applied for each subgroup of each equation starting in the first row of the rearranged matrix and proceeding consecutively down the matrix.

The acceptability criterion when applied to a specific equation, will not consider any new variables introduced to the structural matrix by that equation (platform or output variables).

The acceptability criterion is first applied to the flagged entries in the first subgroup. Next, it is applied to any entries which remain in both subgroup 1 and 2. This is followed by an analysis of remaining entries in SG1, SG2, and SG3. This procedure is iterated until all entries in the equation have been analyzed.

If all remaining flagged entries in the set of subgroups being analyzed are a wholly contained subset of any optimal design variable combination, then the entries are retained in the matrix. Otherwise, the entries in the dependent subgroups of the highest numbered subgroup being analyzed are removed. (Note that a subgroup is considered dependent upon itself.)

The subgroup dependency matrix of Equation (21) and its digraph representation shown in Figure 8 will be used to demonstrate the procedure for applying the acceptability criterion. If the set of entries in subgroup 1 of the row being analyzed [Equation (21)] is a subset of any optimal combination of design variables, nothing would be done to the entries in subgroup 1 [the entry in SG1 is acceptable in Equation (21)]. If the entries in subgroup 1 failed the acceptability criterion, then the entries in subgroup 1 of Equation (26) would be removed from the solution mapping matrix. The algorithm then studies the entries which remain in subgroups 1 and 2. If the entries fulfill the acceptability criterion [those of Equation (21) do], the algorithm proceeds to the entries in SG1, SG2, and SG3. If the *SG1* and *SG2* entries had failed the acceptability criterion, then all entries in *SG1* and *SG2* must be removed. This must be done because *SG1* and *SG2* are dependent on *SG2* (Figure 8*b*). (Note that row two of the subgroup dependency matrix has entries in both columns 1 and 2, which indicates the dependent subgroups of *SG2*.) When the entries in *SG1* *SG2*, and *SG3* are analyzed, only the entries in *SG1* and *SG3* are removed upon failure of the acceptability criteria [the entries in Equation (21) fail]. (Row 3 of the subgroup dependency matrix has the first and third column flagged true; hence *SG2* is independent of *SG3*.) Entries in *SG1*, *SG2*, *SG3*, and *SG4* are removed if the entries fail the acceptability criterion when all the entries are analyzed except the new variable in Equation (21). However, these entries are acceptable, so the analysis of Equation (21) concludes with entries remaining in *SG2* and *SG4*.

The acceptability algorithm (Appendix E) determines if a set of entries are in an optimal combination by determining if the entries are replaceable for a set of platform variables which are known to be optimal. The acceptability algorithm operates on the entries in the solution mapping matrix.

THEORY OF THE ACCEPTABILITY CRITERION

The properties of degree two equations are the basis for the acceptability criterion. For the rearranged matrix

E \ V	1	2	3	f_i
1	X	X		2
2	X		X	2
t_j	2	1	1	

E \ V	1	2	3	4	f_i
1	X	X	X		3
2	X		X	X	3
τ_j	2	1	2	1	

E \ V	1	2	3	4	f_i
1	X	X			2
2	X			X	2
τ_j	2	1	0	1	

E \ V	1	2	3	4	f_i
1	X	X	X		3
2	X	X	X	X	4
τ_j	2	2	2	1	

E \ V	1	2	3	4	f_i
1	X	X	X		
2					
τ_j					

Fig. 9. Examples of some simple systems.

in Figure 9*a*, the platform variable entries in Equation (1) have two optimal combinations (*V*1 or *V*2). Since Equation (2) is a degree two equation, property two of degree two equations demonstrates that new variable 3 is replaceable for variable 1 if variable 1 appears in an optimal combination of design variables. Variable 1 is a platform variable of Equation (1) and therefore, is in an optimal combination. The acceptability criterion is fulfilled, and the entries in Equation (2) would be left in the solution mapping matrix.

For the rearranged matrix in Figure 9*b*, there are three optimal combinations of platform variables in Equation (1):

Combination	Design variables
1	*V*1, *V*2
2	*V*1, *V*3
3	*V*2, *V*3

Both entries in Equation (2) (excepting new variable 4), appear in optimal combination 2. The entries would, therefore, fulfill the acceptability criterion, and the solution mapping and the rearranged matrix are identical (no entries could be removed). Properties of degree two equations can be used to demonstrate this rational. If variable 3 is chosen as a design variable, and if the column it appears in is removed from the matrix, the result is shown in Figure 9*c*. This matrix is identical to the matrix in Figure 9*a*, and the properties of degree two equations can be applied to show that variable 4 is replaceable for variable 1 (this is conditional on variable 3 being also chosen. Similarly, choosing variable 1 as a design variable and removing it from the matrix will demonstrate that variable 4 is replaceable for variable 3. The entries are left in the solution mapping because they describe some additional optimal combinations involving variable 4.

The matrix shown in Figure 9*d* has the same three combinations of platform variables as Figure 9*c*. However, the entries in Equation (2) fail the acceptability criterion. All three entries do not appear in a single optimal combination. The equation cannot be reduced to a degree two equation such that one of the entries which remain is in an optimal combination. The entries in Equation (2) provide no information about additional optimal combinations of design variables (they fail the acceptability criterion), and the entries can be removed. The solution mapping for the rearranged matrix in Figure 9*d* is shown in Figure 9*e*.

EQUIVALENCING DEGREE TWO EQUATIONS

To make the solution mapping more compact and reduce combinational problems, equations which have only two entries remaining in the solution mapping after application of the acceptability criteria and are not platform variable equations can be equivalenced. If the entries in the equation being equivalenced are removed, then the rightmost column which had contained an entry is moved into the leftmost column. The following rules are applicable:

Right column		Left column		Result in left column	Right column		Left column		Result in left column
0	→	0	=	0	1	→	1	=	ϕ
1	→	0	=	1	1	→	ϕ	=	ϕ
0	→	1	=	1	0	→	ϕ	=	ϕ

If, when subsequent rows in the matrix are analyzed a ϕ is encountered, the subgroup containing the ϕ will fail the acceptability criteria.

For the example rearranged matrix in Figure 10*a*, the solution mapping matrix is shown in Figure 10*b* (this requires several intermediate steps). The original matrix has $C_5^7 = 21$ possible combinations of two design variables. The solution mapping contains only $C_1^3 = 3$ possible combinations, all of which are optimal.

Combination	Design variables	Actual combinations
1	*V*1, *V*9	*V*1, *V*2 or *V*1, *V*5
2	*V*1, *V*10	*V*1, *V*3 or *V*1, *V*4 or *V*1, *V*6
3	*V*9, *V*10	*V*2, *V*3 or *V*2, *V*4 or *V*2, *V*6 or *V*5, *V*3 or *V*5, *V*4 or *V*5, *V*6

Variable 9 is equivalent to either variable 2 or 5, and variable 10 is equivalent to variable 3, 4 or 6. The three combinations actually represent eleven optimal combinations. Equivalencing is not necessary to obtain a solution mapping; however, it reduces the number of entries in the solution mapping matrix.

E \ V	1	2	3	4	5	6	7	f_i
1	X	X	X					3
2			X	X				2
3		X			X			2
4				X		X		2
5			X	X		X	X	4
τ_j	1	2	3	3	1	2	1	

Fig. 10*a*. Example of equivalencing degree two equations.

E \ V	1	9	10	f_i
1	X	X	X	
τ_j				

V9 = V2 OR V5
V10 = V3 OR V4
OR V6

Fig. 10*b*. The solution mapping.

ANALYZING THE SOLUTION MAPPING MATRIX

The solution mapping algorithm (Appendix F) combines the logic of the basic principles in the form of an executive algorithm. When the algorithm is applied to the rearranged matrix for the mixer-exchanger-mixer example, the resulting solution mapping is shown in Figure 11. Variable 30 is an equivalence for variables 1, 2, 3, and 4. Figure 12*a* is the digraph representation of the solution mapping entries. The graph is disjoint and can be broken into three separate matrices.

*SG*2 and *SG*4 form a 5 by 3 matrix. One design variable is associated with each of *SG*2 and *SG*4; hence, two design variables must come from the five variables in *SG*2 and *SG*4 to obtain a minimum difficulty solution sequence. The $C_3^5 = 10$ possible combinations can easily be tried to determine that eight are optimal combinations.

E \ V	11	13	18	22	10	12	9	30				17	19	26	23	16	24	25	15	27	28	14	7	6	5	8	20	21	
19	X	X																											
23			X	X																									
18					X	X																							
17	X				X		X																						
8	X						X	X																					
22				X								X	X																
27																													
26																													
21			X									X				X													
25	X																X	X											
28																			X	X	X								
20		X																	X			X							
15																			X			X	X						
14		X						X											X					X					
13		X						X											X						X				
7																													
24																											X	X	

Fig. 11. The solution mapping.

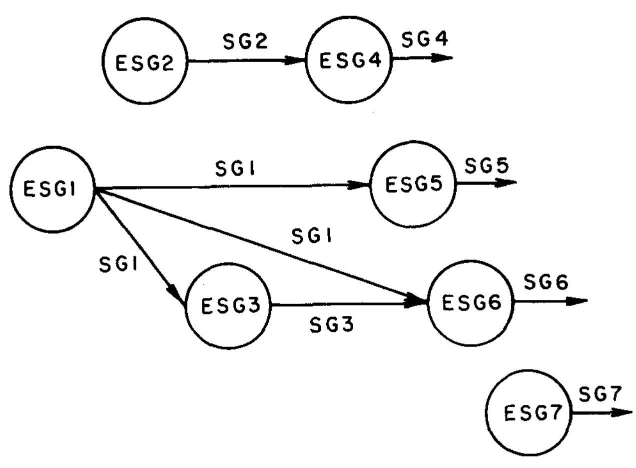

Fig. 12a. Digraph representation of solution mapping.

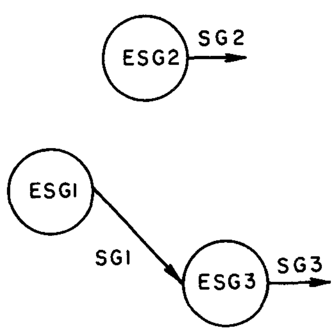

Fig. 12b. Collapsed digraph representation.

SG7 has only two variables and one equation with $C_1^2 = 2$ possible combinations. Both are optimal combinations.

SG1, SG3, SG5, and SG6 form a 15 by 10 matrix. The $C_{10}^{15} = 3\,003$ combinations can be easily analyzed by computer (with the design variable algorithm, (Appendix G). Each optimal combination which contains variable 30 is actually four combinations owing to equivalencing. There are a total of 633 optimal combinations for the matrix.

For the original occurrence matrix, there are 2*8*633 = 10 128 optimal combinations of eight design variables.

COLLAPSING THE DIGRAPH OF THE SOLUTION MAPPING

The computer analysis does not have to be performed to analyze the 3 003 possible combinations associated with SG1, SG3, SG5, and SG6. The digraph representation has SG5 and SG6 with edges directed toward no other subgroup. To obtain a minimum difficulty solution sequence, there must be at least one design variable in SG5 and at least two design variables in SG6 (the design variables associated with these two groups). The designer can choose the best combination of three design variables which satisfy SG5 and SG6 requirements and remove them from the matrix. The remaining matrix will be a 12 by 10 matrix. This matrix will have only sixty-six possible combinations remaining.

If the number of combinations remaining is too large to be analyzed, then the designer can choose optimal design variables in SG4, SG5, SG6, and SG7. The design variable columns can be removed, and a second solution mapping for the reduced system of equations can be obtained. The digraph representation for the large disjoint section will no longer contain SG5 or SG6 (Figure 12b). Some of the variables formerly in SG5 may now appear in SG1. Former SG6 variables may now appear in SG3 or SG1. Iteration of this process will collapse any digraph representation.

SUMMARY

The solution mapping theory results in more information than just the reduction of the combinatorial problems from 3 108 105 possible combinations to 3 015 possible combinations in smaller matrices. The platform design variable combinations are easily and quickly obtained and may satisfy the design variable selection requirements of the designer.

Collapsing the digraph representation allows systematic reduction of the matrix for systems in which combinatorial problems are inherent in the solution mapping matrix. This allows the designer to obtain the optimal combination best suited to a specific design problem by hand in systems which could not be handled previously by high-speed computing machines.

The structural information demonstrated in the digraph representation of the solution mapping defines the subgroups of variables in which the design variables are trapped. The single design variable associated with SG2 and the one design variable associated with SG4 are trapped in these two subgroups. The one design variable in SG2 may be transferred to SG4, but there must be at least one design variable in SG4 to obtain a minimum difficulty solution sequence. Similarly, one design variable is trapped in SG7.

The large disjoint section of the graph can be dissected to obtain more detailed information. There are only two variables in SG5, but at least one is required as a design variable. If both can be chosen as design variables, SG1 must give up its design variable. SG6 requires a minimum of two design variables but can use additional variables from either SG1 or SG3.

Once an optimal combination of design variables is chosen and their columns removed from the matrix, the solution sequence can be obtained by applying the equation ordering and variable group algorithms to the resulting square matrix. This will result in a triangular matrix. The main diagonal will contain the variable for which each equation is solved.

NOTATION

a = integer subscript of variable
A_k = heat transfer area of equipment k (area units)
b = integer subscript of variable
C = numerical constant
C_j = mean molar heat capacity of stream j (energy/mole—deg.)
C_N^M = function for obtaining total number of possible combinations
D = number of design variables or degrees of freedom
Ei = equation numbered i
$ESGi$ = set of equations associated with subgroup i
f = set of mathematical operators
f_i = equation degrees of freedom of equation i
F_j = flow rate of stream j (moles/unit time)
i = integer constant or subscript of variable
j = integer constant or subscript of variable
k = integer constant or subscript of variable
K = number of subgroups in rearranged matrix
M = number of unknown variables in occurrence matrix
n = integer constant or subscript of variable
N = number of equations in occurrence matrix
$P(i, j)$ = partition i, j of rearranged matrix or solution mapping

Q_k = heat transferred in equipment k (energy/unit time)
SGi = subgroup of variables numbered i
t_k = residence time for equipment k (unit time)
T_j = temperature of stream j (deg.)
U_k = heat transfer coefficient for equipment k (energy/area—unit time—deg.)
V_k = liquid volume in equipment k (volume)
Vi = variable numbered i
x = unknown variable
x_i = unknown variable numbered i
$x_{i,j}$ = mole fraction of component i in stream j
ΔT_{1mk} = log mean temperature difference for equipment k (deg.)
ρ_j = mean molar density of stream j (mole/volume)
τ_j = variable degrees of freedom of variable j
ϕ = empty set indicator

LITERATURE CITED

Book, Neil L., "Selection of Design and Iterative Variables in Systems of Algebraic Equations," M.S. thesis, Univ. Colo., Boulder (1974).
Friedman, F., and W. F. Ramirez, *AIChE J.*, **19**, 566 (1973).
Himmelblau, D. M., *Chem. Eng. Sci.*, **22**, 883 (1967).
Lee, W., J. H. Christensen, and D. F. Rudd, *AIChE J.*, **12**, 1105 (1966).
Ramirez, W. F., and C. R. Vestal, *Chem. Eng. Sci.*, **27**, 2243 (1972).
Rudd, D. F., and C. C. Watson, *Strategy of Process Engineering*, J. Wiley, New York (1968).
Steward, D. V., *SIAM Review*, **4**, 321 (1962).

APPENDIX A. THE LEE-CHRISTENSEN-RUDD ALGORITHM

(i). Calculate the variable degrees of freedom considering only the equations not eliminated by the LCR Algorithm. Go to (ii).
(ii). Locate a column with variable degree of freedom equal to one. If no degree one variable exists, go to (vii). Otherwise, go to (iii).
(iii). Locate the row which contains the degree one variable. Go to (iv).
(iv). Eliminate both the row and column which contain the degree one variable. Go to (v).
(v). If all the equations have been eliminated, Go to (vi). Otherwise, return to (i).
(vi). Output all columns which have not been eliminated as the optimal design variable set. Stop.
(vii). Persistent iteration encountered. Algorithm fails.

APPENDIX B. THE EQUATION ORDERING AND VARIABLE GROUP ALGORITHMS

(i). Calculate variable degrees of freedom considering only the equations not eliminated by the equation ordering Algorithm. Go to (ii).
(ii). Proceeding left to right, locate the leftmost column whose variable degree of freedom is one. If no degree one variable exists, go to (iv). Interchange the row of the matrix which contains the degree one variable with the bottom row of the matrix which has not been eliminated by the equation ordering Algorithm. Go to (iii).
(iii). Eliminate the bottom row which does not contain an eliminated equation. Go to (v).
(iv). Persistent iteration encountered. Stop.
(v). If all the equations or all equations except one have been eliminated, stop. Otherwise, go to (i).
(i). Set subgroup counter t to zero. Go to (ii).
(ii). Calculate the equation degrees of freedom considering only those columns of the matrix which have not been eliminated by the variable group algorithm. Go to (iii).
(iii). Proceeding from top to bottom, find the topmost equation whose equation degree of freedom is one. If no degree one equation exists, go to (ix). Store the degree one row outside the matrix and vacate (remove all entries in) the row of the matrix which had held the degree one equation. Go to (iv).
(iv). If the row immediately above the vacated row has an equation degree of freedom equal to zero, go to (vi). Otherwise, go to (v).
(v). Remove the entries from the row immediately above the vacated row and place the entries in the vacated row. The vacated row is now one row higher in the matrix. Go to (iv).
(vi). Put the stored degree one equation into the vacated row. Go to (vii).
(vii). Interchange the column containing the variable in the degree one equation with the leftmost column which has not been eliminated by the variable group algorithm. Go to (viii).
(viii). Eliminate the leftmost column which has not been eliminated by the variable group algorithm. Go to (xiii).
(ix). Find the topmost row which has an equation degree of freedom greater than zero. Define this value as f_{top}. Go to (x).
(x). Increment subgroup encounter t by one. vgout(t,1) = t. vgout(t,2) = the row number of the row whose equation degree of freedom is equal to f_{top}. vgout(t,3) = the column number of the leftmost column which has not been eliminated by the variable group algorithm. vgout(t,4) = $f_{top} - 1$ (the number of design and iterative variables associated with subgroup t). Go to (xi).
(xi). Interchange the f_{top} columns which have entries in the topmost equation whose equation degree of freedom is greater than zero with the f_{top} leftmost columns which have not been eliminated by the variable group algorithm. Go to (xii).
(xii). Eliminate the f_{top} leftmost columns which have not been eliminated by the variable group algorithm. Go to (xiii).
(xiii). If all the columns or all the columns except one have been eliminated by the variable group algorithm, stop. Otherwise, go to (ii).
Matrix vgout contains pointers for defining the rows and columns of subgroups of variables and equations associated with a subgroup.

Supplementary material has been deposited as Document No. 02694 with the National Auxiliary Publications Service (NAPS), c/o Microfiche Publications, 440 Park Ave. South, New York, N.Y. 10016 and may be obtained for $3.00 for microfiche or $5.00 for photocopies.

Manuscript received January 15, 1975; revision received July 11 and accepted September 16, 1975.

Cis-1,4 Polymerization of Butadiene Initiated by $CoCl_2 \cdot 4C_5H_5N\text{-}(C_2H_5)_2AlCl$

The effect of catalyst composition on the polymerization of butadiene initiated by $CoCl_2 \cdot 4Py\text{-}Et_2AlCl\text{-}H_2O$ in a batch reactor has been studied. It was found that the polymerization is first order with respect to monomer and cobalt concentrations, and both the deactivation of catalyst and the chain transfer reaction have great influence in conversion as well as molecular weight. A general conversion model was proposed, and it fits the experimental data well.

C. C. HSU
and
L. NG
Chemical Engineering Department
Queen's University, Kingston, Canada