Singh, C. P. P., and D. N. Saraf, "Simulation of High-Temperature Water-Gas Shift Reactors," *Ind. Eng. Chem., Process Des. Dev.*, **16**, No. 3, 313 (1977).
Smoot, L. D., and P. J. Smith, "Mixing and Gasification of Coal in Entrained Flow Systems," ERDA Rpt., FE-2666-f, **2**, (1979).
Sprouse, K. M., "Modeling Pulverized Coal Conversion in Entrained Flows," *AIChE J.*, **26**, 964 (1980).
Szekely, J., J. W. Evans, and H. Y. Sohn, *Gas-Solid Reactions*, Academic Press (1976).
Thring, N. W., and R. H. Essenhigh, *Chemistry of Coal Utilization-Supplementary Volume*, H. H. Lowry, Ed., Chap. 17, John Wiley & Sons (1963).
Ubhayaker, S. K., D. B. Sticker, and R. E. Gannon, "Modeling of Entrained Bed Pulverized Coal Gasifiers," *Fuel*, **56**, 281 (1977).
Wen, C. Y., "Noncatalytic Heterogeneous Solid Fluid Reaction Models," *Ind. Eng. Chem.*, **60**, No. 9, 34 (1968).
Wen, C. Y., and E. S. Lee, Eds., *Coal Conversion Technology*, Addison Wesley Publishing Co. (1979).
Wen, C. Y., and L. T. Fan, *Models for Flow Systems and Chemical Reactors*, Marcel Dekker, New York (1975).
Wen, C. Y., and T. Z. Chuang, "Entrainment Coal Gasification Modeling," *Ind. Eng. Chem. Process Des. Dev.*, **18**, 684 (1979).
Zahradnik, R. L., and R. J. Grace, "Chemistry & Physics of Entrained Coal Gasification," *Adv. Chem. Ser.*, **131**, 126 (1974).

Manuscript received February 17, 1982; revision received January 7, and accepted February 28, 1983.

Decomposition of Systems of Nonlinear Algebraic Equations

MORDECHAI SHACHAM

Department of Chemical Engineering
Ben Gurion University of the Negev
Beer Sheva, Israel

A new method for decomposing irreducible subsets, in the solution of systems of nonlinear algebraic equations, is presented. This method consists of two steps: (1) elimination of the nonlinearity from some of the equations by replacing nonlinear expressions by new variables; and (2) formulation of a problem of smaller dimension by tearing the linear subset of equations. It is shown that these modifications do not change considerably the convergence rate of the Newton-Raphson and Broyden's methods while reducing the problem's dimension. Computer time reduction up to 80% is demonstrated in the examples solved. An algorithm for elimination of nonlinear expressions, which uses Boolean matrices instead of formula manipulation, is also presented.

SCOPE

Steady-state simulation or design of chemical processes gives rise to the need of solving large sets of nonlinear algebraic equations. In the equation-oriented flowsheeting programs which are currently being developed (Shacham et al., 1981), these may be problems containing tens of thousand of linear and nonlinear equations. The simultaneous solution of such a large number of equations is very time-consuming and may be even impractical. Fortunately, large systems of equations tend to be sparse so that each equation usually involves a small number of variables. This sparsity makes it possible to decompose the large system into a set of smaller problems which can then be solved sequentially. This process (usually denoted partitioning) decomposes the system into "irreducible subsets" which cannot be further partitioned. It has been proved that it is advantageous to partition a system to irreducible subsets (Hernandez and Sargent, 1979), and there are several algorithms available for partitioning.

Irreducible subsets can further be decomposed by "tearing." This operation reduces essentially the dimension of the irreducible subset, by selecting some of the variables as "tear variables" and expressing the additional variables in the subset as explicit functions of the tear variables. There are several difficulties in applying tearing techniques to nonlinear systems, the major ones being: (1) tearing requires formula manipulation which is very expensive in terms of computer time; and (2) the smaller system generated by tearing tends to be more nonlinear and more difficult to solve (Mah, 1972).

In this paper a new tearing method for systems of nonlinear equations is presented. This method does not require formula manipulation, and it either improves or does not change the convergence rate when the Newton-Raphson or Broyden's methods are used for solution.

CONCLUSIONS AND SIGNIFICANCE

The proposed tearing method includes the following steps:

1. Some of the nonlinear equations are linearized by replacing nonlinear expressions by new variables, and by adding new nonlinear equations to the system. The nonlinear expressions for replacement are selected so that the net effect of each replacement is reduction of the number of nonlinear equations in the system. The operations of this step are carried out in a Boolean matrix so that no formula manipulation is required.
2. Tearing of the linear subset of equations is carried out by using partial Gauss elimination in this subset to express all the variables as explicit linear functions of the tear variables. The number of the tear variables is equal to the number of the remaining nonlinear equations. It has been proved that these operations do not change the convergence rate of the Newton-Raphson method. Numerical experiments have shown that they may slightly improve the convergence rate of Broyden's method.

The net effect of the tearing is reduction of the dimension of

the problem from n, where n is the total number of equations, to the number of the nonlinear equations left in the system. Since the computational effort is proportional to the square of the dimension in both the Newton-Raphson and Broyden's methods, quite substantial savings in computer time can be achieved by using this tearing method. In one of the examples we have solved, the dimension was reduced from 39 to 17 by tearing and as a result the computer time was reduced by 80%.

We expect that the proposed tearing technique together with the existing partitioning algorithms will play an important role in making solution of large systems feasible and more efficient.

INTRODUCTION

Steady-state simulation or design of chemical processes gives rise to the need for solving large sets of nonlinear algebraic equations. The need to solve such large systems became even more emphasized lately, with the development of the equation-oriented flowsheeting or simulation programs (Shacham et al., 1981). Simulating the operation of a plant using the equation-oriented approach, one may need to solve a system containing tens of thousands of linear and nonlinear equations.

A system of nonlinear algebraic equation can be written:

$$f(x) = 0 \tag{1}$$

where f is the vector of functions and x is the vector of variables. The Newton-Raphson (NR) method or one of its modifications are usually used to solve the system (Eq. 1). Two possible formulations of the NR method are:

$$x_{k+1} = x_k - J_k^{-1} f(x_k) \tag{2a}$$

and

$$J_k p_k = -f(x_k) \tag{2b}$$

where k is the iteration number, J is the matrix of partial derivatives of f with respect to x (the Jacobian matrix) and $p_k = x_{k+1} - x_k$.

Broyden's quasi-Newton method (Broyden, 1965) is frequently applied for solving chemical engineering problems. This method can be written as:

$$x_{k+1} = x_k - H_k f(x_k) \tag{3}$$

where H_k is the kth approximation to the inversed Jacobian matrix. The H matrix is estimated for the first iteration, and after that it is updated in every iteration according to Eq. 4.

$$H_{k+1} = H_k + \frac{(p_k - H_k y_k) P_k^T H_k}{p_k^T H_k y_k} \tag{4}$$

where $y_k = f(x_{k+1}) - f(x_k)$

The direct application of Eqs. 2 or 3 for large systems is very time-consuming and may be even impractical. The NR method requires calculation of n^2 partial derivatives (where n is the number of equations and unknowns) and storing and inverting the J matrix. The use of Broyden's method requires storage of the H matrix (n^2 storage locations), and the update of this matrix requires about $3n^2$ multiplication operations (Dahlquist et al., 1965, p. 252). In addition, the initial H matrix is often estimated by calculating n^2 partial derivatives using numerical perturbation at the initial point and inverting the resultant Jacobian matrix.

Fortunately systems of equations that represent the operation of a chemical plant tend to be sparse; thus, each equation will usually involve only ten or less variables. This sparsity can be exploited for more efficient solution.

Large sparse systems can usually be "partitioned" into sets of smaller problems which can then be solved sequentially, results from one problem providing data for the next. Since the storage requirements and computation effort of both the NR and Broyden's methods are proportional to n^2, partitioning a system of equations into several problems with smaller n is clearly advantageous. It was shown by Hernandez and Sargent (1979) that the computation cost will usually be minimized by partitioning the system into irreducible subsets. ("Irreducible subset" means a subset that cannot be further partitioned.)

Detailed description of the principles of "partitioning" as well as state of the art reviews of this field were recently published by Gustavson (1981), Sargent (1978), and Shacham et al. (1981).

Irreducible subsets can be further decomposed by "tearing." Using this technique, the number of variables which should be iterated on can be reduced. The subset of equations is rearranged by tearing and then solved in one of the following forms:

1. The tear variables are guessed. New values for all (including the tear) variables are calculated by solving each equation for its output variable. (Output variable of an equation is a variable for which the equation can be directly solved. When calculated, the output variable is a function of only the tear or previously calculated variables.)
2. The tear variables are guessed. Some of the equations are solved directly for the new values of the nontear variables. The residual of the rest of the equations is used to correct the value of the tear variables.

Tearing techniques have been used widely and successfully in recycle calculations in the "sequential modular" process simulation programs, but they have had only limited use in solution of systems of nonlinear equations. There are two major difficulties in applying tearing techniques for nonlinear equations:

First, tearing leads to some compounding of nonlinearities. The compounded effect may deteriorate or even prevent numerical convergence (Mah, 1972; Soylemez and Seider, 1973).

The second difficulty is associated with output variable selection. Tearing requires that some or all of the equations be rewritten so that one of the variables is an explicit function of the others. This cannot be always done. Even if it can be done it requires complicated formula manipulations which are quite expensive in computing time (Hernandez and Sargent, 1979).

In this paper a new method for tearing systems of algebraic equations is introduced. This method does not require formula manipulation, and it either improves or does not change the convergence rate.

In the next section the concept of tearing of linear subsets will be introduced. After that a method for eliminating nonlinearities from some of the equations will be introduced. Several examples will be given to demonstrate the advantages of the proposed technique.

TEARING LINEAR SUBSETS

Let us assume that the system of equations has been already partitioned into irreducible subsets and the system (Eq. 1) represents one of these subsets. Let us further assume that this system contains both linear and nonlinear equations where the total number of equations is n, out of which m are linear equations. The system of equations can be rewritten:

$$Ax = b \tag{5a}$$

$$f_1(x) = 0 \tag{5b}$$

where $f_1 \epsilon R^{n-m}$ is a subvector of f, $b \epsilon R^m$ is a vector of constants, and A is an $m \times n$ matrix of coefficients.

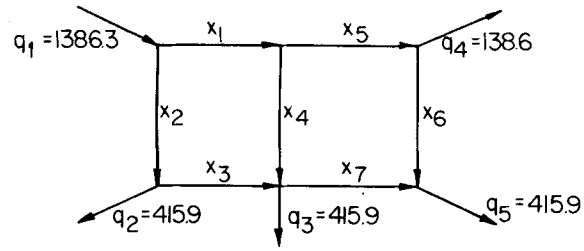

Figure 1. Simple pipeline network and representative equations.

Let us restructure the linear system (Eq. 5a) by reordering or by elimination so as to arrive at the following equivalent system:

$$[LS]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = [\bar{b}] \tag{6}$$

where $x_1 \epsilon R^m$ and $x_2 \epsilon R^{n-m}$ are subvectors of x, L is an $m \times m$ lower triangular submatrix of A with nonzero main diagonal and S is an $mx(n-m)$ submatrix of A.

If values are assigned to the components of the vector x_2 (the vector of the "tear" variables), x_1 can be calculated from Eq. 6 by substitution since L is lower triangular. The solution of Eq. 6 for x_1 can be formally written:

$$x_1 = L^{-1}(\bar{b} - Sx_2) \tag{7}$$

The system of equations consisting of Eqs. 7 and 5b is equivalent to the original system (Eq. 5), but in the modified system only the $n - m$ tear variables are independent. Thus, the dimension of the problem has been essentially reduced from n to $n - m$.

To use the NR method for solving this problem, the Jacobian matrix has to be evaluated. The Jacobian matrix for the original problem (Eq. 5) is the $n \times n$ matrix:

$$J_u = \begin{bmatrix} A \\ \cdots\cdots\cdots \\ \dfrac{\partial f_1}{\partial x_1} \vdots \dfrac{\partial f_1}{\partial x_2} \end{bmatrix} \tag{8}$$

The $(n-m)(n-m)$ Jacobian matrix of the modified problem, df_1/dx_2, can be expressed using the chain rule:

$$\frac{df_1}{dx_2} = \frac{\partial f_1}{\partial x_2} + \frac{\partial f_1}{\partial x_1}\frac{dx_1}{dx_2} \tag{9}$$

Substituting the value of x_1 from Eq. 7 into Eq. 9 gives:

$$J_M = \frac{df_1}{dx_2} = \frac{\partial f_1}{\partial x_2} - \frac{\partial f_1}{\partial x_1} L^{-1}S \tag{10}$$

At this point the algorithm for using the NR method for a system containing linear and nonlinear equation that was modified by tearing can be summarized:

1. Assume an initial value for the tear variables: $(x_{2,0})$
2. Compute $x_{1,k}$ from Eq. 7 and $f(x_k)$ from Eq. 5b (where k is the iteration number; 0 at the beginning)
3. Compute the Jacobian matrix using Eq. 10
4. Use NR iteration (Eq. 2a or 2b) to calculate the new approximation for the tear variables $x_{2,k+1}$
5. Check convergence. If converged, exit; otherwise, go back to step 2

It should be emphasized that the convergence rate of the NR method is the same for the original and the modified system. The reason for that is that the Jacobian matrix is not altered by tearing. One can bring the original Jacobian matrix (Eq. 8) into the modified form (Eq. 10) simply by Gauss-Jordan elimination. (This was shown for example by Mah, 1972.) Thus, when started from the same initial point, the same intermediate values of x_k will be obtained in the original and modified problem. (Note that using the same initial point means that x_0 has to satisfy Eq. 5a when the original formulation is used.)

2a. The modified linear subset

$$\begin{bmatrix} 1 & 1 & . & . & . & . & . \\ . & 1 & -1 & . & . & . & . \\ 1 & . & . & -1 & -1 & . & . \\ . & . & 1 & 1 & . & . & -1 \\ . & . & . & . & 1 & -1 & . \\ -2k_1x_1 & 2k_2x_2 & 2k_3x_3 & -2k_4x_4 & . & . & . \\ . & . & . & 2k_4x_4 & -2k_5x_5 & -2k_6x_6 & 2k_7x_7 \end{bmatrix}$$

2b. Jacobian matrix of the unmodified system

$$\begin{bmatrix} -2(k_1x_1 + k_2x_2 + k_3x_3 + k_4x_4) & 2k_4x_4 \\ 2k_4x_4 & -2(k_4x_4 + k_5x_5 + k_6x_6 + k_7x_7) \end{bmatrix}$$

2c. Jacobian matrix of the modified system

Figure 2. Modification of the system of equations of Figure 1 by tearing of linear subset

When selecting the tear variables, one has to make sure that the resultant L matrix is not singular. Algorithm for selection of tear variables is rectangular systems available in the literature (for example, Edie & Westerberg, 1971).

For complicated problems or for mixed systems of equations and procedures where analytical expressions for the partial derivatives are not available, derivatives can be calculated by numerical perturbation of the tear variables. Calculation of the derivatives this way requires $n - m + 1$ evaluations of the system made up of Eqs. 7 and 5b. In the unmodified system (Eq. 1), the vector of functions has to be evaluated $n + 1$ time, for calculating the derivatives numerically.

Slightly different formulations of Eqs. 7 and 10 are obtained if a complete Gauss-Jordan elimination is carried out when transforming the matrix A into the matrices L and S. In this case $L = I$ where I is the unity matrix and neither inversion of it nor multiplication by it is necessary. The computational effort associated with the use of the two formulations is about the same, but this last formulation is more convenient to use when the elements of J_M have to be expressed explicitely.

In the numerical examples that follow the Gauss-Jordan formulation is used except in example 2. In all the examples, the number of the tear variables was dictated by the number of nonlinear equations present. The tear variables were selected so, that the L matrix be nonsingular. Analytical derivatives are used in all the examples and the inversed Jacobian at $x = x_0$ is used as initial estimate for H_0 of the Broyden's method.

Example 1

Figure 1 shows a water distribution pipeline network and the system of equations that represents this network. (This example was taken from Ingels and Powels, 1964.) The parameters q_1, q_2. . .q_5 are specified inlet and outlet flow rates, and the variables x_1 to x_7 are the unknown internal flow rates. There are two laws that govern flow rates in a pipeline network: 1. the algebraic sum of flows at each node must be zero; and 2. the algebraic sum of pressure drops around any cyclic path must be zero. The application of the first law leads to linear equations (the first five equations in Figure 1); the second law gives nonlinear equations like the last two equations in Figure 1. (For the sake of simplicity and clarity, a constant value for the friction factor was used in this example.)

The variables x_1 and x_6 were selected as tear variables, and using Gauss-Jordan elimination the linear subset of equations was brought into the form of Eq. 7. The modified form of this subset is shown in Figure 2a. The Jacobian matrix for the unmodified system is shown in Figure 2b, and that of the modified system is shown in Figure 2c.

The two forms of this problem have been solved using the NR and Broyden's methods, starting with the initial estimate, Table 1. The convergence of the solution methods for this problem is

TABLE 1. INITIAL ESTIMATE AND FINAL RESULTS FOR PIPELINE NETWORK PROBLEM

	Initial Estimate	Solution
x_1	690.0	515.24
x_2	696.3	871.06
x_3	280.4	455.16
x_4	551.4	245.63
x_5	138.6	269.60
x_6	0.0	131.0
x_7	415.9	284.90

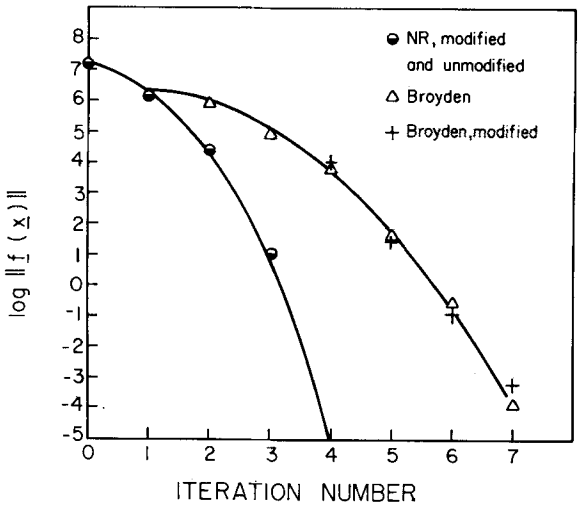

Figure 3. Convergence of solution methods for example 1.

shown in Figure 3. In this figure, the logarithm of the Euclidean norm of the residuals after each iteration is shown.

The initial estimate used satisfies the linear subset of equations. As a result, the convergence of the NR method was identical for the modified and unmodified forms of the system. For Broyden's method there were slight differences in convergence for the two types of formulations. But this difference did not show any clear trend and it is too small to be of any significance.

Example 2

A problem of larger dimension was selected to demonstrate the savings in computer time that can be achieved by tearing the linear subset.

This example is a pipeline network which is taken from Mah and Shacham (1978). The network is shown in Figure 4. All the nodes in this network are at the same elevation and all pipes are 30.4 m long and 0.152 m in diameter. The equations representing this network are similar to those of example 1, except that in this case the dependency of the friction factor on the flow rate was taken into account.

Thirty nine equations are needed for representing this system, out of which 22 are linear equations. This problem was solved in both the modified and unmodified form using the NR and Broyden's methods. Initial estimates and final flow rates as well as additional data for the problem can be found in the article by Mah and Shacham (1978). Number of iterations and computation time for the different methods are shown in Table 2. Convergence was achieved when the Euclidean norm of residuals became smaller than 10^{-4}.

Comparison of the computational times shows that modification by tearing of the linear subsystem reduced the computational effort to about 1/5th of that of the original system for both the NR and Broyden's methods. Thus the great advantage of tearing the linear subset has been clearly demonstrated.

TABLE 2. NUMBER OF ITERATIONS AND COMPUTATIONAL TIMES FOR EXAMPLE 2

Problem Formulation	Solution Method	No. of Iteration	Computation* Time(s)
Unmodified	NR	3	1.578
Modified	NR	3	0.313
Unmodified	Broyden	5	3.043
Modified	Broyden	5	0.556

* The computation time is based on a CDC Cyber 173 computer.

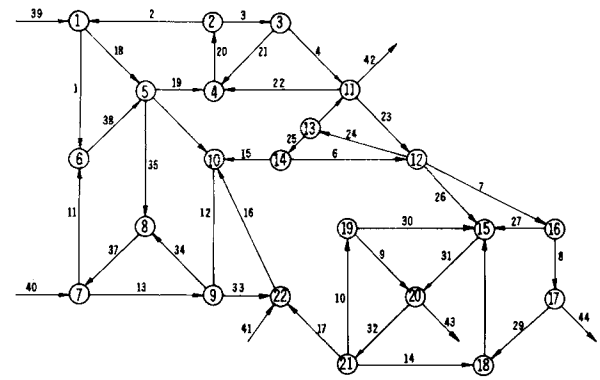

Figure 4. Pipeline network for example 2.

REDUCTION OF NONLINEAR EQUATIONS

The application of the tearing method, as presented previously, is somewhat limited, since it requires some of the equations to be completely linear. In this section we will show how the number of linear equations can be increased and the number of nonlinear equations reduced in a system without changing the convergence rate of the NR method.

A nonlinear equation can be considered in general as summation of three types of terms:

a. Constants
b. Linear terms: a constant multiplied by a variable
c. Nonlinear terms: a constant multiplied by a nonlinear expression

The following theorem will be useful in eliminating nonlinear expressions from equations:

Theorem 1

A nonlinear expression $g(x)$, which appears in one or more equations of the system can be replaced by a new variable: x_{n+1} and a new nonlinear equation: $f_{n+1} = x_{n+1} - g(x) = 0$ which is added to the system, without altering the convergence rate of the NR method for this system. (The proof for this theorem is given in Appendix A.)

In chemical engineering problems there are many examples where the same nonlinear expression appears in several equations. When carrying out steady-state material and energy balance on a CSTR, the same nonlinear expression for the generation of say, species A, may appear in several material balance equations and in the energy balance equation. In such case this nonlinear expression can be replaced by a new variable. The total number of equations will increase as a result of that; but if the number of linear equations generated is larger than the number of nonlinear equations added to the system, the net effect is the reduction of the dimension of the problem. Also, even though the original system was an irreducible subset, after modification it can often be further partitioned.

$$f_1(\underline{x}) = 0.5x_1 + x_2 + 0.5x_3 - x_6/x_7 = 0$$

$$f_2(\underline{x}) = x_3 + x_4 + 2x_5 - 2/x_7 = 0$$

$$f_3(\underline{x}) = x_1 + x_2 + x_5 - 1/x_7 = 0$$

$$f_4(\underline{x}) = -28{,}837x_1 - 139{,}009x_2 - 78{,}213x_3 + 18{,}927x_4 + 8{,}427x_5 + 13{,}492/x_7 - 10{,}690x_6/x_7 = 0$$

$$f_5(\underline{x}) = x_1 + x_2 + x_3 + x_4 + x_5 - 1 = 0$$

$$f_6(\underline{x}) = 400x_1x_4^3 - 1.7837x10^5x_3x_5 = 0$$

$$f_7(\underline{x}) = x_1x_3 - 2.6058x_2x_4 = 0$$

Figure 5. System of equations for partial oxidation of methane.

SELECTION OF NONLINEAR EXPRESSIONS FOR REPLACEMENT

Replacing one or more nonlinear terms by a new variable can be beneficial only if the number of equations that become linear by this replacement is higher than the number of new nonlinear equations generated. This simple rule can be used for selecting nonlinear expressions for replacement. The selection can be done by inspection only if the system is not too complicated. The selection procedure can be, however, carried out by the computer. This analysis is carried out by the following method.

First the system of equations is scanned for nonlinear expressions, where each new nonlinear expression different from all the previous ones is assigned a symbol ne_1, ne_2 . . . etc. The nonlinear expressions can be compared, for example, by assigning a different primary number to every variable and comparing the numerical values of the expressions. Equal numerical value means that the two expressions are equal.

After scanning the equations, an expression occurrence matrix is prepared so that it has the elements:

$$\boldsymbol{T} = (t_{ij}) = \begin{cases} 1 \text{ if expression } e_j \text{ appears in equation } i \\ 0, \text{ otherwise} \end{cases}$$

The rest of the analysis is carried out by using only this Boolean matrix, without the necessity to manipulate algebraic formulas.

Expression frequency means the number of equations in which an expression appears and equals the sum of elements in a column of the matrix $\boldsymbol{T}$. *Equation ranks* is the number of nonlinear expressions involved in an equation and equals the sum of the elements in a row of $\boldsymbol{T}$.

The occurrence matrix can be simplified by using the following rules:

1. Any row of rank 0 can be eliminated and the respective equation included in the subset of the linear equations, since this equation does not contain any nonlinear expressions.

2. A column of frequency 1 indicates that the correspondent nonlinear expression appears only in one equation, and there can not be any advantage in replacing this expression. The row and column in which this expression appears can be eliminated from the matrix and the respective equation added to the subset of nonlinear equations.

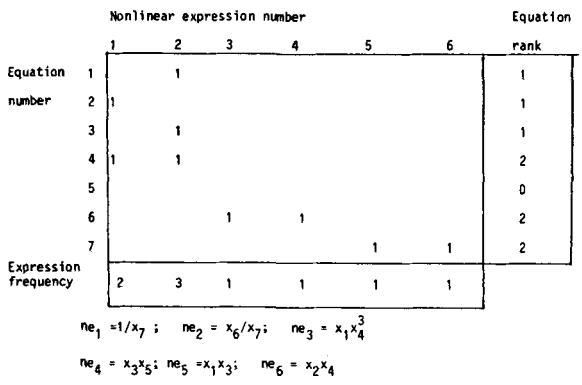

Figure 6. Expression occurence matrix for the problem of Figure 5.

$$f_1(\underline{x}) = 0.5x_1 + x_2 + 0.5x_3 - x_9 = 0$$

$$f_2(\underline{x}) = x_3 + x_4 + 2x_5 - 2x_8 = 0$$

$$f_3(\underline{x}) = x_1 + x_2 + x_5 - x_8 = 0$$

$$f_4(\underline{x}) = -28{,}837x_1 - 139{,}009x_2 - 78{,}213x_3 + 18{,}927x_4 + 8{,}425x_5 + 13{,}492x_8 - 10{,}690x_9 = 0$$

$$f_5(\underline{x}) = x_1 + x_2 + x_3 + x_4 + x_5 - 1 = 0$$

$$f_6(\underline{x}) = 400x_1x_4^3 - 1.7827x10^5\, x_3x_5 = 0$$

$$f_7(\underline{x}) = x_1x_3 - 2.6058x_2x_4 = 0$$

$$f_8(\underline{x}) = x_8 - 1/x_7 = 0$$

$$f_9(\underline{x}) = x_9 - x_6/x_7 = 0$$

Figure 7. Modified form of the system of equations from Figure 5.

After the simplification, the search for expressions that can be replaced starts. At the first stage of the search, a single column is considered at a time. The algorithm for this stage is the following:

1. Select the next column with the highest frequency.

2. Delete this column on a trial basis. If as a result the rank of more than one question becomes zero, the corresponding expression should be replaced with a new variable. Delete this column, and any row whose rank becomes zero, from the matrix. Put the corresponding equations into the linear subset. Generate a new nonlinear equation and put it into the nonlinear subset. If less than two equations have zero ranks as a result of the trial deletion, keep the column in the matrix.

3. If there is more than one row of rank 1 and there are still rows that have not been tested, go back to 1, or else start the second stage of the search.

In many practical cases, this first stage of the search will result in the elimination of all the columns and rows of the occurrence matrix. In other cases, higher order searches will have to be carried out.

The use of theorem 1 with first-order search for simplifying and solving a typical chemical engineering problem will be demonstrated in example 3. Higher-order search method will be introduced after this example.

Example 3

Figure 5 shows a system of nonlinear equations which was taken by Carnahan et al. (1969). This system represents material and energy balance in an adiabatic reactor for partial oxidation of methane. In this system there is only one linear equation. The expression occurrence matrix for this system of equations is shown in Figure 6.

Using the two rules of simplification, the matrix of Figure 6 is reduced to include only rows 1, 2, 3 and 4 and expressions 1 and 2, Eq. 5 is placed in the linear subset and Eqs. 6 and 7 are placed in the nonlinear subset.

Using the first-order search, two nonlinear expressions are replaced with new variables $x_8 = 1/x_7$, and $x_9 = x_6/x_7$, and two new equations are generated: $x_8 - 1/x_7 = 0$ and $x_9 - x_6/x_7 = 0$. The last two columns of the $\boldsymbol{T}$ matrix are deleted so that the search for replacement of the expressions stops. Figure 7 shows the modified form of the system of equations. It contains five linear and four nonlinear equations. This system can, however, be further partitioned, since x_7 appears only in equations f_8 and f_9, and x_6 appears only in f_9. The first seven equations have to be solved simultaneously, after Eq. 8 can be solved for x_8 and Eq. 9 for x_6.

TABLE 3. INITIAL ESTIMATES AND FINAL RESULTS FOR EXAMPLE 3

	Initial Estimate	Solution
x_1	0.208	0.32287
x_2	0.042	0.009224
x_3	0.048	0.046021
x_4	0.452	0.61817
x_5	0.250	0.0037165
x_6	0.340	0.57672
x_7	2.0	2.9779
x_8	0.50	0.33581
x_9	0.170	0.19367

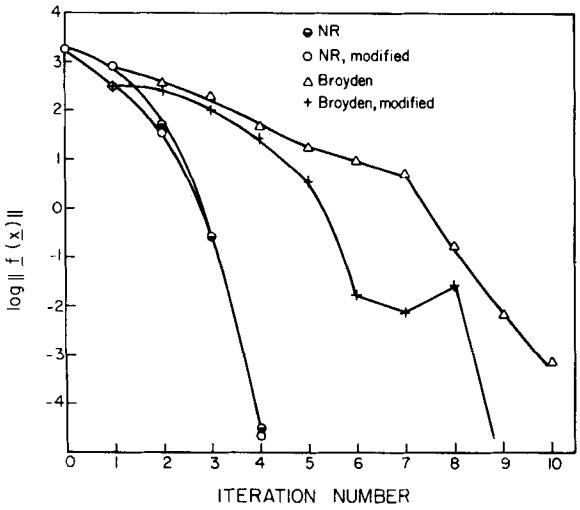

Figure 8. Convergence of the solution methods for example 3.

The variables x_8 and x_9 were selected as tear variables, and the system was decomposed by Gauss-Jordan elimination as in the previous examples. As a result of these operations, the problem's dimensionality was reduced from the original seven to two.

The modified and unmodified forms of this problem have been solved using the NR and Broyden's methods, starting from the initial estimate, Table 3. The convergence of the solution methods for this problem is shown in Figure 8, and the values of the variables at the solution are shown in Table 3.

Figure 8 shows that the convergence is slightly better for the modified form when the NR method is used and better when the modified form of Broyden's method is used.

Thus, tearing a system of nonlinear equations using the proposed method may even improve the convergence instead of deteriorating it. Such deterioration was observed when using other tearing algorithms.

The change in the convergence rate of the NR method for the modified system is a result of partitioning, since this operation does not preserve the convergence rate like the other operations do.

HIGH-ORDER SEARCH FOR REPLACEMENT OF NONLINEAR EXPRESSIONS

Sometimes the use of the simplification rules and first-order searches cannot eliminate all the columns from the expression occurrence matrix. In such cases, the use of higher-order searches (starting with second order) is necessary. The algorithm for an ith-order search is the following:

1. Select i columns with the highest frequency.
2. Delete these columns on a trial basis. If, as a result, the rank of more than i equations become zero, the corresponding expressions should be replaced with new variables. Delete this group of columns and any row whose rank became zero from the matrix. Put the corresponding equations into the linear subset. Generate

$$f_1(\underline{x}) = q_1 - x_1 - k_1x_1x_2/S^2 = 0$$

$$f_2(\underline{x}) = q_2 - x_2 - k_1x_1x_2/S^2 - k_2x_2x_3/S^2 = 0$$

$$f_3(\underline{x}) = q_3 - x_3 - 2x(k_1x_1x_2/S^2 - k_2x_2x_3/S^2) - k_2x_3x_6/S^2 = 0$$

$$f_4(\underline{x}) = q_3 - x_4 + 2k_2x_2x_3/S^2 = 0$$

$$f_5(\underline{x}) = q_5 - x_5 + 1.5k_3x_3x_6/S^2 = 0$$

$$f_6(\underline{x}) = q_6 - x_6 + 2k_2x_2x_3/S^2 - 0.5k_3x_3x_6/S^2 = 0$$

$$S = x_1 + x_2 + x_3 + x_4 + x_5 + x_6$$

Figure 9. System of equations for example 4.

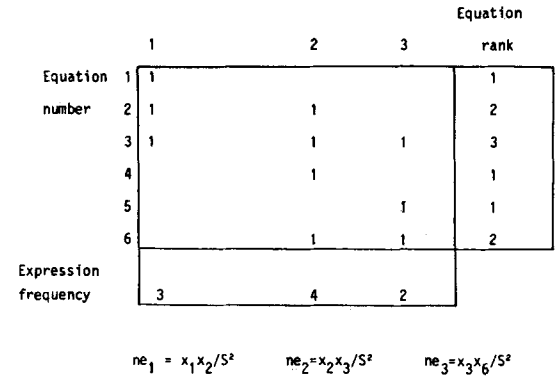

Figure 10. Expression occurrence matrix for the problem of Figure 9.

new nonlinear equations according to the number of expressions that have been replaced and put them in the nonlinear subset. Return to the first-order search. If less than $i \pm 1$ equations have zero ranks as a result of the trial deletion, keep this group of columns in the matrix.

3. Delete the column with the lowest frequency from the search group and replace it with the column of highest frequency that has not been included in the search group yet. Go back to 1. If there are no more columns which have not been included already in the search group, start the i + first stage of the search.

The search for replacement of expressions stops if one of the following conditions is satisfied:

1. All the columns have been eliminated from the matrix.
2. The minimal rank is higher than the maximal frequency.

This algorithm for selecting nonlinear expressions for replacement was tested with several practical problems. In all the cases, the search stopped after the first or the second stage of the search when all the columns have been eliminated from the occurrence matrix.

The use of higher order search will be demonstrated with the following example.

Example 4

Figure 9 shows a system of nonlinear equations which represents material balance on a chemical reactor (The Williams-Otto Process). $q_1, q_1, \ldots q_6$ are specified inlet flow rates; k_1, k_2 and k_3 are specified reaction rate coefficients; and x_1, $x_2 \ldots x_6$ are the unknown outlet flow rates.

All the equations in this system are nonlinear. The expression occurrence matrix for the system is shown in Figure 10.

The matrix cannot be simplified since there are no rows of rank 0 or columns of frequency 1. Deleting a single column at a time zeroes only one equation rank; thus, first-order search cannot simplify the matrix. Deleting columns 1 and 2 together zeroes the ranks of Eqs. 1, 2, and 4. These columns can be deleted and new variables, $x_7 = x_1x_2/S^2$ and $x_8 = x_2x_3/S^2$, are defined together with the corresponding nonlinear equations.

After this step, only column 3 and rows 3, 5 and 6 remain in the matrix. Deleting colum 3 zeroes the ranks of all the remaining rows.

A new variable $x_9 = x_8x_6/S^2$ can be defined and the search can be stopped.

In the final form, the system of equations contains six linear and three nonlinear equations.

NOTATION

A = matrix of coefficients
b = vector of constants
f = vector of functions
g = vector of nonlinear expressions
H = approximation to inversed Jacobian matrix
I = unity matrix
J = matrix of partial derivatives (Jacobian matrix)
k = iteration number
L = lower triangular matrix
m = number of linear equations in an irreducible subset
n = total number of equations in an irreducible subset
ne = nonlinear expression
q = inlet or outlet flowrate
T = expression occurrence matrix
x = vector of variables
x_1, x_2 = subvectors of x

APPENDIX: PROOF FOR THEOREM 1

Theorem 1. A nonlinear expression $g(x)$ which appears in one or more equations of the system can be replaced by a new variable, x_n+1, and a new nonlinear equation, $f_{n+1} = x_{n+1} - g(x) = 0$, which is added to the system, without altering the convergence rate of the NR method for this system.

Proof. The elements of p_k as calculated from Eq. 2b are identical for the original and the modified system. Assume, without loss of generality, that one of the equations, say $f_1(x)$, is a sum of two terms, $f_1(x) = g_1(x) + g_2(2)$. Equation 2b written for this system has the following form:

$$\begin{bmatrix} \frac{\partial g_1}{\partial x_1} + \frac{\partial g_2}{\partial x_1} & \frac{\partial g_1}{\partial x_2} + \frac{\partial g_2}{\partial x_2} & \cdots \frac{\partial g_1}{\partial x_n} + \frac{\partial g_2}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \frac{\partial f_2}{\partial x_n} \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \cdot & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x = x_k}$$

$$\begin{bmatrix} P_{1,k} \\ P_{2,k} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_{n,k} \end{bmatrix} = - \begin{bmatrix} g_1(x_k) + g_2(x_k) \\ f_2(x_k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_n(x_k) \end{bmatrix} \qquad (A1)$$

Now replace the function $g_2(x)$ by a new variable x_{n+1} and add a new equation to the system: $f_{n+1}(x) = x_{n+1} - g_2(x)$. Writing Eq. 2b for the modified system gives:

$$\begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \frac{\partial g_1}{\partial x_2} & \cdots \frac{\partial g_1}{\partial x_n} & 1 \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \frac{\partial f_2}{\partial x_n} & 0 \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots \frac{\partial f_n}{\partial x_n} & 0 \\ \frac{\partial g_2}{\partial x_1} & \frac{\partial g_2}{\partial x_2} & \cdots \frac{\partial g_n}{\partial x_n} & 1 \end{bmatrix}_{x = x_k}$$

$$\begin{bmatrix} P_{1,k} \\ P_{2,k} \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ P_{n,k} \\ P_{n+1,k} \end{bmatrix}_{c = x_k} = - \begin{bmatrix} g_1(x_k) + x_{n+1} \\ f_2(x_k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ f_n(x_k) \\ x_{n+1}, - g_2(x_k) \end{bmatrix} \qquad (A2)$$

Equation A2 can be brought back into the form of Eq. A1 using elementary transformations. The last row in Eq. A2 can be multiplied by -1 and added to the first row. After this transformation, the first row in Eq. A2 becomes identical to the first row in Eq. A1 except that there will be a zero term in column $n + 1$. The subset of Eq. A2 which contains the first n rows and columns can be solved independently, since the variable x_{n+1} appears only in the last equation. This subset of Eq. A2 is identical to Eq. A1, thus, the P_k values calculated by using these two equations will also be identical.

LITERATURE CITED

Broyden, C. G., "A Class of Methods for Solving Nonlinear Simultaneous Equations," *Math. Comp.*, **19**, 577 (1965).

Carnahan, R., H. A. Luther, and J. O. Wilkes, "Applied Numerical Methods," 321, Wiley, New York (1969).

Edie, F. C., and A. W. Westerberg, "Computer-Aided Design. Part 3: Decision Variable Selection to Avoid Hidden Singularities in Resulting Recycle Calculations," *Chem. Eng. J.*, **2**, 114 (1971).

Dahlquist, G., A. Bjork, and N. Anderson, "Numerical Methods," 252., Prentice-Hall, Englewood Cliffs (1974).

Gustavson, F. A., "Some Aspects of Computation with Sparse Matrices," 77, R. S. H. Mah and W. D. Seider, Eds., *Foundations of Computer-Aided Chemical Process Design*, 1, Engineering Foundation, New York (1981).

Hernandez, R., and R. W. H. Sargent, "A New Algorithm for Process Flowsheeting," *Comput. Chem. Eng.*, **3**, 373 (1979).

Ingels, D. M., and J. E. Powers, "Analysis of Pipeline Networks," *Chem. Eng. Prog.*, **60**(2), 65 (1964).

Mah, R. S. H., "Structural Decomposition in Chemical Engineering Computation," 72nd AIChE Nat. Meeting, St. Louis (May 21-26, 1972).
Mah, R. S. H., and M. Shacham, "Pipeline Network Design and Synthesis," *Adv. Chem. Eng.*, **10**, 125 (1978).
Sargent, R. W. H., "The Decomposition of Systems of Procedures and Algebraic Equations," G. A. Watson, Ed., *Numerical Analysis-Proceedings*, Biennial Conf., Dundee, Lecture Notes in Mathematics 630, 158, Springer-Verlag, Berlin (1978).
Shacham, M., S. Macchietto, L. F. Stutzman, and P. Babcock, "Equation Oriented Process Flowsheeting," *Comp. Chem. Eng.*, **6**(2), 79 (1982).
Soylemez, S., and W. D. Seider, "A New Technique for Precedence-Ordering Chemical Process Equation Sets," *AIChE J.*, **19**, 934 (1973).

Manuscript received April 2, 1982; revision received December 6, and accepted December 14, 1982.

Fate of Solids Fed Pneumatically through a Jet into a Fluidized Bed

IFIYENIA KECECIOGLU, WEN-CHING YANG, and D. L. KEAIRNS

Westinghouse Research and Development Center
Pittsburgh, PA 15235

Solid tracer particles were fed pneumatically through a jet into a fluidized bed to simulate the feeding of solids via a pneumatic transport line into a fluidized-bed reactor operating in the slugging-bed mode. The fluidized bed was defluidized instantaneously at different times after the initiation of the tracer particle injection. The bed was then sampled layer by layer to provide the radial and axial concentration profiles of the tracer. Regular and high-speed movies (1,000 frames per second) were taken to study the operation of the fluidized bed and the phenomena of the gas-solid two-phase jet.

Experimental results on solid mixing, jet constriction and slugging frequencies, slugging bed height, slug length, jet penetration, and jet half-angle at three nominal jet velocities of 52, 37, and 25 m/s and corresponding solids loadings are presented. Additional experimental results on jet constriction and slugging frequencies, and slug volume (axial slug size) obtained for a wider range of jet velocities confirm the hydrodynamic trends observed during the tracer particle injection experiments. The results indicate that solids mixing increases, and well-mixed conditions are reached earlier, with an increase in jet injection velocity. The obtained mixing times were correlated successfully in terms of the excess gas velocity. The experimental data on jet penetration and slug motion were satisfactorily correlated by modified versions of existing theoretical relations. The modifications included the effect of the injected solids on jet penetration and jet half-angle and also the effect of our semicircular column geometry and single wall-slug configuration on the observed slug motion.

SCOPE

Fluidized-bed reactors have been widely acclaimed in the process industry for their advantageous characteristics of good mixing, high rates of heat and mass transfer, mechanical robustness, and capability of continuous operation. Research seeking to harness these favorable reactor characteristics to the efficient and environmentally acceptable processing of fossil fuels (e.g., coal combustion, coal and oil shale gasification) has been undertaken intensively in recent years.

In gas-solid fluidized beds, the gaseous reactant is introduced uniformly through a distributor plate, while the solids are introduced via individual feed points. As a result, one has to rely on the characteristics of the flow field to promote solids mixing within the reactor.

A commonly employed method of feeding solids into a fluidized bed is pneumatic injection of the solids through a two-phase gas-solid jet. If the gas-solid reaction is fast, undesirable, localized, nonuniform temperature and concentration conditions may appear around the solids feeding points in the fluidized bed. Such an occurrence may compromise the performance of the fluidized-bed reactor. Therefore, from the points of view of reactor design and selection of optimum reactor operating conditions, knowledge of the particle history inside the fluidized bed is essential.

The fate of solids fed pneumatically through a jet into a fluidized bed depends on the particle size and density, the jet velocity, the solid loading in the jet, the bed height, the bed diameter, the jet nozzle design, and the fluidization conditions in the bed. Despite the importance of the fate of pneumatically fed solids to reactor performance, the complex phenomena affecting solids mixing have not as yet been studied systematically.

A simple but effective technique was developed in this study to determine the extent of solids mixing and the deviation from the ideal flow patterns in different regions within a fluidized-bed reactor by sampling the injected solid tracer particles upon defluidization.

Experimental results on solids mixing and the related flow field phenomena in a slugging fluidized bed are presented at three nominal jet velocities of 52, 37, and 25 m/s. The experiments were performed at atmospheric pressure and ambient temperature.