AN INTERACTIVE GRAPHIC INTERFACE TO ILLUSTRATE SPARSE MATRIX/VECTOR METHODS

D.C. Yu
Member

S.T. Chen
Student Member

F.L. Alvarado
Senior Member

Dept. of EECS
University of Wisconsin-Milwaukee
Milwaukee, WI

Dept. of ECE
University of Wisconsin-Madison
Madison, WI

ABSTRACT - An interactive and color graphical front-end package to assist in the study of sparse matrix/vector methods is presented in this paper. The package is written in C and can be installed on any 286 PC machine. It is designed using an open software architecture. With this package, students can study the sparse matrix/vector methods in a portable and highly user friendly environment. A description of the package, a detailed demonstration of the user interface capabilities, and the hardware requirements are provided in this paper.

KEYWORDS - Sparse Matrix Methods, Sparse Vector Methods, Education, PC, Color Graphics, User Interfaces.

INTRODUCTION

Sparse matrix/vector methods are invaluable tools in many power system computational applications. Sparse matrix/vector methods include techniques for node ordering [1-4,8], matrix factorization [1,18], sparse vector methods [5], partial refactorization [6,7], matrix inverse factors [8,9], matrix partition [8,9], and parallel computation [10-17]. However, many of the techniques are heuristic, which implies further improvement can always be made. For instance, the existing node ordering techniques can be divided into two groups. One group, which includes Scheme 2 [1] tries to minimize the degree of the node, which in turn minimizes the number of fills in the L matrix. The other group, which includes MLMD [3], tries to minimize the length of the path graph. This minimizes the number of operations in each branch of the path graph. In the fast forward (FF) and fast backward (FB) substitutions [5], partial refactorization (PR), and parallel computational applications, the algorithms belonging to the second group are better options. However, in the matrix factorization and the conventional forward/backward applications, the algorithms in group one are preferred alternatives.

90 WM 072-9 PWSR A paper recommended and approved by the IEEE Power Engineering Education Committee of the IEEE Power Engineering Society for presentation at the IEEE/PES 1990 Winter Meeting, Atlanta, Georgia, February 4 - 8, 1990. Manuscript submitted September 1, 1989; made available for printing January 5, 1990.

From the above explanation, it can be seen that a careful study of the objective and network characteristics is needed before a particular algorithm can be selected.

It is possible to gain insight into sparse matrix techniques by working with graphical illustrations along with computer program exercises. For example, the mechanism and advantages of the FF and FB approaches can be seen clearly if the path graph of the matrix is presented. The effects of different ordering methods can be understood more easily by showing the L matrix, the W matrix (the inversion of L), and the path graph. However, when the network size increases, it becomes more difficult to manually draw graphical representations. Thus, a computer graphical demonstration package dealing with sparse matrix/vector methods with a user-friendly and graphic-oriented I/O is highly desirable. This paper presents an interactive graphic computer package which facilitates the studies of the topology of the sparse matrix/vector methods by producing a user-friendly environment.

DESCRIPTION OF THE PACKAGE

This paper presents a comprehensive sparse matrix educational tool for manipulating the sparse matrices within 286 PCs running under the DOS operating system. This package has an open software architecture which allows students to add new routines developed using any programming language. This is true as long as the programs have a predetermined I/O format in a form of a simple ASCII file. Thus, this package enables students to study sparse matrices in a very flexible way. This package will be used in graduate courses at the University of Wisconsin — both in Madison and Milwaukee.

This sparse matrix/vector method package consists of six topics. These are: matrix factorization, path graph construction, partial refactorization, node ordering, FF/FB substitutions, and W-matrix partitioning. Each matrix operation is implemented as a module. The modularized structure makes any modification of the programs very easy to handle. Furthermore, a modularized structure helps students understand the relationship between the various matrix operations, even though execution speed might suffer. However, since the purpose of this package is primarily educational, clarity is considered more important than efficiency.

The concept of menu-driven interfaces is implemented to aid the man-machine interfaces in this package. In order to allow students to do comparative studies, four windows are created in the package. A comparative study could be comparisons of the effects of different methods for the same subject using the same network. The study could also be displays of different subjects for the same network. Each window is capable of displaying matrices, path graphs, and matrix partitions. Users can select any combination of these. The size of the display can also be changed.

We now describe the six "topics" that can be studied with this interactive package:

(1) Matrix factorization

Sparse-oriented matrix factorization of matrix A is a necessary step in solving the sparse linear equation $Ax = b$. This step is also known as LU decomposition, where $LU=A$. However, additional fills will also be generated in the LU matrices in this process. If $a(i,j)$ is zero, a fill will occur at $a(i,j)$ when $a(i,k)$, $i > k$, and $a(k,j)$, $j > k$, are nonzeros. In this module, when a student enters a permutation vector, the corresponding LU matrices will be presented automatically. Figure 1 illustrates the LU matrices of a 20 node example [5] which will be used for illustrative purposes throughout this paper. The original elements are marked as "X", the fills are marked as "O", and the number of original elements and the number of fills are also shown. By placing the cursor on any diagonal term in the LU, the fills caused by that particular node are highlighted. Furthermore, students can use a mouse to interchange nodes in the network or to modify the network. The fills are updated automatically. These abilities give students a better graphical explanation about the causes of fills.

(2) Path graph construction

A factorization path for any node k is an ordered list of nodes starting at k . The list contains the number of the node of the first nonzero element in column k of L . This new node is then taken as a column and the process is repeated until the last node is reached. The union of all the singleton paths of the network is called the path graph of the network. The path graph introduced in [5], is one of the most important breakthroughs in advancing sparse vector methods. It graphically explains why solutions with sparse independent vectors involve only a few elements of the lower triangular factor. Likewise, it shows that only a few elements of the upper triangular factor need to be operated on in solutions for limited solution vectors. It illustrates why changes to a limited number of elements in the matrix A produces limited changes in the corresponding LU matrices. Furthermore, the path graph is very useful in developing algorithms for solving linear equations in a parallel environment [14-17].

After constructing the LU matrices, the path graph can be easily generated in this module as shown in

Figure 2. Zoom-in, shifting, and scaling functions are also provided to accommodate larger path graphs. Users can also interchange the nodes in the network or modify the network. The path graph is updated automatically. Figure 3 shows the new path graph due to the interchange between nodes 2 and 15.

Figure 1. The LU matrices of the 20-node example (Scheme 2)

Figure 2. The path graph of the 20-node example (Scheme 2)

Figure 3. The path graph of the 20-node example due to the interchange between nodes 2 and 15

(3) Partial refactorization

Partial refactorization (PR) methods update the LU factors of a matrix to reflect changes in some of its elements. With the aid of a path graph, the PR techniques can be understood easily. In this package, if students want to study the mechanism of the PR, he/she can use a mouse to select the given function. Then the students can select the elements needed to be modified in the LU matrices. Consequently, the associated paths in the path graph are highlighted as shown in Figure 4. Since each column and row in the LU, which corresponds to the nodes in the highlighted paths, require refactorization. These columns and rows are highlighted as shown in Figure 5, where the affected elements in these columns and rows are displayed in boldfaced "X"s (for original nonzeros) and in solid circles (for LU fills).

Figure 4. Highlighting the nodes affected by the modifications in row 2/column 2

Figure 5. Highlighting the elements in the LU matrices affected by the modifications in row 2/column 2

(4) Node ordering

The original goal of node ordering is to reduce excessive fills in the LU decomposition. However, some of ordering algorithms accept a few more fills in L in order to reduce the length of the path graph. Applying different ordering techniques to the same network will result in different node sequences. As a result, the path graph, the LU matrices, and the W matrix will also be different.

In the proposed ordering module, many existing ordering algorithms are provided. Users are encouraged to add their own techniques to it. These algorithms plus the user-friendly environment allow students to do thorough and/or comparative studies concerning the effects of various ordering methods on the network. Figure 6 shows the LU matrices, the W

Figure 6. The LU matrices, the W matrix, and the path graph of the example (Scheme 2)

matrix, and the path graph of a 20 node system. The results are based on Scheme 2. Figure 7 shows the LU matrices and the path graphs. Figure 8 shows the W matrices and the path graphs. These results are based on Scheme 2 and the MLMD. These show that MLMD generates a shorter path graph; however, it generates more fills in LU matrix. These also show that the MLMD generates less fills in the W matrix. It is our opinion, these displays and the ability to allow students to experiment with new ideas, help students to gain insight into ordering methods.

(5) FF/FB substitutions

In many power applications, the right-hand side b

vector of linear equations is sparse. However, the solution vector x is not generally sparse. If b is sparse, only a subset of the columns of L is needed for forward substitutions. This is called the FF. If only certain elements of x are of interest, only a subset of the rows of U is needed for the backward substitutions. This is called the FB. The mechanism of the FF and FB can be illustrated by the path graph. Assume a nonzero b element corresponds to a node in a path, the FF substitution originated from that b element can be interpreted as the nodal equation of a directed network in which each node is affected by only the lower-numbered nodes in the subgraph. The value of a node in FF can not be calculated until the values of all the lower-numbered nodes in the

Figure 7. Comparisons of the LU matrices and the associated path graphs under different ordering algorithms (Scheme 2 and MLMD)

Figure 8. Comparisons of the W matrices and the associated path graphs under different ordering algorithms (Scheme 2 and MLMD)

subgraph are known. The branches above a junction, however, can be computed in any order. A similar rule applies to the FB.

In the proposed FF/FB mode, after students select the nonzero elements in the b vector, the associated paths in the path graph are highlighted as shown in Figure 9. The corresponding columns and rows in the LU matrices are also highlighted as shown in Figure 10. In the display, the affected elements in these columns and rows are indicated by boldfaced "X"s (for original nonzeros) and solid circles (for LU fills). These results are based on Scheme 2 ordering.

Since various ordering algorithms have different effects on the path graph, their effects on the FF/FB will be different also. The FF/FB mode allows students to compare the FF/FB based on different ordering schemes. Figures 11 and 12 show the results obtained from MLMD.

Figure 9. Highlighting the nodes required by the FF/FB (Scheme 2)

Figure 10. Highlighting the corresponding elements in the LU matrices required by the FF/FB (Scheme 2)

Figure 11. Highlighting the nodes required by the FF/FB (MLMD)

(6) W -matrix partitioning

Solving linear equations is a necessary step in many power system problems. How to make this process more efficient is always an important task. If we view the conventional LU decomposition and forward/backward substitutions as the first generation of development in this area, then the path graph, FF/FB, and PR are considered the second generation. The third generation probably will be the undergoing developments of solving this problem in concurrent machines.

There are several suggested approaches to this subject [10-17]. One approach is to use matrix inverse factors [8,9]. The inverse factors matrix is known as the W matrix. It is shown in [8] that linear equations can be solved in three serial steps as follows:

$$\begin{aligned} z &= W b \\ y &= D^{-1} z \\ x &= W^t y \end{aligned}$$

All of the multiplications required by the above equations may be performed in parallel; however, not all of the additions can be performed simultaneously. Thus, parallel operations, based on the W matrix, in the direct solution of the sparse matrix problem can be explored. However, the W matrix may have too many nonzero elements for the proposed approach to be practical for a very large system. The solutions to this are (1) reordering the nodes to enhance the sparsity in W , and (2) partitioning W matrix, with the solution to be carried out in one step per partition, so that the nonzeros in each partition are less.

Different ordering algorithms produce different numbers of fills in the W matrix. The structure of W is implied by the path graph. For example, the nonzero entries in column i of W are given by the nodes that are included in the path of node i . Also, the nonzero entries in row i of W are given by the nodes in which path node i is included. Thus, ordering algorithms, such as MLMD and Scheme W, which

reduce the length of the path will produce less fills in W . Since the number of partitions equals the number of serial steps in the direct solution, reducing the number of required partitions is an essential task in this proposed parallel solution method. Reference [9] shows that ordering techniques such as MLMD are useful in reducing the number of partitions. Also in [9], several partition algorithms are proposed. These algorithms show that by allowing a few fills in each partition, the number of partitions can be reduced substantially.

The relationship among the various ordering algorithms and their associated path graphs, W matrices, and W partitions can be demonstrated very easily. Figure 8 shows the path graphs and W matrices based on Scheme 2 and the MLMD. If students want to see how an individual path will affect the W matrix, he/she can use a mouse indicating a particular node. Not only will the subsequent nodes in the path be highlighted as previously shown in Figure 4, but the affected columns and rows in the W matrix will be highlighted as shown in Figure 13. In the display, the affected elements in these columns and rows are indicated by boldfaced "X"s (for original nonzeros), solid circles (for LU fills), and solid triangles (for W fills). Furthermore, the partition mode will allow students to experiment with a varying number of fills in each partition to study the effects on the number of partitions. This display is shown in Figure 14.

HARDWARE AND SOFTWARE CONSIDERATIONS OF THE PACKAGE

This package can be installed on any 286 PC machine. An IBM EGA graphics adapter is required in this package, which provides 640×350 resolution and 16 different colors. A mouse is used to aid the man-machine interface, and a keyboard is used for conventional data entry.

This package is designed as a menu-driven software. All the modules and functions can be easily accessed from proper menus in a hierarchical manner. On-line help menus, which give a brief explanation to each of the module and functions, are also provided.

The graphics programs in this package are written in C language. However, this package is designed to accept analytical programs written in almost any language as long as the programs have a predetermined I/O format in a form of a simple ASCII file. For example, work is underway to interface this package with the Sparse Matrix Manipulation System (SMMS) [19]. The SMMS is a collection of independent routines which handle sparse matrices. All the routines in the SMMS are written in PASCAL.

A print screen function is provided in this package to enable students to get a hard copy of the display. Currently, the maximal size of the matrix is set at 40 by 40. However, with the shift function, changes can be made very easily to accommodate larger-sized networks.

Figure 12. Highlighting the corresponding elements in the LU matrices required by the FF/FB (MLMD)

Figure 13. Highlighting the corresponding elements in the W matrix affected by the selected path in the path graph

CONCLUSION

Sparse matrix/vector methods are very important tools in many power system applications. Students who are interested in power engineering certainly should understand this subject well. An interactive and color graphical package to assist the study of sparse matrix/vector methods has been presented. This package helps students visualize sparse matrices and helps students gain intuitive insight into sparse matrix operations. This graphics front-end package is designed so students are encouraged to add to it routines developed using almost any programming language. The proposed package runs on any 286 based PC. Six "topics" are currently included: matrix factorization, path graph construction, partial refactorization, node ordering, FF/FB substitutions, and W -matrix partitioning. Each matrix operation is implemented as a module. The modularized structure makes any modification of the programs very easy to handle. Using this package, students can study the sparse matrix/vector methods in a portable and highly user-friendly environment. Here, the information received is much easier to comprehend.

Figure 14. Comparisons of W matrices under different number of allowed fills in each partition

REFERENCES

- W.F. Tinney and J.W. Walker, "Direct Solutions of Sparse Network Equations by Optimally Ordered Triangular Factorization," *Proc. of IEEE*, Vol. 55, No. 11, pp. 1801-1809, Nov. 1967.
- A. Gomez and L.G. Franquelo, "Node Ordering Algorithms for Sparse Vector Method Improvement," *IEEE Trans. on PWRs*, Vol. 3, No. 1, pp. 73-79, Feb. 1988.
- R. Betancourt, "An Efficient Heuristic Ordering Algorithm for Partial Matrix Refactorization," *IEEE Trans. on PWRs*, Vol. 3, No. 4, pp. 1181-1187, Aug. 1988.
- A. Gomez and L.G. Franquelo, "An Efficient Ordering Algorithm to Improve Sparse Vector Methods," *IEEE Trans. on PWRs*, Vol. 3, No. 4, pp. 1538-1544, Nov. 1988.
- W.F. Tinney, V. Brandwajn, and S.M. Chan, "Sparse Vector Methods," *IEEE Trans. on PAS*, Vol. 104, No. 2, pp. 295-301, Feb. 1985.
- O. Alsac, B. Stott, and W.F. Tinney, "Sparsity-Oriented Compensation Methods for Modified Solutions," *IEEE Trans. on PAS*, Vol. 102, No. 5, pp. 1050-1060, May 1983.
- S.M. Chan and V. Brandwajn, "Partial Matrix Refactorization," *IEEE Trans. on PWRs*, Vol. 1, No. 1, pp. 193-200, Feb. 1986.
- M.K. Enns, W.F. Tinney, and F.L. Alvarado, "Sparse Matrix Inverse Factors," Paper 88 SM 728-8, 1988 IEEE Summer Power Meeting, Portland, Oregon, July 24-29, 1988.
- F.L. Alvarado, D.C. Yu, and R. Betancourt, "Partitioned Sparse A^{-1} Methods," Paper 89 SM 679-2-PWRs, 1989 Summer Power Meeting, Long Beach, CA, July 9-14 1989.
- O. Wing and J.W. Huang, "A Parallel Triangulation Process for Sparse Matrices," *Proc. Int. Conf. Parallel Processing*, Wayne State University, pp. 207-214, 1977.
- F.L. Alvarado, "Parallel Solution of Transient Problems by Trapezoidal Integration," *IEEE Trans. on PAS*, Vol. 98, pp. 1080-1090, May-June 1979.
- F.M. Brasch Jr., J.E. Van Ness, and S.C. Kang, "Design of Multiprocessor Structures for Simulation of Power System Dynamics," EPRI Report, EL-1756, Project 1355-1, March 1981.
- Beny Neta, "LU Factorization on Parallel Computers," *Comp. & Maths with Appls*, Vol. 11, No. 6, pp. 573-579, 1985.
- R. Betancourt and F.L. Alvarado, "Parallel Inversion of Sparse Matrices," *IEEE Trans. on PWRs*, Vol. 1, No. 1, pp. 74-81, Feb. 1986.
- Ali Abur, "A Parallel Scheme for the Forward/Backward Substitution in Solving Sparse Linear Equations," *IEEE Trans. on PWRs*, Vol. 3, No. 4, pp. 1471-1478, Nov. 1988.
- D.C. Yu and H. Wang, "A New Approach for the Forward and Backward substitutions of Parallel Solution of Sparse Linear Equations - Based on Dataflow Architecture," *Proceeding of PICA'89*, Seattle, WA., pp. 9-15, May 1-5, 1989.
- D.C. Yu and H. Wang, "A New parallel LU Decomposition Method," Paper 89 SM 680-0-PWRs, 1989 Summer Power Meeting, Long Beach, CA, July 9-14 1989.
- I.S. Duff, A.M. Erisman, and J.K. Reid, "Direct Methods for Sparse Matrices," Oxford Science Publications, Clarendon Press, Oxford, 1986.
- F.L. Alvarado, "The Sparse Matrix Manipulation System," SIAM Symposium on Sparse Matrices, Salishan Lodge, Gleneden Beach, Oregon, May 22-24, 1989.

David C. Yu is an assistant professor in the Dept. of EECS at the UW-Milwaukee.

Shin-Tzo Chen is a graduate assistant and is a doctoral student at the UW-Milwaukee.

Fernando L. Alvarado is a Professor in the Dept. of ECE at the UW-Madison.