ChE lecture

CLOSE ENCOUNTERS OF A SPARSE KIND*

ARTHUR W. WESTERBERG
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

The goal is to present methods for solving sets of nonlinear algebraic equations, e.g.:

$$f_1(x_1,x_2,\ldots.x_n) = 0$$
$$f_2(x_1,x_2,\ldots.x_n) = 0$$
$$\vdots$$
$$f_m(x_1x_2,\ldots.x_n) = 0$$

Two cases are of interest

Square Case $n = m$
Nonsquare Case $n > m$

For the latter case n-m variables are degrees of freedom for the problem. Their values must be supplied from elsewhere.

We are interested in the case where the equations are sparse, i.e. each equation explicitly contains only a few of the variables. The literature contains two general approaches for solving:

- Tearing (with convergence acceleration).
- Newton-Raphson with sparse matrix methods.

Therefore we have two problems, $n = m$ and $n > m$, and two approaches to consider.

We shall first present an example problem, a flash calculation. For it we shall apply the essential ideas for solving by the tearing method and then by using the Newton-Raphson method.

*This paper was presented as the first annual tutorial lecture for the ASEE meeting held in Vancouver, British Columbia, in June, 1978. The material appears in more detail in the book Process Flowsheeting (Westerberg, Hutchison, Motard and Winter (1977)). It has been taught as part of an elective course on computer-aided process design to both seniors and graduate students in chemical engineering.

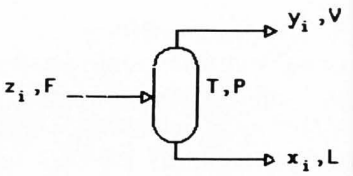


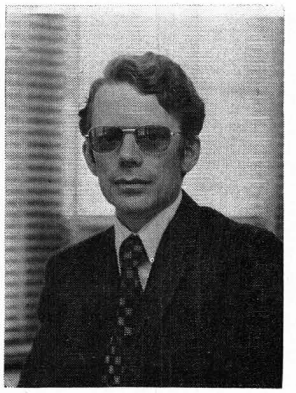
Fig. 1 A Flash Unit

EXAMPLE PROBLEM

A diagram for a flash unit appears in Figure 1. The model is as follows, where the physical properties are treated very simply.

Material Balance

$$y_iV + x_iL = z_iF \qquad i = 1,2,\ldots c \qquad (1)$$
$$V + L = F \qquad (2)$$



Arthur W. Westerberg received his degrees in chemical engineering at Minnesota, Princeton, and Imperial College, London. He then joined Control Data Corporation in their process control division for two years. In 1967 he joined the University of Florida where he remained for nine years. In 1976 he joined the faculty at Carnegie-Mellon University. He was Director of the Design Research Center from 1978 to 1980 and just became Head of Chemical Engineering this January.

Equilibrium

$$y_i = K_i x_i \quad (3)$$

Physical Properties

$$PK_i = P_i^o \quad \text{(Raoult's Law)} \quad (4)$$

Antoine's Equation for Vapor Pressure

$$P_i^o = 10^{(A_i - B_i)/(C_i + T)} \quad (5)$$

Other

$$\Sigma x_i - \Sigma y_i = 0 \quad (6)$$

$$\Sigma z_i = 1 \quad (7)$$

The problem we address next is to develop a solution procedure to solve these equations. We note that, for c = 3 components, the number of equations is 15 (Equation (1) 3 equations; (2) 1; (3) 3; (4) 3; (5) 3; (6) 1; (7) 1; for a total of m = 15 equations). The variables for the problem are:

$y_i, x_i, z_i, K_i, P_i, i = 1,2,3$ = 15 variables

V,L,F,T,P = 5 variables

For a total of n = 20 variables. Thus there are five variables in excess of the number of equations or n – m = 5 degrees of freedom for the problem.

Deriving a Solution Procedure for Square Case

We need first to reduce our problem to 15 equations in 15 unknowns. To do this we add five specifications. We shall set values for the five variables z_i, z_2,P,T, and F, a set of specifications corresponding to that for a so-called isothermal flash calculation. Figure 2 is an "incidence matrix" (or "occurrence matrix") for the 15 equations in the remaining 15 variables.

Our first task is to partition the equations if possible. For a sparse, square set of equations one will often find that a subset of the equations can be found which involves n_1 equations in precisely n_1 unknowns. These n_1 equations in n_1 unknowns can be solved first and by themselves. These n_1 equations and n_1 variables may then be deleted from the problem, leaving us with a set of $n-n_1$ equations in $n-n_1$ unknowns. We may again attempt to locate within this reduced equation set a further subset of n_2 equation in precisely n_2 unknown. Again, these may be solved next and by themselves. We can repeat this activity until no further reduction in the problem is possible. This task is known as partitioning the equations, and the order in which we then solve these partitions is called a precedence order. Solving a large problem as a sequence of small problems is clearly much easier to do.

The steps involved in partitioning a set of

We need first to reduce our problem to 15 equations in 15 unknowns. To do this we add five specifications. We shall set values for the five variables z_i, z_2,P,T, and F, a set of specifications corresponding to that for a so-called isothermal flash calculation.

equations are as follows. First, we assign a unique variable to each equation. This assignment is called an output assignment for the equations, and the circled variables in Figure 2 are such an assignment. For a small problem finding an output assignment can be done quickly by hand. For larger problems one may reformulate the problem as an assignment problem in linear programming and solve using the very efficient algorithms available for that particular problem type. Note that we have required an explicit output assignment here, that is, one in which each assigned variable

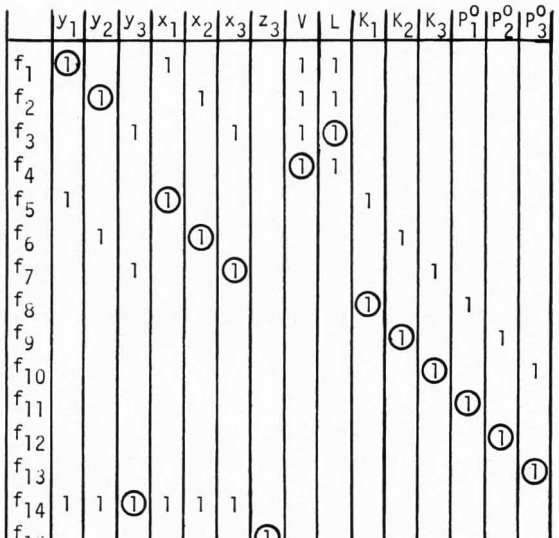


FIGURE 2. Incidence Matrix for Square Case. Circled incidences indicate an "output assignment" for the equations.

must appear explicitly within the equation to which it is assigned.

If an explicit output assignment has been found for the equations, then one can use path tracing algorithms to find the partitions and the precedence order for the partitions. To implement a path tracing algorithm we first establish a precursor list for the variables in our problem.

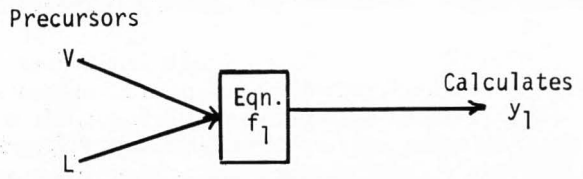


FIGURE 3. Information Flow for an Equation.

Figure 3 illustrates the essential ideas behind this precursor list. Figure 4 is a completed precursor list for the flash problem. Note that each equation is identified by its assigned output variable. The precursors to that variable are the other variables occurring in the equation. For example, equation f_1 has an assigned output variable y_1. The other three variables appearing in equation f_1 are x_1, V and L. In order to calculate y_1 using equation f_1 we would need to have values x_1, V and L. With a precursor list for our equations we may readily execute a path tracing algorithm to find the groups of equations which form the partitions for our equation set. We proceed as follows.

Variables	y_1	y_2	y_3	x_1	x_2	x_3	z_3	V	L	K_1	K_2	K_3	P_1^o	P_2^o	P_3^o
Precursors	x_1	x_2	y_1	y_1	y_2	y_3	-	L	y_3	P_L^o	P_2^o	P_3^o	-	-	-
	V	V	y_2	K_L	K_2	K_3			x_3						
	L	L	x_1						V						
			x_2												
			x_3												

FIGURE 4. Precursor List for Flash Problem.

Path tracing involves tracing through the precursor lists, starting with any variable.

y_1 has precursor x_1 which has precursor y_1

or

$y_1 \leftarrow x_1 \leftarrow y_1$

Clearly y_1 and x_1 are in an "information" loop; i.e., they each require the other to be calculated. We thus merge them and their precursor lists and treat them as a single item on our list. Thus

$(y_1x_1) \leftarrow V \leftarrow L \leftarrow y_3 \leftarrow y_1$

Clearly $(y_1x_1)VLy_3$ are also in a loop.

Continuing

$(y_1x_1VLy_3) \leftarrow K_1 \leftarrow P_1^o$ no precursor.

Thus P_1^o may obviously be calculated first and by itself. It is deleted from all lists and placed first on our list of partitions. If we were to continue we would ultimately find the partitioning and precedence order implied by the reordered incidence matrix in Figure 5. We note only the last partition involves more than one equation in one unknown. For each partition we must then derive a solution procedure. The last eight equations in eight unknowns require simultaneous solution.

Tearing Approach

Tearing is used to locate a solution procedure which requires one to guess only a few (say t) of the variables and then use n-t of the equations to calculate directly the values of the remaining n-t variables in terms of those guesses. The t unused equations may then be used as error functions which should be zero if our guesses are correct. If not zero we need to reguess. If we assume that each variable may readily be calculated in terms of the remaining variables appearing in the equation, then the following algorithm may be used.

Algorithm (A quick and dirty, but effective, algorithm.)

1. Select as the next equation the one containing the fewest new variables. Repeat until all equations selected.
2. Select strategically among new variables introduced with each equation one to be calculated by it. If no new variables are introduced, select the latest variable introduced earlier which remains unassigned and which has a cause/effect relationship to new equation.

Applying this algorithm gives the following result:

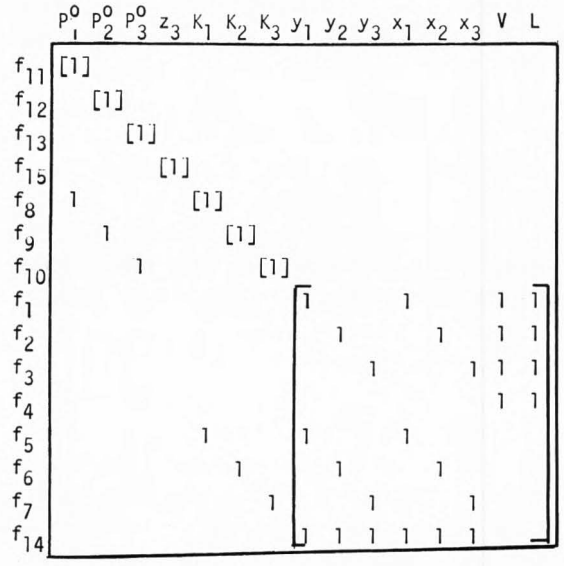
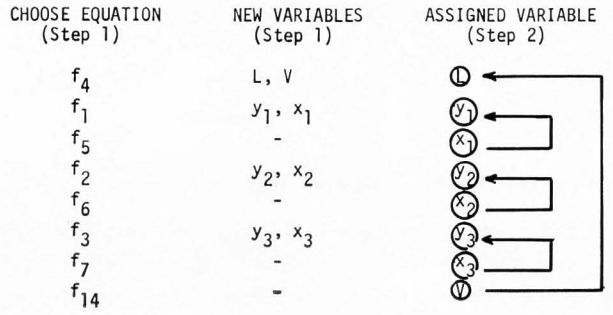
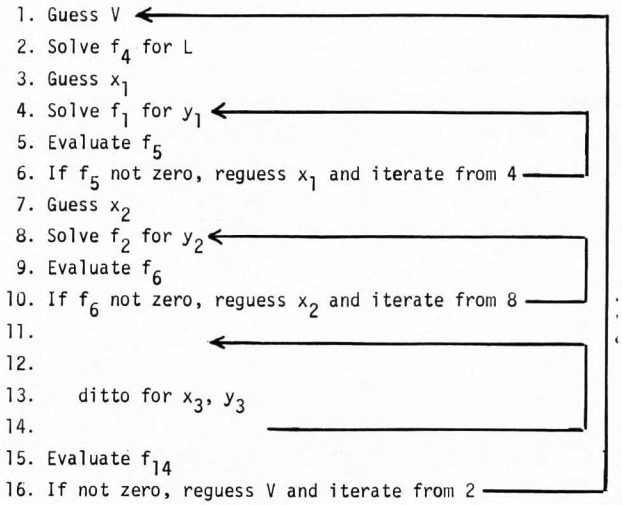


FIGURE 5. Partitions and their Precedence Order for Flash Problem.



which implies the following solution algorithm.



One could also solve by saving the evaluation of f_5 (Step 5), f_6 (Step 9) and f_7 (Step 13) until the end and evaluating all three with f_{14}, then reguessing V, x_1, x_2 and x_3 simultaneously.

Note that equations (f_1, f_5) are linear in x_1 and y_1 and could be solved directly as a pair of linear equations. Thus the iteration in Step 6 is not necessary. (f_2, f_6) and (f_3, f_7) are also linear in (x_2, y_2) and (x_3, y_3), respectively.

Newton-Raphson With Sparse Matrix Methods

The Newton-Raphson method is to linearize our equations about a current guess:

$$f_1(x_1 + \Delta x_1, x_2 + \Delta x_2, \ldots x_n + \Delta x_n) \approx$$
$$f_1(x_1, x_2, \ldots x_n) + \frac{\partial f_1}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial f_1}{\partial x_n}\Delta x_n$$
$$f_2(x_1 + \Delta x_1, \ldots x_n + \Delta x_n) \approx$$
$$f_2(x_1, x_2, \ldots x_n) + \frac{\partial f_2}{\partial x_1}\Delta x_1 + \ldots + \frac{\partial f_2}{\partial x_n}\Delta x_n$$
$$\vdots$$
etc

$$\mathbf{f}(\mathbf{x} + \Delta\mathbf{x}) \approx \mathbf{f}(\mathbf{x}) + \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}^T}\right)_{\mathbf{x}} \Delta\mathbf{x}$$

The approach is to find the change required in $\mathbf{x}$ such that the linearized equations become zero at $\mathbf{f}(\mathbf{x} + \Delta\mathbf{x})$. We thus get the Newton-Raphson equations:

$$\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}^T}\right)_{\mathbf{x}} \Delta\mathbf{x} = -\mathbf{f}(\mathbf{x})$$

which are a set of linear equations. The solution algorithm is as follows:

Algorithm

1. Guess $\mathbf{x}$
2. Evaluate $\mathbf{f}(\mathbf{x})$ (which should be zero at solution)
3. Evaluate Jacobian matrix $\left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}^T}\right)$ at $\mathbf{x}$.
4. Solve NR eqns for $\Delta\mathbf{x}$
5. Let $\mathbf{x}$ be replaced by $\mathbf{x} + \Delta\mathbf{x}$ and iterate from (2) until $\mathbf{f}(\mathbf{x})$ is very small.

The Jacobian matrix has most coefficients equal to zero; it is very sparse. It should be solved using sparse matrix methods. For the flash equations the Jacobian matrix is as follows.

$$\frac{\partial \underline{f}}{\partial \underline{x}^T} = \begin{array}{c} \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_{14} \end{array} \begin{array}{c} \begin{array}{cccccccc} y_1 & y_2 & y_3 & x_1 & x_2 & x_3 & L & V \end{array} \\ \begin{bmatrix} V & & & L & & & x_1 & y_1 \\ & V & & & L & & x_2 & y_2 \\ & & V & & & L & x_3 & y_3 \\ & & & & & & 1 & 1 \\ 1 & & & -K_1 & & & & \\ & 1 & & & -K_2 & & & \\ & & 1 & & & -K_3 & & \\ -1 & -1 & -1 & 1 & 1 & 1 & & \end{bmatrix} \end{array}$$

which is then evaluated in each iteration at the current values of all the variables.

The essential idea in using sparse matrix methods is to develop a pivot sequence for solving the linear equations which creates the fewest new nonzero elements in our coefficient matrix as we solve the equations. Each nonzero coefficient requires added work in terms of multiplies and adds which we desire to avoid. A very quick and dirty algorithm for finding a better pivot sequence is as follows.

Algorithm (one of many)

1. Select row with fewest entries
2. Select in that row, column with fewest entries
3. If selected element not almost zero, select as next pivot and proceed.

Figure 6 shows this pivot selection algorithm applied to our flash equations for the first two of

It is very easy to solve several hundred to a few thousand sparse linear equations.

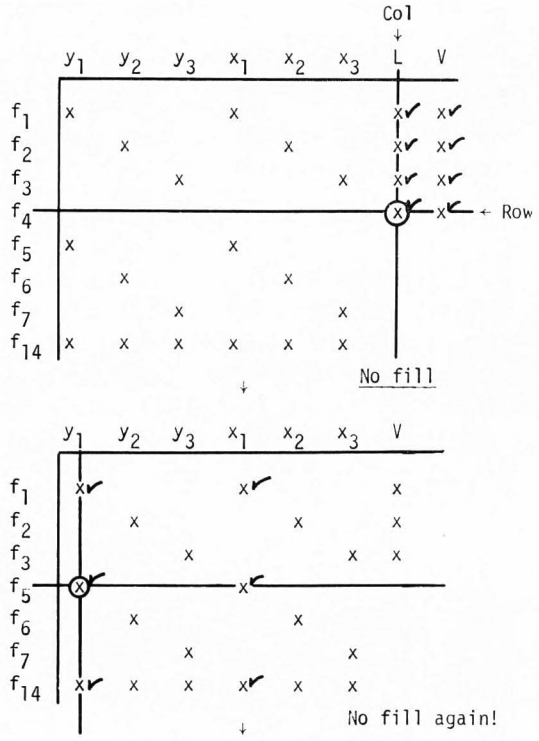


FIGURE 6. The First Two Pivot Selection and Elimination Steps for Flash Problem.

the eight pivots selected. The check marks indicate coefficients which are altered when using the selected pivot to eliminate the nonzeros in the pivot column during Gaussian elimination. The first pivot selected is in the row for f_4 and in the column for L. Gaussian elimination then requires us to zero out all other nonzeros in the pivot column. This elimination is done by subtracting the appropriate multiple of the pivot row from all rows having a nonzero in the pivot column. Only those elements checked will be changed by this step, and hopefully only a few of them will be changed from zero to nonzero, i.e. "filled". Here all changed elements are already nonzero and no nonzeros "fill" in either step. If we continue we would discover only one nonzero fills by the time all eight pivot/elimination steps are completed. One could count the numbers of multiplications and additions needed to do the forward elimination here and discover the numbers are very small relative to handling all the zero valued coefficients too.

It is very easy to solve several hundred to a few thousand sparse linear equations.

Solving Nonsquare Equation Sets

A simple approach is to skip all partitioning and precedence ordering steps. Then the left over variables (those never assigned or pivoted) become the decision variables for the problem.

Other Topics

We have so far presented only a part of the material needed to solve sparse nonlinear equations. We should also present material on each of the following topics.

1. Methods to reguess tear variables, i.e.:
   - Secant Method (1 variable)
   - Generalized Secant Method (n variables)
   - Broyden's Method (n variables)
2. What to do if $||f||$ does not decrease after a step is taken in the N-R method, e.g.
   - take a smaller step in N-R direction.
   - use $||f||^2 = \sum_{i=1}^{n} f_i^2$ as an objective and seek a new direction more in the direction of steepest descent for this objective while also taking a shorter step (Levenberg/Marquardt algorithm)
   - use "continuation" method which converts N-R problem to one in integrating ODE's.
3. Scaling
   For each variable, determine a "nominal" value, then
   - Scale each equation so a typical term using nominal variable values is about unity.
   - Scale each variable so its nominal value is about unity.
4. Rewrite equations to avoid divisions (division by zero is fatal), e.g.:
   - For $e^{b/T} - T = 0$, let $y = b/T$ (a new variable) and then replace the single equation with the two equations
     $e^y - T = 0$
     $yT - b = 0$
   - For $ln\ x - x = 0$
     Use $y = ln\ x$ and replace equations with
     $e^y - x = 0$
     $y - x = 0$

In this case *ln* x does not itself involve division but, when one forms the Jacobian element $\partial f/\partial x = 1/x - 1$, we have a division, which if x becomes zero will destroy the N-R equations by giving an infinite coefficient.

SUMMARY

Advantages/Disadvantages of Tearing vs. N-R with Sparse Matrix Methods

- Tearing is the more commonly used method.
- Tearing requires much less computer storage, usually.
- Tearing can be made quite efficient, particularly if all variables to be iterated are converged together, rather than solved with loops imbedded inside loops.
- Newton-Raphson approach is really very good if done correctly. We regularly solve hundreds of equations in 5 to 10 iterations total.
- The AERE Harwell subroutine MA28* is readily available to perform all the sparse matrix-equation solving portion of the problem.
- N-R requires Jacobian elements be evaluated or estimated. We find this easy to do because we add new variables to keep the algebra simple.
- Newton-Raphson approach permits sensitivity calculations to be performed easily and for little computational effort. □

REFERENCE

Westerberg, A. W., H. P. Hutchison, R. L. Motard, P. Winter, *Process Flowsheeting*, Cambridge University Press, Cambridge, England (1979).

*Contact: Numerical Analysis Group, Bldg. 8.9, AERE Harwell, OX11RA, England. The routine MA28 is part of the Harwell library of scientific routines which is available for £150. (Very cheap at that price.)

In Memorium

JOHN D. STEVENS

John D. Stevens, professor of Chemical Engineering at Iowa State University, died April 1, 1980, after a short illness. He was a faculty member at Iowa State for 15 years, during which time he received several teaching awards including the Western Electric Fund Award. He had served as National President of Omega Chi Epsilon. Dr. Stevens published several papers in the area of emulsion polymerization and crystallization.

ChE conferences

PHYSIOLOGICAL SYSTEMS FOR ENGINEERS

July 7-11, University of Michigan
Quantitative presentation of the terminology, anatomy, and performance of mammalian physiological systems. Especially for engineers and other scientists with little formal training in biological science. For further information: Engineering Summer Conf., 300 Chrysler Center, N. Campus, U. of Michigan, Ann Arbor, MI 48109.

NEW DEVELOPMENTS IN MODELING, SIMULATION, AND OPTIMIZATION OF CHEMICAL PROCESSES

July 21-30, M.I.T.
Basic principles and techniques for computer-aided design and control of industrial-scale chemical processes. Topics: steady state process simulation, process optimization, dynamic modeling and simulation of chemical processes, computer-aided process synthesis, comprehensive problem-oriented computing systems for chemical process design. For further information: Dir. Summer Session, MIT, Rm. E19-356, Cambridge, MA 02139.

ADVANCES IN EMULSION POLYMERIZATION AND LATEX TECHNOLOGY

June 9-13, Lehigh University
For further information: Dr. M. S. El-Aasser, ChE Dept., Lehigh University, Bethlehem, PA 18015

SHORT COURSES ON EMULSION POLYMERS

August 18-22, Davos, Switzerland
For further information: Dr. Gary Poehlein, ChE Dept., Georgia Institute of Tech., Atlanta, GA 30332