Solution of Material Balances for Flowsheets Modelled with Elementary Modules: The Unconstrained Case

A study is made of the structure and properties of material balance simulation problems, and a technique is developed for their efficient solution. The method requires neither simultaneous solution of all balance equations nor iterative convergence methods. Instead, for each stream mixing point in the flow sheet, a vector balance equation is developed which contains as unknowns only mixer output streams. This unique set of vector equations is sequenced for solution by using precedence ordering and substitution techniques. It is shown that only as many vector equations need to be solved simultaneously as there are streams which would require iteration in the conventional sequential approach. Once the mixer output streams are calculated, the remaining intermediate process streams are evaluated directly with no further equation solving. Computational results are presented showing the efficiency of the method.

MOHINDER K. SOOD
G. V. REKLAITIS
and
J. M. WOODS
School of Chemical Engineering
Purdue University
West Lafayette, Indiana 47907

SCOPE

The solution of material balances for large scale process flow sheets without the incorporation of detailed unit operations models is a basic chemical engineering problem of considerable practical importance both in preliminary design and in the study of existing plants. Three computer oriented strategies for solving flow sheet material balances have been reported or are in use at the present time: the completely simultaneous approach, the recycle stream iteration approach, and the split-stream approach of Rosen-Nagiev (Rosen, 1962). All of these approaches employ a representation of flow sheets in terms of a fixed repertoire of standard models which are linear in the stream flows. The simultaneous solution method necessitates assembly and storage of very large equation sets, requires the use of sparse matrix techniques, and does not directly exploit the inherent network structure of flow sheets in which species flows are aggregated into streams. The recycle iteration approach requires the selection of streams to be estimated and iterated upon and necessitates time consuming iterations with uncertain convergence. The Rosen-Nagiev technique uses simultaneous solution of the sets of material balances written around each stream mixing point aggregated by species. In general, in the presence of reactions, the species balances become coupled, and the entire system of sets of equations must be solved in iterative passes.

Recognizing that the material balance simulation problem is inherently a highly structured linear problem, the goal of this paper is to analyze the general problem structure and to develop a technique which requires no iterations and a minimum of simultaneous equation solving. The key to the method is the development of a vector balance equation for each stream mixing point in the flow sheet. The equation for each mixing point is constructed by tracing upstream the input streams to that mixer until either another mixer output or a process input stream is reached and aggregating the stream transformation operators of the flow sheet units encountered along the way. These equations can be constructed very efficiently from the specification of the flow sheet stream connections using symbolic manipulation. Again, employing only symbolic manipulation, the equations can be sequenced for solution using equation ordering and equation substitution techniques. Once the calculation order is defined, the equation coefficients are evaluated numerically and the equations solved for the mixer output streams in the required sequence. With the mixer output flows known, all other process streams can be determined by proceeding downstream from each mixer and evaluating each sucessive unit output stream from its known input. The overall strategy can thus be viewed as a decomposition of the complete set of balance equations, both on the basis of the network structure of the flow sheet and by means of equation ordering and sequencing techniques into a few small blocks which need to be solved simultaneously and a large number of blocks which can be evaluated sequentially.

CONCLUSIONS AND SIGNIFICANCE

The properties of unconstrained flow sheet material balance problems modeled in terms of a fixed set of four simple unit modules were analyzed; a solution method was presented; a computer program implementing the technique was developed and was shown to be efficient in time and storage requirements.

The key properties of unconstrained balance problems were shown to be that the unit equations are linear in the species flows, that the naturally induced order of unit calculations is in the flow direction, and that the unconstrained problem will always have the correct number of specifications to yield a unique solution. It was demonstrated that all recycle loops must be closed by mixers

0001-1541-79-9543-0209-$01.25. © The American Institute of Chemical Engineers, 1979.

and that a minimum number of these mixers can be identified which will break all recycle streams. A simple labeling procedure was presented for identifying these mixers. This procedure can also be used with conventional recycle iteration methods. A methodology was developed for assembling vector mixer balance equations. These equations were shown to be unique to the flow sheet structure and were proven under mild restrictions to yield unique solutions. An algorithm which employs equation substitution and precedence ordering phases was developed for sequencing these equations into small simultaneously solvable blocks. A computer program employing the proposed method was implemented, and test results are given for six examples: four moderately sized literature problems and two large scale coal conversion process flow sheets. The program was shown to require only modest computer storage and time. A few sample results indicate that the approach compares very favorably against the conventional recycle iteration approach.

It is generally recognized that computer aided flow sheet calculations should be carried out at the simplest level appropriate to the goals of the study and to the reliability of the available information. During the initial stages of flow sheet study, whether of a new process or an existing one, simple material balances may alone suffice. At this level, the balance specifications may be set from plant data, if an existing plant is being considered, or from engineering experience or small scale data, if a new process flow sheet is under development. At later stages, detailed process material and energy balances involving accurate physical property estimates, specific reaction kinetics, and elaborate unit operations calculations may be required. Typically, during the course of a process study, the flow sheet calculations ought to cycle through the several levels of detail as the flow sheet is modified and refined (Agarwal et al., 1978). Unfortunately, most existing programs use the same calculation strategy regardless of the level of detail of the calculations involved. Namely, flow sheet calculation programs, almost universally, employ the sequential modular approach which requires iteration on tear streams to accommodate cycles in the information flow network (Motard et al., 1975). While this mode of calculation may be justified when detailed simultaneous material and energy balances are to be carried out, it is, as pointed out by Hutchinson and Leesley (1973), quite inefficient if only the flow sheet material balances are to be calculated.

Two computer oriented approaches to material balancing have been reported in the literature to date. Rosen (1962) presented an approach based on Nagiev's split-stream concept (Nagiev, 1964) in a paper discussing the integration of flow sheet material balance calculations with repeated updating of the parameters of the simple material balance models using detailed unit models. In this method, a block of mixing point material balances are written for each species in the flow sheet, and each set of balances for a species is solved separately. If reactions are present, then the species balance sets become coupled. A decoupling is forced by introducing hypothetical process input streams corresponding to the reaction product generation rate and requiring that these must be specified in advance. Since, in general, these flows are not known in advance, the entire system of sets of species equations must be solved repeatedly until convergence to the final values is achieved. The SYMBOL program, described by Hutchinson (1974), appears to be the only other program specifically designed for flow sheet material balances which does not use stream iterations. In SYMBOL, the complete set of linear material balances associated with a given flow sheet is solved simultaneously. Setup of the many balance equations typically required is done in the program itself via several types of equation generation modules. But this solution strategy necessitates assembly and storage of a very large equation set, requires the use of sparse matrix techniques, and does not directly exploit the inherent network structure of the flow sheet. In this paper, we present an alternate approach to flow sheet material balancing which can be viewed as a compromise between the sequential and the simultaneous calculation strategies. It is a decomposition scheme which uses simultaneous solution only for certain streams that in the conventional approach would require tearing and calculates the remaining streams in a purely sequential fashion. This calculation strategy is based on key concepts first proposed by Kneile (1975) in an unpublished thesis.

THE UNCONSTITUTIONAL MATERIAL BALANCE PROBLEM

For purposes of material balances, any process unit can be represented as either a component separator, a flow splitter, a stoichiometric reactor, a stream mixer, or some combination of these foul elementary operations. With the exception of the mixer, each of these elementary operations or modules can be viewed as a linear transformation which operates on the module input stream to obtain the module output stream or streams.

In particular, the component separator, a device which separates a single input stream into J output streams according to constant component split fractions, can be represented by the matrix material balance equations

$$\mathbf{N}^{(j)} = \mathbf{T}^{(j)}\mathbf{N}^{(\mathrm{in})} \quad j = 1, .., J$$

The S component vectors $\mathbf{N}$ in the equations are vectors of component species flows, and the arrays $\mathbf{T}^{(j)}$ are diagonal entries $t_{ss}^{(j)}$ corresponding to the fraction of species s in the input stream which appears in the output stream j. By convention, $t_{ss}^{(j)} = 0$ if species s is not present in the input stream. Thus, for all species s such that $N_s^{(\mathrm{in})} \neq 0$, $\sum_j t_{ss}^{(j)} = 1$.

The flow splitter, a separator which divides a single input stream into J output streams with identical compositions, can be represented by the same type of transformation equations. However, in this case, the nonzero diagonal entries of any transformation matrix $\mathbf{T}^{(j)}$ are all identical.

The stoichiometric reactor is simply a single input stream, single output stream device which converts a specified fraction X_k of a selected key reactant k according to a single reaction equation with known stoichiometric coefficients. This module can be represented by

$$\mathbf{N}^{(\mathrm{out})} = \mathbf{C}\ \mathbf{N}^{(\mathrm{in})}$$

where **C** is an array with elements

$$C_{lm} = \begin{cases} 1 & \text{if } l = m \neq k \\ (1 - X_k) & \text{if } l = m = k \\ -\dfrac{\sigma_l}{\sigma_k} X_k & \text{if } m = k \text{ and } l = 1, \ldots, S \\ & \qquad \text{but } l \neq k \\ 0 & \text{otherwise} \end{cases}$$

and where σ_s is the stoichiometric coefficient of species s in the reaction defined with $\sigma_s > 0$ for products and $\sigma_s < 0$ for reactants. Note that under this convention for writing the reactor balances, a reactor involving R reactions is represented by R transformation matrices $\mathbf{C}^{(r)}$, $r = 1, \ldots, R$. Hence, the conventional species balance

$$N_s^{(\text{out})} = N_s^{(\text{in})} + \sum_{r=1}^{R} \sigma_{sr} r_r \quad s = 1, \ldots, S$$

written in terms of a sum of rates of reaction r_r will be replaced by a product

$$\mathbf{N}^{(\text{out})} = \mathop{\pi}_{r=1}^{R} \mathbf{C}^{(r)} \mathbf{N}^{(\text{in})}$$

In this representation, the reactions are assumed to be taking place in series, rather than simultaneously. Thus, the conversion for each reaction r must be defined on the basis of the hypothetical key component flow rate after reaction $r - 1$ has taken place. The advantages of this mode of representation will become apparent from the subsequent discussion.

Finally, the stream mixer is a device which combines I input streams into a single output stream and is represented by the vector equation

$$\mathbf{N}^{(\text{out})} = \sum_{i=1}^{I} \mathbf{N}^{(i)}$$

Associated with this set of elementary modules is a set of module parameters which we call normal simulation specifications. They are the split fractions $t_{ss}^{(j)}$ for every species and every separator and splitter output stream and the reaction stoichiometry and key reactant conversions for all reactions in all reactors.

In addition to these module parameters, there is a further set of parameters associated with any flow sheet described in terms of these modules. This third set of normal specifications consists of the species flow rates in all external input streams to the process.

A material balance problem in which all the normal parameters are assigned known values is called an unconstrained material balance problem. The unconstrained material balance problem (UMBP) has three significant properties:

1. All the balance equations are linear equations in the species flows.
2. The naturally induced order of sequential module calculations is in the direction of flow.
3. The balance problem will always have the exact number of specifications required to yield a unique solution.

The first property is obvious, since in a normally specified problem all the module equations reduce to constant coefficient linear equations. The second follows immediately from the structure of the transformation equations: given the input stream vector and the transformation matrix, the output vectors of each module can be calculated as simple matrix-vector products. On the other hand, calculation in the reverse order requires simultaneous solution of the linear transformation equations.

The third property, although intuitive, is less obvious but nonetheless can be readily proved by induction. The property clearly holds for single module flow sheets. Suppose it is true for a $N - 1$ module flow sheet, and suppose an N^{th} naturally specified module is added along with any additional natural specifications so that the resulting N module flow sheet is unconstrained. Regardless of the identity of the module inserted into the flow sheet, the only unknowns which are introduced with the additional module are the module output streams. Yet, since the module is naturally specified, its output streams can always be calculated once its input streams are known. These input streams can only be of three types: process input streams, streams previously existing in the $N - 1$ module flow sheet, and streams added as output streams to one or more of the previously existing $N - 1$ modules. In the first case, since process input streams are assumed specified as part of the natural specifications, these inputs to the N^{th} module are obviously known. In the second case, the inputs to the N^{th} module can be calculated since, by hypothesis, they are calculatable from the input stream to the corresponding source module of the $N - 1$ unit flow sheet. Consequently, the specifications of the ideal N module problem must be sufficient to calculate all streams provided that the specifications for the $N - 1$ module problem are sufficient. An alternate demonstration of property 3 can be developed by recognizing that the unconstrained material balance problem is a matrix signal flow graph with known source nodes and known arc transmittances.

It should be recognized that while the natural specifications will always be correct in number, no guarantees can be given that the resulting set of linear equations will have maximum rank or that feasible, that is, nonnegative, flows will result from these specifications. Rank of the equations and the feasibility of the resulting solution will be determined by specific values of the natural parameters selected rather than by the flow sheet structure. Some care must therefore be exercised in selecting parameter values. For convenience in the subsequent discussion, a set of UMBP specifications will be called proper if the natural parameter values are such that the complete set of flow sheet module equations has maximum row rank. that is, rank equal to the total number of unknown species flow rates. A UMBP specification set will be called improper if it is not proper. Two easily recognized cases of improper specifications are given in the appendix to this paper; others may well exist which are less transparent.

CONVENTIONAL SOLUTION STRATEGIES FOR UMBP

Since the unconstrained material balance problem can be viewed as a large set of linear equations which generally will have maximal row rank, unconstrained material balance problems can always be solved by simultaneous solution as reported by Hutchison (1974). However, simultaneous solution does not fully exploit the inherent sparsity of the equation set (each stream vector is involved in at most two transformation equations); it does not take advantage of the naturally induced calculation direction, nor does it exploit the fact that the species balance equations are naturally grouped into blocks corresponding to streams. These structural features are exploited by the conventional sequential solution strategies. However, this is achieved at the price of iterations on selected tear streams, even though the underlying problem is linear. The conventional sequential solution strategies determine the module calculation sequence using three stages of analysis:

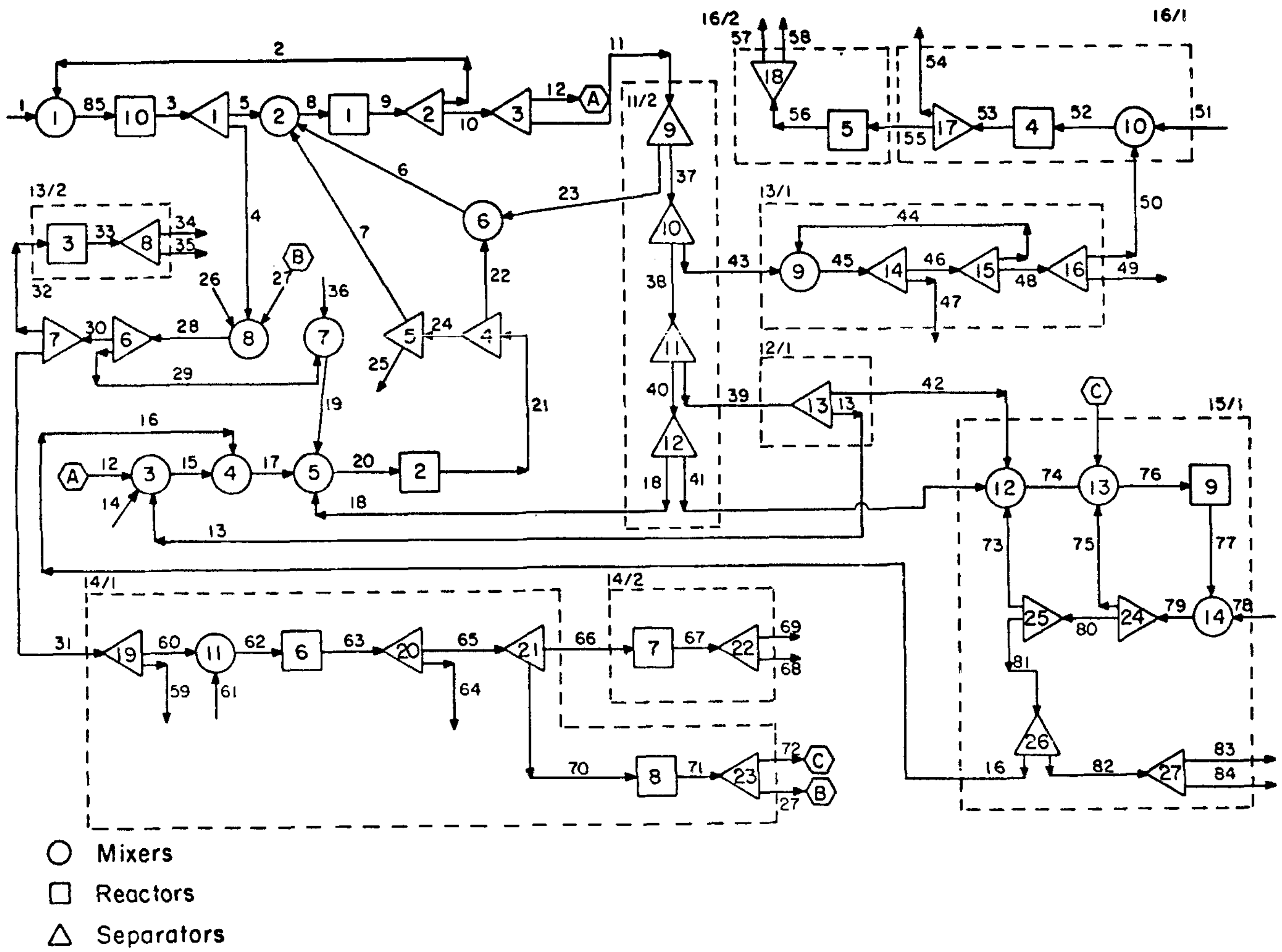

Fig. 1. COED coal conversion plant flow sheet (R. M. Parsons, 1975).

1. Partitioning: identifying all closed recycle loops in the flowsheet.

2. Tearing: selecting the streams which when torn will both break all recycle loops and will yield an optimal iterative process. Usually optimal means the fewest tear streams (Motard et al., 1975).

3. Sequencing: ordering the loop calculations so that the inputs of each module are known before its outputs are calculated.

A voluminous literature exists on various algorithms for carrying out all or portions of these three analysis stages. Some of the more recent advances can be found in the papers of Barkley and Motard (1972), Uphadye and Grens (1972), Pho and Lapidus (1973), Motard et al. (1975), and Genna and Motard (1975).

There has been and continues to be considerable discussion as to the relative merits of various module ordering algorithms for the general case when detailed unit operations modules are employed in a computer aided flow sheet study. However, as shown in the subsequent discussion, these algorithms are inefficient when applied to the unconstrained material balance problem because they do not exploit the structure of the modules themselves.

First of all, in the UMBP the problem of identifying all flow sheet recycle loops is remarkably simple. In order to close a recycle loop it is necessary to employ a module with multiple inputs. Since the mixer is the only such unit, all loops must be terminated with a mixer. The next question, that of deciding which streams require tearing, also resolves itself readily. Since all loops must terminate with mixers, tearing of all of the mixer output streams is sufficient to break all recycle loops. Hence, in the UMBP the maximum number of streams which must be torn is given by the number of mixers in the flow sheet. Of course, this is but an upper bound. Some mixers, for example, only serve to introduce a process feed stream into a recycle loop actually terminated by another mixer. These nonessential mixers can be identified by a path tracing procedure initiated with each mixer. Before discussing such a procedure, it is necessary to make the concepts employed more precise by introducing the following terminology:

Table 1. Initial Branch Source Stream Table

Mixer number	MOS	Branch source streams*
1	85	*1*, 8
2	8	85, 20, 6
3	15	8, *14*, 8
4	17	15, 79
5	20	19, 17, 8
6	6	20, 8
7	19	*36*, 28
8	28	*26*, 85, 62
9	45	8, 45
10	52	*51*, 45
11	62	28, *61*
12	74	8, 8, 79
13	76	62, 74, 79
14	79	76, *78*

* Process input streams are underlined.

1. An essential mixer set for any given flow sheet is the smallest set of mixers, whose output streams must be torn in order to break all recycle loops. The mixers contained in that set are called essential mixers. The nonessential mixers in a flow sheet are all mixers not included in the essential mixer set. An essential mixer set is not necessarily unique. However, the cardinality of the set is an invariant property of the flow sheet.

2. A mixer branch is the sequence of streams encountered in tracing a mixer input stream upstream in the process network until a mixer output or process input stream is encountered.

3. A branch source stream is the first mixer output or process input stream encountered in tracing a mixer branch. It is the last stream of the branch.

Since reactors, separators, and splitters are all single input modules, each mixer input stream will have associated with it a single branch which is independent of the order in which the input stream tracing is carried out. Since the branches associated with a mixer are unique and the branch source streams associated with each branch are unique, the branch source streams associated with a given mixer are unique characteristics of that mixer. Hence, given any arbitrary flow sheet, one can construct and associate with that flow sheet a characteristic table of mixer output streams and their associated branch source streams.

Identification of a set of essential mixers can then proceed from that table as detailed below. However, to simplify the exposition, we assume that all source streams that are process input streams are deleted from the table.

Mixer Labeling Procedure

Initially, label all mixers as essential and eligible. In executing the procedure, all ties for selection may be decided arbitrarily.

Step 1: scan the table to find any eligible mixer output stream (MOS) which has a single source stream. If none exists, go to step 4. Otherwise, continue with step 2.

Step 2: If the MOS does not appear in its associated branch source stream list, label it as nonessential and go to step 3. Otherwise, flag this MOS as ineligible for further consideration and go to step 1.

Step 3: Search the table of mixer branch source streams and replace each occurrence of the MOS labeled nonessential with the branch source streams associated with the nonessential MOS. Flag this MOS as ineligible for further consideration and go to step 1.

Step 4: Search the table to find the eligible MOS with fewest source streams. If no eligible MOS's remain, go to step 5. Otherwise, go to step 2.

Step 5: The number of required tear streams is equal to the number of essential MOS's, and these MOS's will serve as tear streams.

This procedure has elements of similarity with the signal flow based technique reported by Barkley and Motard (1972) in its focus on identification of self-loops. However, our reliance on elementary modules and mixer output streams avoids the need for loop identification, graph reduction, and two way node considerations.

We illustrate the use of this labeling procedure with the table of MOS's and their branch source streams corresponding to the flow sheet stream in Figure 1. This flow sheet represents the Conceptual COED Coal Conversion Plant reported by the Ralph M. Parsons Company (1975). The branch source stream list for each of the fourteen mixers, given in Table 1, is constructed by tracing the mixer inputs until a mixer output or process input stream is encountered. For instance, mixer 2 has three input streams. The first input, stream (5), can be traced upstream to stream (85), the output of mixer 1. The second input, stream (7), can be traced to stream (20), the mixer 5 output stream. Finally, the third input stream (6) is itself a mixer output stream; hence, it does not need to be traced further. The branch source streams are thus (85), (20), and (6). The remainder of the branch source stream table is constructed analogously. This table then serves as input to the mixer labeling procedure.

Table 2. Summary of Mixer Labeling Operations

Mixer number	MOS	Label	Branch source stream updating*
1	85	NE	8
2	8	E	85, 20, 6, 8, 28, 76
3	15	NE	8
4	17	NE	15, 79, 8, 76
5	20	NE	19, 17, 8, 28, 76
6	6	NE	20, 8, 28, 76
7	19	NE	28
8	28	E	85, 62, 8, 28
9	45	E	8, 45
10	52	NE	45
11	62	NE	28
12	74	NE	8, 79, 76
13	76	E	62, 74, 79, 28, 76, 8
14	79	NE	76

* Process input streams not shown and duplicate indexes in branch source stream list deleted.

If we begin the procedure with mixer 1, which has a single source stream, we can immediately label mixer 1 and its output stream (85) as nonessential. Stream (85) occurs in the source stream lists for mixers 2 and 8. Step 3 then requires substitution of (85) by (8). The procedure returns to step 1 to consider another mixer with a single source stream, mixer 3. Mixer 3 is clearly nonessential. Stream (15) occurs in the source stream lists for mixers 4; hence, (15) is replaced by (8). Continuing, mixers 7, 10, 11, and 14, each with single source stream, are labeled nonessential. Next, mixers 4 and 6, each with two source streams, are labeled nonessential before mixer 8 is selected. Since stream (28) occurs both as MOS for mixer 8 and in its source stream list, mixer 8 is essential. Continuing in this fashion, we generate the substituted source stream table shown in Table 2. The table indicates that mixers 2, 8, 9, and 13 are essential. Because of the choice in the order in which the mixers are considered, this set of essential mixers is not necessarily unique. However, any set of essential mixers for this flow sheet will contain four mixers.

In the conventional approach, these essential MOS's would be the tear streams which would be iterated upon in consecutive calculation passes through the flow sheet. The order in which these tear streams would be updated and converged is dependent upon the final substituted table of branch source streams obtained from the labeling procedure. For each essential MOS, its associated branch source stream list will contain at least one essential MOS. Those essential MOS's whose branch source stream list contains only one MOS can be converged as independent loops. Those whose lists contain multiple MOS's will have to be converged in blocks involving common MOS's. For instance, in the previous example, streams (8), (28), and (76) will have to be converged simultaneously. After these streams have converged, stream (8) will be known, and then stream (45) can be converged by itself. The calculation using the sequential approach is generally inefficient because it does not exploit the linearity of the module transformation equations. By exploiting

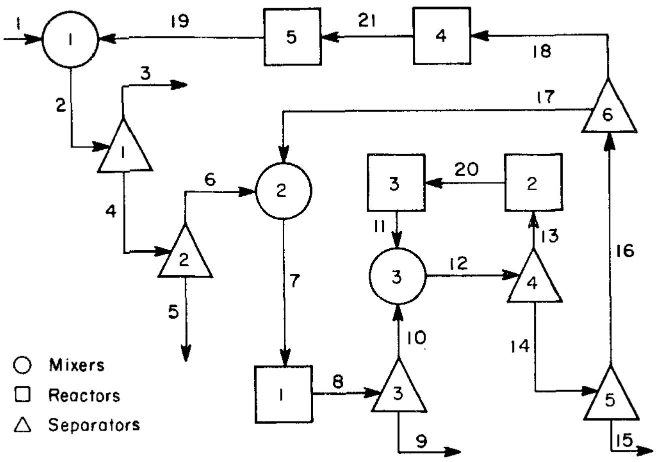

Fig. 2. Example three mixer flow sheet.

the structure of the unconstrained material balance problem, iteration on essential MOS's can be replaced by direct solution of linear equations involving only the essential MOS's.

THE MIXER EQUATIONS

As shown in the previous sections, each mixer has associated with it a unique set of branches, and each such branch terminates with a unique branch source stream. Since the relationship between each adjacent pair of streams in a branch is described by a linear transformation matrix, an explicit expression relating each mixer input stream to the source stream of that branch can be obtained by simply accumulating a product of the transformation matrices of each module of that branch. The linear transformation thus obtained relating each mixer input stream to its branch source stream can be substituted into the mixer balance equation. The result is a vector equation expressing the mixer output stream as a linear function of mixer output streams and process input streams.

To make these operations more precise, we define the following:

1. A branch transformation matrix is the accumulated product array formed by successively multiplying from the right the module transformation matrices encountered in tracing a branch upstream to its source stream.
2. A mixer equation is a vector equation formed by summing the products of the branch transformation matrices and their source stream vectors for a mixer and setting their sum equal to the mixer output stream vector.

The mixer equation is an explicit linear vector equation which will contain as vector unknowns the MOS's given in the corresponding mixer branch source stream list. Since the branches and branch source streams associated with a mixer are unique, the branch transformation matrices are unique. Hence, we can conclude that there will exist a single characteristic mixer output equation for each mixer.

To illustrate the formulation of these equations, consider the flow sheet shown in Figure 2. The flow sheet involves three mixers. The first of these, MIXER 1, has two input streams and hence two branches. The first branch terminates with source stream (1), and its transformation matrix will consist simply of an identity matrix **I**. The source stream for the second branch is stream (12), and its transformation matrix will be

$$\mathbf{C}^{(5)}\mathbf{C}^{(4)}\mathbf{T}^{(18)}\mathbf{T}^{(16)}\mathbf{T}^{(14)}$$

The mixer equation will thus be

$$\mathbf{N}^{(2)} = \mathbf{N}^{(1)} + \mathbf{C}^{(5)}\mathbf{C}^{(4)}\mathbf{T}^{(18)}\mathbf{T}^{(16)}\mathbf{T}^{(14)}\mathbf{N}^{(12)}$$

In the same fashion, the mixer equation for MIXER 2 will be

$$\mathbf{N}^{(7)} = \mathbf{T}^{(6)}\mathbf{T}^{(4)}\mathbf{N}^{(2)} + \mathbf{T}^{(17)}\mathbf{T}^{(16)}\mathbf{T}^{(14)}\mathbf{N}^{(12)}$$

and that for MIXER 3

$$\mathbf{N}^{(12)} = \mathbf{C}^{(3)}\mathbf{C}^{(2)}\mathbf{T}^{(13)}\mathbf{N}^{(12)} + \mathbf{T}^{(10)}\mathbf{C}^{(1)}\mathbf{N}^{(7)}$$

Note that the mixer equations can be easily computed and efficiently stored since the branch transformation matrices are formed as products of only two types of matrices: diagonal matrices, the $\mathbf{T}^{(j)}$, and matrices containing a diagonal plus a nonzero column, the $\mathbf{C}^{(r)}$. Since products of diagonal matrices will result in a diagonal matrix, in many cases, for instance the mixer (2) output equation of the above example, the output equation coefficients will simply be diagonal matrices. In the presence of a reactor in a branch, the branch transformation matrix may contain nonzero off diagonal elements. However, it is easy to show that the product of two $\mathbf{C}^{(r)}$ matrices defined in terms of the same key reactant will have the same structure as the original $\mathbf{C}^{(r)}$'s. Moreover, multiplication of two $\mathbf{C}^{(r)}$ matrices defined in terms of two different key reactants will result in a product matrix with two nonzero columns in the same location as those of the original $\mathbf{C}^{(r)}$ matrices. These algebraic properties can be employed to great advantage in any program implementing these equations.

The utility of the mixer equations lies in the fact that once they are solved for the unknown mixer output streams, then the remaining flow sheet streams can be recovered by simply tracing each mixer output downstream and calculating each succeeding module output stream vector by applying the appropriate module transformation to the module input stream vector. These calculations simply involve matrix-vector multiplications. No further equation solving is necessary. The key question in this strategy is, of course, whether the mixer equations are sufficient to calculate the mixer output stream vectors. The answer to this question is provided by the following elementary proposition.

If the UMBP is properly specified, the set of mixer equations will yield a unique solution of all mixer output streams.

Since the number of mixer equation sets is equal, by construction, to the number of mixer output streams, a proof of this proposition simply involves demonstration that under the stated hypothesis the set of mixer equations must have rank equal to the total of the number of species present in each mixer output stream. From the properties of a UMBP and the definition of proper specification, it is evident that if the specification is proper, the complete set of flow sheet module equations must have maximum row rank. Construction of each mixer equation involves nothing more than row elimination (Gaussian elimination) applied to the subset of flow sheet equations consisting of the mixer balances plus the transformation equations associated with each stream which is involved in the mixer branches. Since row elimination preserves linear independence, the resulting mixer equations must have rank equal to the number of species present in the mixer output streams.

As pointed out in our discussion of essential mixers, not all mixer output streams need to be torn to break all flow sheet loops; rather, only the essential mixer output streams will have to be tear streams. The equivalent situation in terms of the mixer output equations is that not all mixer equations need to be solved simultaneously; rather, only those corresponding to essential mixers will require solution. The nonessential mixer output streams can be directly calculated once the essential output streams are known. To locate the essential mixers, the labeling procedure given in the previous section could be applied to the mixer equations themselves using algebraic substitutions rather than substitution of branch source stream lists. For instance, in the case of the mixer output equations associated with the flow sheet given in Figure 2, the mixer 1 equation is clearly nonessential. When $\mathbf{N}^{(2)}$ is eliminated from the remaining two mixer equations by substitution, the mixer 2 output equation becomes

$$\mathbf{N}^{(7)} = \mathbf{T}^{(6)}\mathbf{T}^{(4)} [\mathbf{N}^{(1)} + \mathbf{C}^{(5)}\mathbf{C}^{(4)}\mathbf{T}^{(18)}\mathbf{T}^{(16)}\mathbf{T}^{(14)}\mathbf{N}^{(12)}] + \mathbf{T}^{(17)}\mathbf{T}^{(16)}\mathbf{T}^{(14)}\mathbf{N}^{(12)}$$

while the mixer 3 equation remains unchanged. The above equation is itself nonessential [$\mathbf{N}^{(7)}$ does not appear in the right-hand side]; hence, $\mathbf{N}^{(7)}$ can be eliminated by substitution into the mixer 3 equation to obtain a single matrix equation for $\mathbf{N}^{(12)}$, the essential mixer output stream.

Note that although the substituted expressions can and do become rather cumbersome because of the growing number of terms involving matrix products, computer implementation can be carried out very easily since the substitutions can be carried out symbolically. Numerical calculation of the coefficients of the mixer output equations only have to be executed once the final set of essential mixer equations have been assembled.

Of course once the essential mixer equations are identified and constructed, efficient solution requires that these vector equations be further ordered and decomposed so that the size of the blocks which must be solved simultaneously is as small as possible. A complete procedure which will both select the essential mixer equations and order their solution is given in the next section.

SOLUTION OF THE MIXER EQUATIONS

The general form of the mixer equation involving K mixer output streams is

$$\mathbf{N}^{(k)} = \sum_{j=1}^{K} \mathbf{M}_{jk}\mathbf{N}^{(j)} + \mathbf{q}^{(k)} \quad k = 1, K$$

where the $\mathbf{q}^k$ are vectors of constants resulting from branches with source streams which are process input streams, and $\mathbf{M}_{jk}$ is a matrix of coefficients of the j^{th} mixer output stream in the k^{th} mixer equation.

Effective ordering of these equation sets should incorporate the following four tasks:

1. Identification of mixer equations which can be solved separately early in the calculations.

2. Identification of mixer output streams which can be determined late in the calculations.

3. Identification of mixer output streams which can by substitution be rendered individually solvable.

4. Identification of equations which must be solved in blocks.

Elements of ordering procedures which accomplish those four tasks have been reported in the computer aided design literature for various equation precedence ordering and recycle determination applications. Christensen and Rudd (1969) reported an algorithm which accomplishes tasks 1 and 2. Soylemez and Seider (1973) developed a procedure which in part addresses all four of these tasks for a class of scalar equations. The specialized algorithm reported here incorporates ideas and constructs derived from this earlier work. For instance, as do most ordering procedures, the method operates upon a structural matrix associated with the equation to be ordered. However, in our application the structural matrix is constructed in terms of vector rather than scalar equations. In particular, the structural matrix contains a column j for every mixer output stream vector and a row i for every mixer output equation. The elements s_{ij} of the array are assigned values 0 or 1 depending upon whether or not mixer output stream j occurs in mixer equation i. In addition, we supplement the structural array with three sets of indexes; the occurrence frequency, the substitution frequency, and the term index. The occurrence frequency of a stream j is equal to the total number of times stream j appears in the right-hand side of the set of mixer equations. The substitution frequency of stream j is the product of the occurrence frequency of that stream and the number of terms occurring in the right-hand side of the mixer output equation in which stream j is an output. The term index of a row of the structural matrix is equal to the number of nonzero entries contained in that row.

The three phase algorithm summarized below uses these indexes and the structural matrix to generate an ordering of the mixer equations which will make the blocks of equations which must be solved simultaneously as small in size as possible. The first phase of the logic consists of the Christensen-Rudd algorithm and serves to identify the matrix equations which can be handled individually. The phase II, or substitution, algorithm takes advantage of the special structure of the mixer equation to eliminate by substitution into the right-hand sides of all equations the output stream vectors which are not involved in self-loops. The algorithm uses the substitution frequency index to preferentially select for substitution the MOS's which will generate the fewest new matrix terms and thus will require the least matrix operations for each substitution. This, of course, is a heuristic which has proved satisfactory in test cases but cannot be guaranteed to result in the fewest matrix operations overall. The phase III, or blocking, algorithm essentially involves a systematic examination of all possible combinations of sets of unordered mixer equations involving from two up to the total number of unordered equations. After each simultaneous solution block is identified, the remaining unordered equations are resubjected to phases I and II of the algorithm prior to any further passes through phase III. However, in all of the flow sheets we have examined to date, a single pass through phase III has generally sufficed. This is because most flow sheets, even those of considerable size such as Figure 1, will contain no more than, say, six essential mixers.

THE MIXER EQUATION ORDERING LOGIC

Step 1: construct the structural matrix and set the positive precedence ordering index p and the negative precedence ordering index n to zero. Phase I: the CR algorithm.

Step 2: locate an i^{th} row of the structural matrix that has a single nonzero entry. If no such row exists, go to step 4.

Step 3: locate the column j which contains the nonzero element present in the i^{th} row. Set $s_{ij} = 0$. $j = 1, .., K$. Set $p = p + 1$ and assign the i^{th} mixer equation the precedence order p. Go to step 2.

Step 4: locate a j^{th} column of the structural matrix that has a single nonzero entry. If no such column exists, go to step 6.

Step 5: locate the row i which contains the nonzero element present in the j^{th} column. Set $s_{ij} = 0$, $i = 1, \ldots,$ K. Set $n = n - 1$ and assign the i^{th} mixer equation the precedence order n. Go to step 2.

Step 6: if all equations have been ordered, go to step 18. Otherwise, go to step 7 of the substitution algorithm. Phase II: the substitution algorithm.

Step 7: calculate the updated occurrence frequency for all mixer output streams. If a mixer output stream occurs on the right-hand side of its own equation, or if that output stream has already been ordered, set the corresponding occurrence frequency to a large number θ.

Step 8: for each stream whose occurrence frequency is not equal to θ, calculate the substitution frequency. If no stream with occurrence frequency less than θ can be found, go to step 10 of the blocking algorithm.

Step 9: find the stream j with the lowest substitution frequency. For each k, $k \neq j$, $k = 1, K$ such that $s_{kj} \neq 0$, update the row of the structural matrix to reflect the substitution of equation j to eliminate stream j. Reset $s_{kj} = 0$. Go to step 2. Phase III: determination of equation blocks.

Step 10: calculate the term indexes. Set NTEST = 1.

Step 11: set NTEST = NTEST + 1. If all equations have been ordered, go to step 18.

Step 12: find the first equation i with term index = NTEST. If no such equation exists, go to step 11.

Step 13: set $J = 1$ and INDEX $(J) = i$.

Step 14: search for another equation l with term index $\leqq$ NTEST. If no such equation exists, go to step 11.

Step 15: if the INDEX (1) row of the structural matrix does not contain the l^{th} row, go to step 14.

Step 16: set $J = J + 1$ and INDEX $(J) = l$. If J is less than NTEST, go to step 14.

Step 17: set $p = p + 1$. Set the term indexes of equation INDEX (J), $J = 1$, NTEST equal to zero. Assign these equations the precedence order p and set the corresponding rows and columns of the structural matrix to zero. Go to step 2.

Step 18: the mixer equations should be solved beginning with the equations indexed 1 and continuing in increasing index value until the equation with index p is solved. Next, solve the equation indexed n and continue in increasing index value until the equation indexed -1 is solved. Equations with the same index should be solved as a block.

Note that the equation used for substitution is not itself removed from the structural matrix. Instead, after substitution, that equation and the remaining substituted mixer output equations are resubjected to the CR algorithm. Note, further, that in the CR algorithm no distinction is made between matrix equations which must be solved simultaneously for a stream and those that can simply be evaluated for that stream. That distinction is made when the actual calculations must be carried out. Finally, the reason in step 7 for setting to a large number the occurrence frequency of a mixer output stream which occurs on both sides of its mixer equation is that use of this equation for substitution would, in general, require inversion of a matrix. Matrix inversion is computationally much more time consuming than solution of an equation set for which that matrix is the coefficient array. Consequently, substitutions requiring inversion are disallowed.

To illustrate the application of this ordering algorithm, let us apply it to the flow sheet given in Figure 1. For purposes of this example, ties will be broken by always choosing the column furthest to the left. We first construct a structural matrix for the mixer equations and next calculate the term, occurrence frequency, and substitution frequency indexes. The results are shown in Table 3. When the phase I procedure is applied to the structural matrices MOS 52 and 45 and their corresponding mixer equations, (10) and (9) are immediately assigned negative precedence orders (-1 and -2, respectively) and their associated columns and rows deleted. The phase II procedure is next initiated, and on the basis of the substitution frequency index, variable 15 is selected to be substituted out using its mixer equation, row 3. After updating the structural matrix and returning to the phase I logic block, we assign mixer equation (3) negative precedence order -3. The ordering continues in this fashion, alternating between the substitution and CR phases, updating the indexes each time until only Equations (2), (8), (14) remain unordered. At that point, the phase III procedure is initiated, it is determined that these three equation sets must be solved simultaneously, and all three of the equations are assigned positive precedence order $+1$. The algorithm terminates, and the precedence order which is obtained consists of simultaneous solution of Equations (2), (8), (14), in substituted form as a block; followed by consecutive evaluation of Equations (13), (11), (5), (1), (12), (6), (4), (7), (3) in substituted form; simultaneous solution of Equation (9); and evaluation of Equation (10). Note that the identification of essential mixers obtained is somewhat different than that resulting from the mixer labeling procedure alone. This occurs because in the equation ordering, logic preference is given to substitutions which at each stage produce the fewest new terms.

An alternate scheme for the ordering and solution of the mixer output equations can be obtained by recourse to the matrix signal flow graph representation of the UMBP. Riegle and Lin (1972) have reported an algorithm which will obtain explicit gain expressions relating the essential nodes in the signal flow graph to the source nodes. Their algorithm, called the optimal topological method, is optimal in the sense that the gain expression contains the minimum number of explicit matrix inverses. In particular, they show that the number of explicit inverses required will be no larger than the numbers of essential mixers. This algorithm was adopted by Kneile (1975) to calculate the essential mixer output streams. However, the algebraic complexity of the matrix operations, the large intermediate storage required, and the unnecessary burden of generating matrix inverses rather than just solutions of linear equation sets caused this code to be substantially less efficient than the ordering approach developed in this paper. The application of the OTM is likely to become comparable in arithmetic operations to the equation ordering approach only if all essential mixer equations must be solved simultaneously and if the number of mixers is at least as large as the number of species in the flowsheet.

COMPUTER PROGRAM AND RESULTS

In summary, the overall UMBP solution technique developed in this paper consists of the following major steps:

Step 1: generation of mixer equations.

Step 2: ordering of the mixer equations.

Step 3: ordered solution of these equations.

Step 4: sequential generation of the remaining flow sheet streams.

A program named material balance program version I (MBP-I) implementing these four steps has been coded. The program generates the mixer equations in symbolic form. Each mixer equation is stored as a string of coded integers separated by delimiters. Each cluster of coded integers between delimiters is associated with a branch and represents the arrays whose product forms the branch transformation matrix. The last coded integer before each delimiter is the source stream identifier. This symbolic form of the mixer equation is used during the substitution

Table 3. Structural Matrix and Indexes for Parson's COED Flow Sheet

	Stream No.														
Mixer No.	85	8	15	17	20	6	19	28	45	52	62	74	76	79	Term index
1	1	1													2
2	1	1			1	1									4
3		1	1												2
4			1	1										1	3
5		1		1	1		1								4
6		1			1	1									3
7							1	1							2
8	1							1			1				3
9		1							1						2
10									1	1					2
11								1			1				2
12		1										1		1	3
13											1	1	1	1	4
14													1	1	2
Occurrence frequency	2	6	1	1	2	1	1	2	2	0	2	1	1	3	
Substitution frequency	4	24	2	3	8	3	2	6	4	0	4	3	4	6	

phase. Once the substitution phase is completed and the mixer equation set has been precedence ordered, the substituted equations are processed in the sequence and in the blocks specified by the precedence order. As each block of substituted equations requiring simultaneous solution reaches the processing stage, the substituted mixer equation strings are decoded, the coefficient matrix associated with each term in the equations is calculated, and the results are stored in appropriate locations. Once each complete block of equations which must be solved simultaneously has been generated, the linear equation solver is called, and the appropriate mixer output streams are calculated. Once all mixer outputs have been determined, the intermediate process streams are calculated by tracing each mixer output stream in the forward direction until the next mixer is encountered. The generation, ordering, and substitution of the symbolic equations require very little CPU time (less than 1/3 s in the largest problem reported in Table 4). The major computational effort lies in the simultaneous solution of the blocks of substituted mixer equations. The program has been tested on a number of flow sheet examples using the Purdue University CDC-6500 computer. The examples, some of which are drawn from Kneile (1975), are contained in a user's manual (Sood and Reklaitis, 1977) along with the instructions on the use of the program. Table 4 summarizes some of the salient features of these examples when specified as unconstrained problems.

Further details of the test problems are given in the users manual which is available upon request from the authors. The number of equivalent equations listed in Table 4 refers to the total number of species balance equations associated with the elementary module formulation of the problem. This number can serve as an estimate of the number of equations which would have to be solved simultaneously by a simultaneous solution program which uses the elementary module representation to facilitate automatic equation generation.

The times reported in Table 4 reflect largely the burden of simultaneous equation solving. In the fifth example, a block of three vector mixer equations (sixty-six scalar equations) and a block containing a single mixer equation (twenty-two scalar equations) must each be solved simultaneously. In the sixth example, a block of two vector mixer equations (thirty-four scalar equations) must be solved simultaneously. It is the size of these blocks that causes these two problems to require the largest times.

Table 4. Core and Execution Time Requirements for Ideally Specified Flow Sheet Examples

	Number of						Number of blocks containing				
	Mixers	Reactors	Separators	Streams	Species	Number of equivalent linear equation	One essential mixer	Two essential mixers	Three essential mixers	CORE requirement (K WORDS)	Execution time (CPU s)
Chlorobenzene flow sheet (Thatcher, 1962).	3	2	5	17	5	75	1	—	—	43	0.3
Ammonia flow sheet (Reklaitis, 1973)	4	5	4	21	9	153	1	—	—	44	0.7
Cracking gas flow sheet (Nagiev, 1964)	3	5	6	21	6	120	1	—	—	43	0.4
Natural gasoline flow sheet (Motard et al., 1969)	6	1	9	28	10	250	1	—	—	44	0.9
Coal conversion flow sheet (Parsons, 1975)	13	10	27	85	22	1 716	1	—	1	74	18.4
OCR coal conversion flow sheet (Parson, 1974; Kneile, 1975)	18	13	32	105	17	1 615	—	1	—	74	9.8

It should be noted that these results were obtained using a conventional (dense) Gaussian elimination method using partial pivoting. Preliminary results indicate that in the case of the fifth problems, solution times can be reduced by as much as one-third if a rudimentary sparse equation solver is employed. It should further be noted that the core requirements indicated in the table would have been even less if only those routines that are needed for solving an unconstrained flow sheet were loaded, and if the complete stream table were not stored.

Although detailed comparative tests have not been carried out, the mixer equation approach is inherently more efficient than either the direct simultaneous approach or Rosen-Nagiev's method because a much smaller equation set must be solved simultaneously. In fact, in the case of the latter method, simultaneous solution must, in the presence of reactions, be carried out repeatedly until the unknown reaction product flows are converged. When we compare the mixer equation approach against the sequential modular approach, it must first of all be noted that both will require determination of the same number of unknown species flows, those of the essential mixer output streams. The former method determines these flows by simultaneous linear equation solving; the latter approach determines these flows iteratively. The comparison thus reduces to the question of whether a set of linear equations is more efficiently solved iteratively or by direct Gaussian elimination. The latter approach is increasingly favored by those in the field, especially in view of its high reliability. For illustrative purposes, the sixth test problem listed in Table 4 was solved using a sequential modular program. The tear streams were the output streams of the two mixers which required simultaneous solution in MBP-I. The initial values of these tear streams were set at zero. Solution using successive substitution required 1 120 iterations and 119 s if a relative error tolerance of 10^{-4} was employed, and 2 165 iterations and 221 s if a tolerance of 10^{-7} was employed. Solution using bounded Wegstein convergence promotion required 550 iterations and 62.7 s for a relative error tolerance of 10^{-7}. These results compare quite poorly with the 9.8 s required by MBP-I to obtain an exact solution. Of course, these results are influenced by the choice of tear streams and their initial estimates; some choices converge more rapidly, others diverge.

ACKNOWLEDGMENT

The authors wish to acknowledge the seminal contributions of Professors L. B. Koppel and R. A. Greenkorn in the development of this approach to solving flow sheet material balances. The work was performed under ERDA contract No. EX-76-C-01-2275.

NOTATION

$\mathbf{C}^{(r)}$	=	reactor transformation matrix, S × S
K	=	number of mixer output streams
$\mathbf{M}_{jk}$	=	branch transformation matrix of the j^{th} mixer output stream in the k^{th} mixer equation
$\mathbf{N}^{(j)}$	=	vector of species flows in stream j
$\mathbf{q}^{(k)}$	=	vector of constants in mixer equation k
R	=	number of reactions
r_r	=	rate of reaction r, moles per unit time
S	=	number of species in the flow sheet
s_{ij}	=	element of the mixer equation structural matrix
$\mathbf{T}^{(j)}$	=	separator transformation matrix, S × S
$t_{ss}{}^{(j)}$	=	element of transformation matrix $\mathbf{T}^{(j)}$
X_k	=	conversion of species k
σ_{sr}	=	stoichiometric coefficient of species s in reaction r

LITERATURE CITED

Agarwal, J. C., I. V. Klumpar, and F. D. Zybert, "A Simple Material Balance Model," *Chem. Eng. Progr.*, **74**, 68 (1978).

Barkley, R. W., and R. L. Motard, "Decomposition of Nets," *Chem. Eng. J.*, **3**, 265 (1972).

Christensen, J. H., and D. F. Rudd, "Structuring Design Computation," *AIChE J.*, **15**, 94 (1969).

Crowe, C. M., and M. Nishio, "Convergence Promotion in the Simulation of Chemical Processes—The General Dominant Eigne Value Method," *ibid.*, **21**, 528 (1975).

Genna, P. L., and R. L. Motard, "Optimal Decomposition of Process Net Works," *ibid.*, 656 (1975).

Hutchison, H. P., and M. E. Leesley, "A Balanced Approach to Process Design by Computer," *Computer Aided Design*, **5**, 228 (1973).

Hutchison, H. P., "Plant Simulation by Linear Methods," *Trans. Inst. Chem Engrs.*, **52**, 287 (1974).

Kneile, R. G., "Solution of Material Balance Problems for Process Design," Ph.D. thesis, Purdue Univ., W. Lafayette, Ind. (Dec., 1975).

Motard, R. L., H. McLee, and R. W. Barkeley, *CHESS-Chemical Engineering Simulation System: User's Guide*, Technical Publishing Company, Houston, Tex. (1969).

Motard, R. I., M. Shacham, and E. H. Rosen, "Steady State Chemical Process Simulation," *AIChE J.*, **21**, 417 (1975).

Nagiev, M. F., *The Theory of Recycle Process in Chemical Engineering*, Pergamon Press, Ltd., Oxford, England (1964).

The Ralph M. Parsons Company, "Determination Plant: Clean Boiler Fuels from Coal," *OCR R&D Rept. No.* 82, Interim Report No. 1 (1974).

———, "Commercial Complex Conceptual Design/Economic Analysis: Oil and Power by COED Based Coal Conversion," *ERDA R&D Rept. No. 144*, Interim Report No. 1 (Sept., 1975).

Pho, T. K., and L. Lapidus, "Topics in Computer-Aided Design: I. An Optimum Tearing Algorithm for Recycle Systems," *AIChE J.*, **19**, 1170 (1973).

Reklaitis, G. V., Class Notes, Purdue Univ., W. Lafayette, Ind. (1973).

Riegle, D. E., and P. H. Lin, "Matrix Signal Flow Graphs and an Optimum Top-Logical Method for Evaluating their Gains," *IEEE Trans. Circuit Thry.*, **19**, 427 (1972).

Rosen, E. M., "A Machine Computation Method for Performing Material Balances," *Chem. Eng. Progr.*, **58**, 69 (1962).

Sood, M. K., and G. V. Reklaitis, "User's Manual: Material Balance Program—version I," School of Chemical Engineering, Purdue Univ., W. Lafayette, Ind. (June, 1977).

Soylemez, S., and W. C. Seider, "A New Technique for Precedence Ordering Chemical Process Equation Sets," *AIChE J.*, **19**, 934 (1973).

Thatcher, C. M., *Fundamentals of Chemical Engineering*, pp. 356-358, C. E. Merrill Books, Columbus, Ohio (1962).

Upadhye, R. S., and E. A. Grens, II, "An Efficient Algorithm for Optimum Decomposition of Recycle Streams," *AIChE J.*, **18**, 533 (1972).

APPENDIX

Although an UMBP always has the correct number of specification, these specifications may be improper. Two easily recognized cases of such improper specifications are the internally recycling inert species, which neither enters nor leaves the system, and the completely recycled reactant that is not a key species in any of the reactions.

Case 1

Consider the simple flow sheet shown in Figure 3*a* which could represent an absorber and reboiler stripper pair involving absorbent S. For any known feed N^0 consisting of species *A*, *B*, but no S and with known split fractions for both separators, the problem clearly is naturally specified. From the module equations

$$\mathrm{N}^{(3)} = \mathrm{T}^{(3)}\mathrm{N}^{(2)} = \begin{bmatrix} 0 & & \\ & 0 & \\ & & 1 \end{bmatrix} \mathrm{N}^{(2)} \qquad \text{for SEP 2}$$

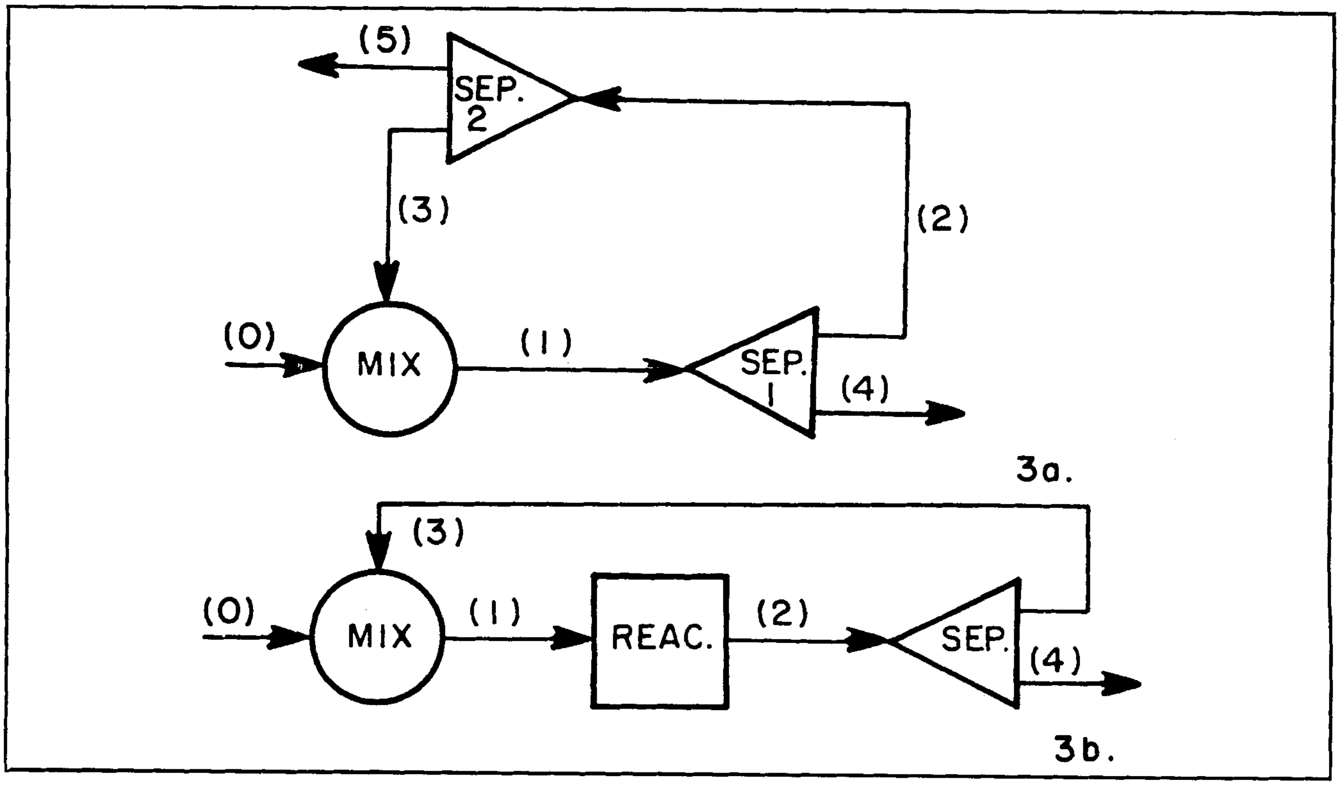

Fig. 3. Flow sheets for improper specification examples.

$$\mathbf{N}^2 = \mathbf{T}^{(2)}\mathbf{N}^{(1)} = \begin{bmatrix} 0 & & \\ & 1 & \\ & & 1 \end{bmatrix} \mathbf{N}^{(1)} \qquad \text{for SEP 1}$$

and

$$\mathbf{N}^{(1)} = \mathbf{N}^{(3)} + \mathbf{N}^{(0)}$$

for the mixer. One can construct the equation array

$$\begin{bmatrix} \mathbf{I} & -\mathbf{T}^{(3)} & 0 \\ 0 & \mathbf{I} & -\mathbf{T}^{(2)} \\ -\mathbf{I} & 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{N}^{(3)} \\ \mathbf{N}^{(2)} \\ \mathbf{N}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{N}^0 \end{bmatrix}$$

Upon block elimination, this system reduces to

$$\begin{bmatrix} \mathbf{I} & -\mathbf{T}^{(3)} & 0 \\ 0 & \mathbf{I} & -\mathbf{T}^{(2)} \\ 0 & 0 & \mathbf{I} - \mathbf{T}^{(3)}\mathbf{T}^{(2)} \end{bmatrix} \begin{bmatrix} \mathbf{N}^{(3)} \\ \mathbf{N}^{(2)} \\ \mathbf{N}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{N}^0 \end{bmatrix}$$

This array has maximum row rank providing $[\mathbf{I} - \mathbf{T}^{(3)}\mathbf{T}^{(2)}]$ has maximum row rank. It obviously does not. Thus, the corresponding mixer equation

$$[\mathbf{I} - \mathbf{T}^{(3)}\mathbf{T}^{(i)}]\,\mathbf{N}^{(1)} = \mathbf{N}^0$$

or

$$\begin{bmatrix} 1 & & \\ & 1 & \\ & & 0 \end{bmatrix} \mathbf{N}^{(1)} = \mathbf{N}^0$$

cannot be solved for the flow of solvent *S* in stream (1). To resolve the dilemma, the flows of *S* must be separately specified, or a makeup and loss of *S* from the system must be shown.

Case 2

Consider the elementary recycle loop shown in Figure 3*b*. If the feed stream, the stoichiometry of the reaction, and a key reactant conversion, as well as the split fractions in the separator, are all specified, the problem is a UMBP. Suppose the reaction is $A + B \rightarrow C$, with 25% conversion of *B*. All of *A*, 40% of the *B*, and no *C* are recycled. The module equation will be

$$\mathbf{N}^{(3)} = \mathbf{T}^{(3)}\mathbf{N}^{(2)} = \begin{bmatrix} 1 & & \\ & 0.4 & \\ & & 0 \end{bmatrix} \mathbf{N}^{(2)} \qquad \text{for the separaor}$$

$$\mathbf{N}^{(2)} = \mathbf{C} \quad \mathbf{N}^{(1)} = \begin{bmatrix} 1 & -0.25 & 0 \\ 0 & 0.75 & 0 \\ 0 & 0.25 & 1 \end{bmatrix} \mathbf{N}^{(1)} \qquad \text{for the reactor}$$

and

$$\mathbf{N}^{(1)} = \mathbf{N}^{(3)} + \mathbf{N}^{(0)} \qquad \text{for the mixer}$$

The corresponding array of equations will be

$$\begin{bmatrix} \mathbf{I} - \mathbf{T}^{(3)} & 0 \\ 0 \quad \mathbf{I} & -\mathbf{C} \\ -\mathbf{I} \quad 0 & \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{N}^{(3)} \\ \mathbf{N}^{(2)} \\ \mathbf{N}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{N}^0 \end{bmatrix}$$

which, when reduced, becomes

$$\begin{bmatrix} \mathbf{I} - \mathbf{T}^{(3)} & 0 \\ 0 \quad \mathbf{I} - & \mathbf{C} \\ 0 \quad 0 & \mathbf{I} - \mathbf{T}^{(3)}\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{N}^{(3)} \\ \mathbf{N}^{(2)} \\ \mathbf{N}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{N}^0 \end{bmatrix}$$

This array has maximal row rank if $[\mathbf{I} - \mathbf{T}^{(3)}\ \mathbf{C}]$ has maximal row rank. It does not, and thus the corresponding mixer equation cannot be solved for the flow of *A* in stream (1):

$$[\mathbf{I} - \mathbf{T}^{(3)}\mathbf{C}]\ \mathbf{N}^{(1)} = \mathbf{N}^0 = \begin{bmatrix} 0 & 0.25 & 0 \\ 0 & 0.7 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{N}^{(1)}$$

The problem can be rectified by either allowing some loss of *A* with the product stream or by choosing *A* as key reactant. For instance, if 25% conversion of *A* is specified, then the mixer equation becomes

$$\begin{bmatrix} 0.25 & 0 & 0 \\ 0.1 & 0.6 & 0 \\ 0 & 0 & 1 \end{bmatrix} \mathbf{N}^{(1)} = \mathbf{N}^0$$

Both of these situations are easily recognized in problem formulation. However, if they are not caught prior to solution, our material balance approach will terminate when a zero row is generated in the course of the solution of the essential mixer output equations.

Manuscript received December 21, 1977; revision received August 18, and accepted September 6, 1978.