

A Parallel Computing Strategy for Multicomponent Separation Calculations

Alfred J. O'Neill
Daniel J. Kaiser
Mark A. Stadtherr

Department of Chemical Engineering
University of Illinois
1209 W. California St.
Urbana, IL 61801
U.S.A.

Abstract

Multicomponent separation calculations, whether single columns or complexly interlinked systems of multiple columns, represent large scale computational problems and are thus attractive applications for the parallel computing architectures of machines such as the Cray-2 supercomputer and the Alliant FX/8. Frequently, these problems are formulated as a large set of nonlinear equations solved by a Newton-Raphson or comparative successive linearization technique. They thus require the solution of a very large system of sparse linear equations which often represents a large fraction of the overall computing time. We present an efficient parallel technique for the solution of such large sparse systems, based on an ordering of equations by plate. The resulting linear systems take on an almost block tridiagonal (BTD) form, with off-BTD blocks resulting from recycle streams or tower sidestreams. We present extensions to current multiprocessor schemes for BTD systems which permit a straightforward solution for single or interlinked distillation columns. Pivoting schemes and strategies for efficient matrix partitioning during the solution stage are also outlined.

Introduction

The solution of the large sparse linear equation systems which arise in the solution of systems of interlinked distillation columns, or even single columns, is a critical factor in determining computational efficiency. We present an efficient parallel technique for the solution of such large sparse systems on multiprocessor machines such as the Cray-2 and the Alliant FX/8. An equation-based solution procedure for single distillation columns based on ordering the equations by plate leads to a sparse, narrow-banded system of linearized equations. Such an approach was proposed by Naphali and Sandholm (1971) and has proven very successful. This formulation may also be used as the basis for solution methods for interlinked systems of columns, as shown by Hofeling and Seader (1978) and Stadtherr and Malachowski (1982).

Single Columns

Consider the case of an n plate column separating c components, with plate 1 a condenser and plate n the reboiler. For a single column with no sidestreams the matrix of partial derivatives that arises takes the shape of a block tridiagonal matrix (see Figure 1). The structure of the blocks is known and shown here. Here X represents a nonzero submatrix within the blocks, whereas I stands for the identity matrix and 0 for a zero submatrix. The A_i matrix is the matrix of derivatives of the functions of plate n with respect to the variables on plate $n-1$, and thus is zero below the diagonal. More importantly for the purpose of avoiding fill-in during the computation, the partial derivatives of the equilibrium functions on plate n with respect to the components on plate $n-1$ are all zero, resulting in zero values in all columns left of the center column. Thus the A matrix has size and structure

$$A_i = \begin{matrix} & & c & 1 & c \\ c & & 0 & 0 & -I \\ 1 & & 0 & X & X \\ c & & 0 & 0 & 0 \end{matrix}$$

Here the numbers and letters above and to the left of the matrix indicate the sizes of the individual sub-blocks, and c again represents the number of components. The first row represents the partial derivatives for the material balance, followed by rows for the enthalpy balance derivative and equilibrium relationship derivatives, respectively.

$$R = \begin{bmatrix} I_4 & * & 0 & 0 & 0 & 0 \\ * & I_5 & 0 & * & 0 & 0 \\ * & 0 & I_6 & * & 0 & 0 \\ 0 & 0 & * & I_7 & 0 & * \\ 0 & 0 & * & 0 & I_{12} & * \\ 0 & 0 & 0 & 0 & * & I_{13} \end{bmatrix}$$

Figure 2: Reduced Matrix R for $n = 16$

processor of a multi-processor machine, a significant speed-up in the most computationally demanding stage of the solution is achieved. Perhaps more importantly, fill-in of empty rows which occurs during the reduction of non-zero elements in off-diagonal blocks is now contained within each partition, substantially reducing the number of floating-point operations required during the reduction. The BTD matrix which arises during the Napthali-Sandholm formulation is ideally suited for such a parallel technique as the blocks responsible for fill-in during the solution stage, for systems with a reasonable number of components, are demonstrably small.

A trade-off exists between the desire to reduce fill-in during the calculation stage and the most efficient use of the available processors. This trade-off occurs at the partitioning stage. The reduced matrix R, which is solved on a single processor prior to the parallel solution stage, contains $2(p-1)$ row blocks, where p is the number of partitions. Thus, if a 12 row matrix is divided into four partitions, one for each of four processors, the "reduced" matrix to be solved contains six rows. The time required to solve this matrix will far exceed that to reduce the partitioned blocks. A partitioning scheme which balances the need to reduce fill-in with the requirement for the most efficient use of the available processors can easily be coded given *a priori* information on the matrix size and structure.

Interlinked Distillation Columns

Straightforward modifications to Sameh's algorithm that facilitate handling off-diagonal blocks arising from sidestream and recycle flows permit the extension of the method to systems of interlinked

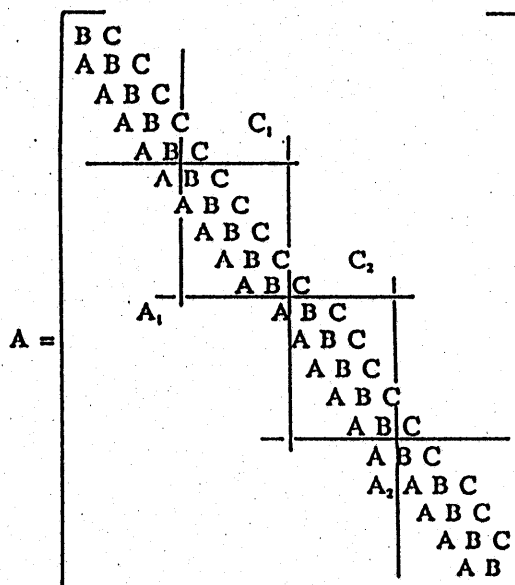


Figure 3: Block Structure of Jacobian Matrix for Interlinked System of 20 Trays

columns (Figure 3). Such systems typically contain a number of blocks far off the diagonal arising from sidestreams, recycles and pumparounds. In the trivial case where the off-BTD blocks fall in the columns defining any of the partitions (either the Y_i or W_i vector), no modification to the method is needed. These columns are included within the reduced matrix and a solution returned before the parallel solution of the partitions begins. In the case of blocks which fall between these columns (i.e. within the partitions themselves) no additional variables are introduced to the sub-systems solved on each processor and the solution may proceed normally.

Two options arise when off-diagonal blocks occur in positions outside any of the sub-systems defined by the partitions. In this case, additional unsolved variables are brought into any partition containing such blocks, and the sub-system can no longer be solved independently. For systems with a minimum number of off-BTD blocks it is advantageous to modify the reduced matrix R to include the row and column of the identified block. While this increases the size of the reduced matrix, it permits the parallel solution of the sub-systems to continue unhindered. For systems for which the number of off-BTD blocks which fall into this category is prohibitively large, the best approach is to intelligently re-partition the matrix. The criteria for such partitioning is no longer a precisely even division of labor between each processor, but rather

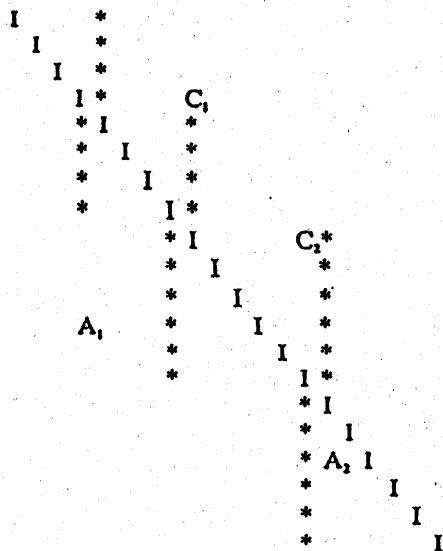


Figure 4: Re-partitioning of Matrix to Accommodate Off-Diagonal Blocks

the inclusion of those columns which contain the off-diagonal blocks (Figure 4). In the example shown, four blocks are displaced from the diagonal, only one of which would be solved by the reduced matrix in an even partitioning strategy (A_1 , as it lies in the Y_1 vector of the fourth partition). The inclusion of the remaining three columns in the matrix R would produce a reduced matrix of order nine, too large to permit the method to retain its economy. An advance re-partitioning of the matrix to accommodate the off-diagonal blocks, as shown, permits the reduced matrix to remain small and produces a load imbalance in the third and fourth processor of no more than fifty percent.

Conclusions

Given the availability of multiprocessor machines, such as the Alliant FX/8, Cray X-MP/48, and Cray 2, an algorithm to exploit the capabilities of these machines is clearly advantageous. The structure of the BTM matrix which arises during the Naphthali-Sandholm formulation, together with the favorable size of the off-diagonal blocks, make their algorithm well suited for a parallel solution technique. The case in which blocks arising from sidestreams and recycles may be handled, with little or no loss of efficiency, is an additional advantage of the method. Pivoting strategies, both *a priori* and concurrent with the calculations, are necessary to ensure diagonal dominance at every stage of the solution.

The development of concrete and general guidelines for efficient matrix partitioning will permit the reduced matrix to remain small in most cases of interest, ensuring an economical solution. A priority inclusion policy for columns and rows in the reduced matrix has been coded and demonstrates reasonable efficiency for a wide variety of cases.

Care must be taken in the case of systems with a large number of off-BTD blocks if the method is to remain viable. Load imbalance between the processors, if excessive, can sabotage efficiency. When possible any additional load should be assigned to the processors handling the first and last sub-systems (processors 1 and 4 in our example), since these partitions contain one less column (either a Y_1 or W_1 vector) requiring solution. As noted by Barry *et al* (1989), this inherent imbalance will cause processors 2 and 3 to perform 33% more computations than the remaining two during the retrieval phase. In order to facilitate a more equitable work balance among the four processors during the retrieval phase, portions of the Y_1 and W_1 vectors could be shared among processor tasks and thus copied into global memory. However, experiments concerning the trade-off between workload balancing and the minimization of shared data on Alliant FX/8 CEDAR prototypes have shown little advantage to such techniques for matrices which are not uncommonly large.

References

- Berry, M. W. and A. Sameh, "Multiprocessor Schemes For Solving Block Tridiagonal Linear Systems," *Int. J. Supercomputer Applic.*, 2, 3, 37 (1988).
- Hofeling, B. S. and J. D. Seader, "A Modified Naphthali-Sandholm Method for General Systems of Interlinked Multistaged Separators," *AIChE J.*, 24, 6, 1131 (1978).
- Naphthali, L. M. and D. P. Sandholm, "Multicomponent Separation Calculations by Linearization," *AIChE J.*, 17, 148 (1971).
- Sameh, A., "On Two Numerical Algorithms For Multiprocessors," *Proc. of NATO Adv. Res. Workshop on High-Speed Comp.*, New York, Springer-Verlag (1983).
- Stadtherr, M. A. and M. A. Malachowski, "On Efficient Solution of Complex Systems of Interlinked Multistage Separators," *Comput. Chem. Eng.*, 6, 121 (1982).