Multi-Processor Cholesky Decomposition of Conductance Matrices

Trevor Maguire

Abstract – Electromagnetic Transients (EMT) Simulation based on the Dommel algorithm requires a solution for the node voltage vector [v] in the equation [G][v] = [In] in each time-step. In general, [G] is a symmetric positive definite conductance matrix with fixed sparsity fill, but the content of the filled locations changes with switching. [In] is the vector of nodal injections. Accordingly, one approach for solving for [v] in real-time simulation involves sparse decomposition of [G] in each time-step using the Cholesky method followed by sparse forward and backward operations on [In] to produce [v] on a single processor core.

Initially, the decomposition, forward and backward operations appear to be poorly suited for parallel computation because of the serial nature of the required operations.

However, based on work in other areas of science, this paper describes a practical method for using multiple processor cores for solving the overall nodal equations [G][v] = [In] even when all nodes represented in the sparse [G] are connected by conductances.

Keywords: Electromagnetic Transients Simulation, Cholesky Decomposition, Parallel.

I. Introduction

Electromagnetic Transients (EMT) Simulation based on the Dommel algorithm requires a solution for the node voltage vector [v] in the equation [G][v] = [In] in each time-step. In general, [G] is a symmetric positive definite conductance matrix with fixed sparsity fill, but the content of the filled locations changes with switching. [In] is the vector of nodal injections. Accordingly, one approach for solving for [v] in real-time simulation involves sparse decomposition of [G] in each time-step using the Cholesky method, followed by sparse forward and backward operations on [In] to produce [v] on a single processor core.

Two factors indicate toward the future use of multiple processor cores for the decomposition of the connected sparse matrices [G] used in EMT simulation.

First, the study of smart grid technology often requires the simulation of lower voltage portions of the grid where transmission lines are shorter. In such portions of the grid, the travel times of the transmission lines can often be less than a typical time-step which precludes representing the lines with traveling wave line models. Without traveling wave line models, it is less likely that the overall conductance matrix [G] can be replaced by several smaller conductance matrices. Therefore, there is a trend toward modeling networks which require larger connected conductance matrices [G].

Second, processor clock speeds are no longer increasing significantly from year to year.

Consequently, it is expected that the use of multiple processor cores will be required in future real-time EMT simulation in order to decompose and do the forward and backward operations for the ever larger conductance matrices while continuing to use existing or smaller time-steps.

In other areas of science, sparse positive definite symmetric matrices with dimensions in the 10s of thousands [1] are partitioned so that parallel Cholesky decomposition can be applied using 10s or 100s of processor cores. This paper describes initial efforts to adapt the methods used in these other areas to provide parallel decomposition of conductance matrices in EMT simulation.

As one adaptation of the methods, the present paper describes a practical method for using multiple processor cores for solving the overall nodal equations when the network contains multiple embedded subnetworks, all connected by conductances. In this adaptation, one processor core is used to perform calculations related to each subnetwork and another processor core performs calculations related to the overall network matrix in which the subnetworks are embedded.

An example of such a subnetwork, described in this paper, is a 12-pulse HVDC valve group model which includes 12 valves, three three-winding transformers, two bypass switches, two disconnects, a fault branch and a reactor. The model contains (N=) 10 inner nodes, (M=) 5 perimeter nodes and 17 switches. The N inner nodes are connected by conductances only to other inner nodes and/or to the M perimeter nodes.

The conductance matrix for the example valve group model is shown in Figure 1. The "A" area of the matrix is associated with the 10 inner nodes while the "D" area is associated with the 5 perimeter nodes.

[1]T.L. Maguire is with RTDS Technologies Inc., Winnipeg, Canada (e-mail: tlm@rtds.com).

Paper submitted to the International Conference on Power Systems Transients (IPST2011) in Delft, the Netherlands June 14-17, 2011

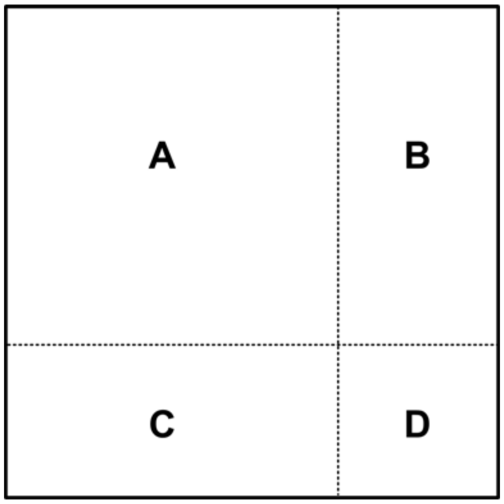

Fig. 1. Conductance Matrix of the HVDC Valve Group Model

It is well known that the dimension of the overall network conductance matrix [G] can be reduced in dimension by elimination of the N inner nodes of each subnetwork from the overall network conductance matrix [G]. The elimination operation requires the preparation of an overlay conductance matrix [Go] of dimension MxM for addition into the reduced dimension overall network matrix [Gr] in each time-step. The computation of the overlay matrix [Go] is according to (1).

$$[Go] = [D] - [C][A]^{-1}[B] \qquad (1)$$

In theory, the computation of [Go] using (1) could be performed in each small time-step by a subnetwork processor core and then sent to the main network processor. However, the inversion of the eliminated node area "A" in Figure 1 is computationally intensive and it is better to avoid the matrix inversion in real-time simulation.

Also, in theory, the overlay matrix [Go], to be added to the perimeter node area of the reduced overall network matrix [Gr], could be prepared in advance of the real-time simulation for each possible combination of switch states. However, computing 2^{17}=131,072 matrices in advance of the simulation and storing them for quick access in real-time EMT simulation is not a practical approach and does not support the use of branches with conductance values that can change over a continuous range.

The difficulties associated with using the [Go] matrix calculated using (1) led to a search for other available techniques. Fortunately, the large dimensions of the matrices in other areas of science [1] makes it imperative in those areas that multiple processors must be brought to bear on the solution of equations of form similar to [G][v] = [In]. This paper describes initial efforts to adapt the experience of other areas to the EMT simulation problem.

In order to avoid using an excessively large number of equations in this paper, the Appendix contains typical "C" language code for a right-looking Cholesky decomposition and associated forward and backward operations. These operations involve using columns F to L inclusive of the lower triangle of a symmetric positive definite conductance matrix [G] and a nodal injection vector [In]. The operations described in the Appendix are referred to in this paper by notation: DECOMP(F:L), FORWARD(F:L) and BACKWARD(L:F).

As an example of the use of the notation, when the equation [G][v] = [In] is solved on one processor, the sequence of operations required is a DECOMP(1:d) operation followed by a FORWARD(1:d) operation and a BACKWARD(d:1) operation where "d" is the dimension of [G]. This converts the nodal injection vector [In] into the node voltage vector [v].

II. Background

Fortunately, the symmetric positive definite matrix equations solved in other areas of science [1][2] are relatively large compared to those typically used in power system EMT simulations. Such other matrices [1] can be of dimensions in excess of several 10s of thousands. This has required the development of techniques in those scientific areas which enable a single large connected positive definite symmetric matrix to be decomposed by 10s or 100s of processors acting in parallel. The present paper cannot begin to catalog the numerous techniques that have been developed. However, the "spectral nested dissection" [2] technique is reviewed here because of it's apparent applicability to our EMT problem.

The "spectral nested dissection" [2] is one example of the "nested dissection" techniques that recursively divide the nodes of the matrix into groups in order to determine a good parallel ordering of the matrix such that it can be factored efficiently by parallel processors.

This review of the "spectral nested dissection" technique begins with the description of a constituent "spectral dissection" [2] technique which divides the solution of [G][v] = [In] between two processors. In this technique, a set of nodes is determined, the removal of which divides the overall matrix [G] into two disconnected parts. In the terminology of graph theory, the set of nodes constitute a "vertex separator". A good "vertex separator" is one that has a small number of nodes and separates the remaining nodes into two groups of approximately the same size.

In spectral dissection [2], a "Laplacian" matrix [Q] is prepared corresponding to the particular symmetric positive definite matrix [G] being considered. The matrix [Q] has the same dimension as the corresponding [G]. With respect to content, the matrix [Q] has diagonal elements of value "n" where "n" is equal to the number of non-zero off-diagonal elements in the particular corresponding row (or column) of [G]. The matrix [Q] also has off-diagonal elements of value -1 wherever the corresponding [G] matrix has an off-diagonal element that is non-zero. The remaining elements of [Q] are equal to zero. Assuming that all nodes in the matrix [G] are connected together through conductance, the first eigenvalue of [Q] is 0.0 and the remaining eigenvalues are positive. The smallest positive eigenvalue has a corresponding eigenvector referred to as the “Fiedler” vector [1]. The median value of the elements of the Fiedler vector can readily be found. Also, each node number in the matrices [Q] and [G] has a corresponding element in the Fiedler vector. The separation algorithm [1] proceeds by identifying a group of nodes X' which have corresponding values in the Fiedler vector which are less than the median value of the Fiedler vector. Remaining nodes, not in the group X', are placed in another group, Y'. If several elements of the Fiedler vector are equal to the median, corresponding nodes are shifted from one group to the other group until the number of nodes in each group differs by, at most, 1. The [G] matrix is thus partitioned into two groups of nodes, X' and Y'. The algorithm proceeds by finding a subset of X' and a subset of Y' respectively referred to herein as X” and Y”. For a node in X' to be in the subset X”, the node in X' must share a non-zero off-diagonal entry (a g connection) in the [G] matrix with a node in the Y' node group. Nodes in the Y' group gain membership in the Y” subset in a similar manner. In graph theory terminology, the nodes X” and Y” (vertices) and the g connections (edges) form a bipartite graph. A typical bipartite graph is shown in Figure 2 with the X” node (vertex) group on the left and the Y” node (vertex) group on the right joined by g connections (edges) illustrated as lines. Either the X” group or the Y” group can be used as a “vertex separator”, the removal of which partitions [G]. However, often there exists a vertex separator with some nodes in X” and some nodes in Y” which contains fewer nodes than either the X” group or the Y” group. This minimum size vertex separator is referred to as the “minimum cover” of a bipartite graph and is found in conjunction with finding the “maximum matching” of the bipartite graph [1].

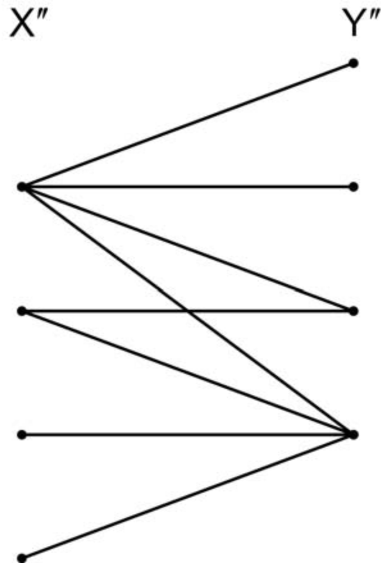

Fig. 2. A Typical Bipartite Graph

In Figure 3a), the vertex separator nodes, re-ordered into the largest node number locations of the matrix are illustrated as the X/Y portion of the matrix. The two separated node groups, X and Y, are associated with the portion of the matrix marked respectively as X and Y that are located to the left of the X/Y portion. The nodes in each of the separated groups are re-numbered so as to be consecutively placed. The separated X and Y node groups are only connected through the vertex separator nodes X/Y. This is illustrated in Figure 3a) by the absence of matrix entries in the portion of the matrix marked “nil”. Due to the lack of connection in the matrix to the left of the separator nodes, it is possible to think of the portion of the matrix to the left of the separator nodes as 2 parallel matrices as illustrated in Figure 3b).

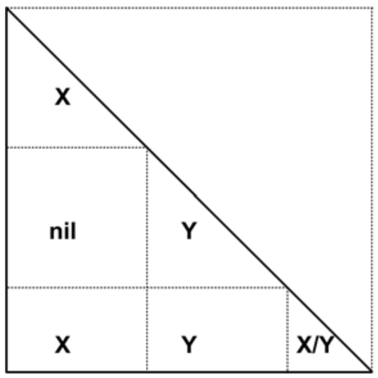

Fig. 3a).Normal Planar Representation of a Matrix from Spectral Dissection

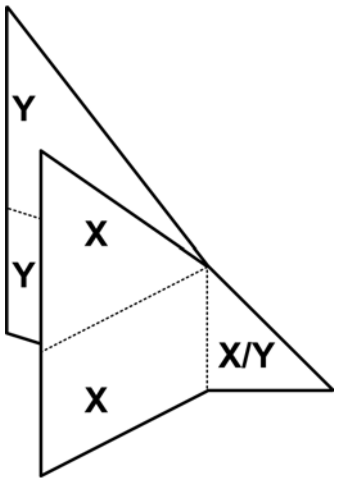

Fig. 3b). Parallel Representation of a Matrix from Spectral Dissection

Consequently, if there are “a” nodes in the separated node group X as shown in Figure 3b), then one processor core (X processor) can be used to perform a right-looking Cholesky decomposition on X for the “a” columns in X according to the decomposition operation, DECOMP(1:a) described in the Appendix. Simultaneously, if there are “b” nodes in the separated node group Y, then another processor core (Y processor) can be used in parallel to perform a right-looking Cholesky decomposition on the Y area illustrated in Figure 3b). Each of the processors can keep track of the net change on locations in X/Y due to subtractions that occur in the right-looking DECOMP(1:a) and DECOMP(1:b) operations. Subsequently only one processor proceeds to do the decomposition operation on the X/Y area which contains "c" columns. If the X processor will do the decomposition for the X/Y area, then the Y processor sends a list of adjustments for the X/Y area to the X processor. The X processor makes the listed adjustments to the X/Y area. If we number the first column of X/Y as "d", then the X processor does the decomposition operation DECOMP(d:d+c-1) on the X/Y area.

To obtain a solution for [G][v] = [In], the decomposition operation must be followed by forward and backward operations. Fortunately, these can also be done in parallel for the X and Y areas.

For the X area, a forward operation FORWARD(1:a) is done on the X processor as described in the Appendix. Similarly, for the Y area, a forward operation FORWARD(1:b) is done on the Y processor. When the X processor completes the FORWARD(1:a) operation, the vector of injections being operated on by the X processor has been reduced to a maximum dimension of "c", associated with the number of columns in X/Y. The vector of injections being operated on by the Y processor has also been reduced to a maximum dimension of "c". The vector of maximum dimension "c" is passed from the Y processor to the X processor where one vector of dimension "c" is created by addition. The X processor then continues the forward operation on the X/Y area with the operation FORWARD(d:d+c-1).

For the above case, the backward operation is started by the X processor which does a backward operation BACKWARD(d+c-1:d) on the X/Y area. This produces the node voltages for the X/Y area. These node voltages are passed to the Y processor so that both processors can continue independently to produce node voltages for the X and Y areas. The X/Y area voltages provide input for any elements located below row "a" in the modified local injection vector on the X processor. The X/Y area voltages also provide input for any elements below row "b" in the modified local injection vector on the Y processor. The backward operations are completed by a BACKWARD(a:1) operation on the X processor and a BACKWARD(b:1) operation on the Y processor.

The above description of an example solution is focused on the division of the [G][v] = [In] solution between two processors. Fortunately, the technique can be recursively applied to divide the solution between a multitude of processors. Figure 4 illustrates the dissection of the X area into X and W areas and the Y area into Y and Z areas. The dissection illustrated in Figure 4 supports the use of 4 processors. Of course, this nested dissection could be extended to 8 processors and beyond. This recursive bifurcation is referred to in the literature [2] as "spectral nested dissection".

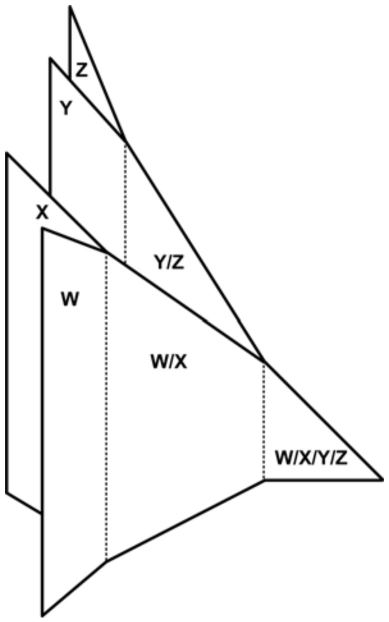

Fig. 4. Product of Spectral Nested Dissection

III. Handling Multiple Embedded Subnetworks

Often the overall network conductance matrix [G] contains models that can be considered as subnetworks. The HVDC valve group model mentioned in the Introduction section of this paper is an example of such a subnetwork model. The model has N internal nodes and M perimeter nodes respectively associated with the diagonals of the A and D areas of the matrix in Figure 1. Figure 5 illustrates two instances of the lower triangle of the model overlay matrix [Gm] being connected into the lower triangle of a reduced-dimension overall network conductance matrix [Gr] of dimension d. The reduction in dimension of matrix [Gr] exists because [Gr] does not contain the N internal nodes of either [Gm] matrix.

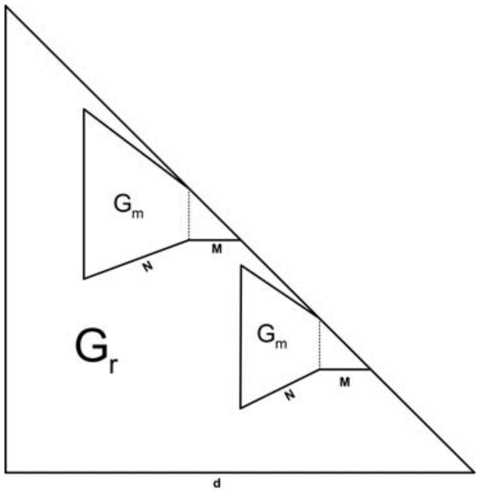

Fig. 5. Representation of Embedded Subnetwork Triangular Matrices

The content of the matrix [Gm] can change in every time-step as the valves in the model switch between the ON and OFF states. Consequently, it is necessary to do a DECOMP(1:N) operation on each valve group processor as a first step in solving for all the node voltages. When this DECOMP(1:N) operation is complete, each valve group processor sends the resulting MxM lower triangle G overlay to the network processor. After the valve group processor completes the DECOMP(1:N) operation, the processor conducts a FORWARD(1:N) operation on the injection vector for the valve group model which is of dimension P = N + M. When the FORWARD(1:N) operation is complete, the valve group processor sends the modified M lowest locations in the model injection vector to the network processor.

The network processor receives the MxM lower triangle conductance overlay and the injection vector of dimension M from each valve group processor. The network processor adds the MxM conductance overlays onto the base matrix [Gr]. It also adds the M dimension valve group injection vectors into the d dimension network injection vector. The completion of these additions allow the network processor to do a DECOMP(1:d) operation followed by a FORWARD(1:d) operation.

The network processor then begins the BACKWARD(d:1) operation to produce the network node voltages. The network processor sends the appropriate M valve group perimeter node voltages calculated as part of the BACKWARD(d:1) operation to each valve group processor. The BACKWARD(d:1) operation does not produce the N internal nodes voltages for each valve group.

Each valve group processor places the M perimeter node voltages into the M bottom locations of the modified injection vector that was previously produced on the valve group processor by the FORWARD(1:N) operation. The valve group model then solves for the N valve group internal node voltages by conducting a BACKWARD(N:1) operation. This completes the calculation of all the node voltages.

It should be noted that Figure 5 illustrates the ordering of the M perimeter nodes in the matrix [Gr] as being the same as in the [Gm] matrix. However, this is not actually required. In fact, the M perimeter nodes in [Gr] do not even need to be contiguously located. It is also possible to connect embedded subnetwork triangular matrices [Gm] into the multi-processor matrix solutions shown in Figures 3b) and 4.

IV. Demonstration of the Embedded Subnetwork Technique

Section III above describes the technique of embedding a subnetwork matrix [Gm] into an overall reduced dimension network matrix [Gr]. This section describes the practical use of the technique in implementing a 12 pulse HVDC valve group model that contains 10 interior nodes, 5 perimeter nodes and 17 switching elements.

Figure 6 illustrates the icon for the 12 pulse HVDC valve group prepared for use in an RTDS ® real-time simulator. The 10 internal nodes are labelled as R, P, AV1, BV1, CV1, M, AV2, BV2, CV2 and N. The 5 perimeter nodes are labelled as A, B, C, CT and AN. Recently the model has been expanded to optionally support 4 windings on the 3 single-phase transformer to enable filters and reactive power support to be connected to the converter transformer. In that case, the model has 10 internal nodes and 8 external nodes and the resulting [Gm] matrix is of dimension 18.

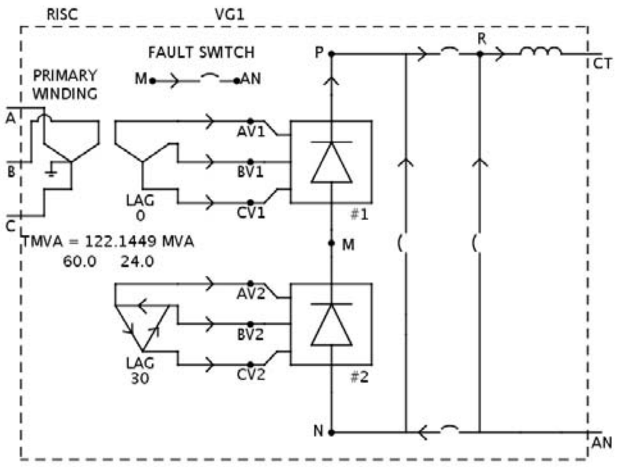

Fig. 6. 12-Pulse HVDC Valve Group Icon

Figure 7a) and 7b) illustrates typical plots of DC current and rectifier valve 1 voltage for a simulation containing the 12 pulse valve group model feeding a rated DC resistance and with firing delay of 0.5 radians. Figure 7a) illustrates curves for the case where there is no mutual inductance between the two 6 pulse valves groups, X12, X13 and X23 being respectively 0.1786, 0.1786 and 0.3572 per unit. Figure 7b) illustrates curves for the case where there is mutual inductance between the two 6 pulse valve groups, X12, X13 and X23 being respectively 0.1786, 0.1786 and 0.1786 per unit. The valve voltage plot in Figure 7b) contains the 2 expected additional commutation notches per cycle which match those illustrated in the textbooks [6]. An particular, there is an additional upward directed notch early in the voltage wave and a downward directed notch late in the wave.

The new model has a number of new features in addition to the ability to model 3 phase 4 winding transformers. One such feature is a user-specifiable forward voltage drop for the individual valves which is useful in properly modeling blocking and bypassing of 12 pulse groups in ultra high voltage (UHV) HVDC simulations.

V. Conclusions

This paper has briefly reviewed the efforts of leading researchers [1][2][3] in the area of “Spectral Nested

Dissection" (SND) techniques in effectively ordering large dimension symmetric positive definite matrices to facilitate parallel decomposition. The applicability of these techniques to EMT simulation has also been briefly explained.

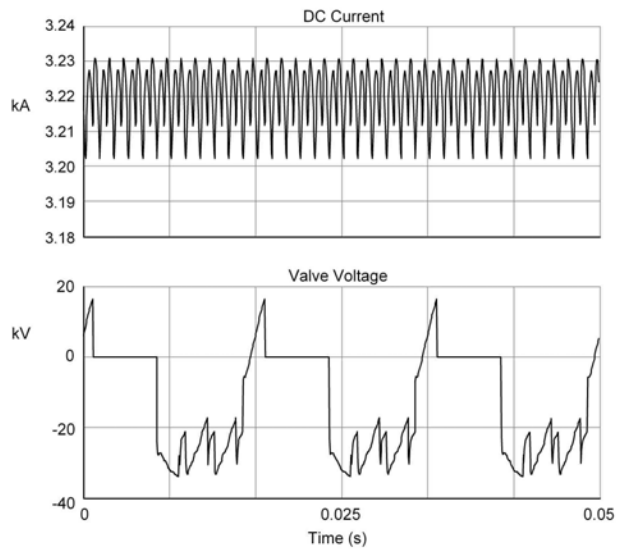

Fig. 7a). Valve Voltage - With No Transformer Mutual Inductance

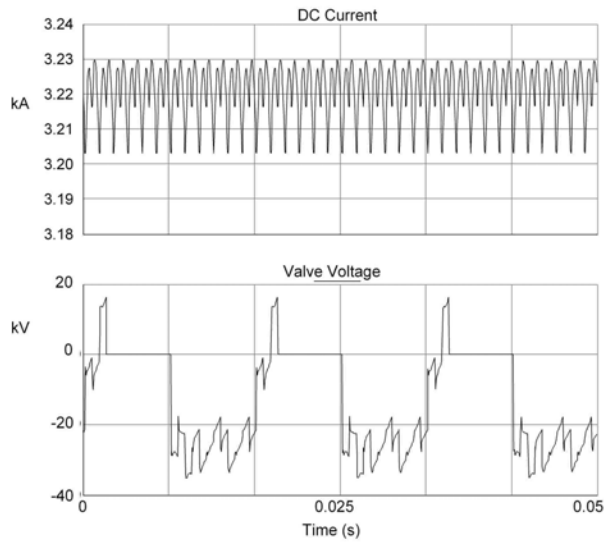

Fig. 7b). Valve Voltage - With Transformer Mutual Inductance

In addition to SND techniques, other techniques such as "Multiple-Minimum-Degree" (MMD) [4] and "Sparspak Automatic Nested Dissection" (AND) [5] may provide methods that can be employed in the EMT simulation area. Evaluating these alternative techniques is an area of future research.

This paper also explains a practical approach for removing the calculations for subnetwork conductance matrices from the main network solution processor and for performing those calculations in parallel on separate embedded model processors. An implementation of those techniques is provided in the 12 pulse HVDC valve group model.

The techniques explained in this paper have already enabled the real-time simulation of networks that require larger connected conductance matrices. It is expected that more research in this area can provide even greater benefits.

VI. Appendix

The following C language code segments are a typical right-looking Cholesky decomposition and associated forward and backward operations using columns F to L inclusive of the symmetric positive definite dxd conductance matrix, [G]. In right-looking Cholesky factorization, the matrix is traversed by column from the left to the right with all columns to the right of the current column being updated immediately.

The decomposition operation and the result therefrom include a typical modification to avoid division operations during the subsequent forward and backward operations. The vector [In] is converted from a nodal injection vector to a vector of node voltages by the combined effect of all of the forward and backward operations.

For purposes of notation, we define an operation DECOMP(F:L) illustrated in the following C code as the in-place Cholesky decomposition operations using multiplication factors from columns F to L inclusive of the lower triangle of a dxd symmetric positive definite matrix [G]. If there are any columns to the left of column F, then operations for those columns must be completed in advance of the DECOMP(F:L) operation.

```
for(j=F; j<=L; j++)
{
  /* modify the diagonal of column j */

  H         = 1.0/sqrt( G[j][j] );
  G[j][j]   = H;

  /* modify the column  */
  /* below the diagonal */

  for(i=j+1; i<=d; i++)
  {
    G[i][j] = H * G[i][j];
  }

  /* modify locations to      */
  /* the right of the column  */

  for(i=j+1; i<=d; i++)
  {
    H = G[i][j];

    for(k=i; k<=d; k++)
    {
      G[k][i] -= H * G[k][j];
    }
   }
 }
```

For purposes of notation, we define an operation

FORWARD(F:L) illustrated in the following C code as the forward operations on the [In] vector using columns F to L inclusive of the dxd [G] matrix result that was produced by the DECOMP(F:L) operation. If there are any columns to the left of column F, then operations for those columns must be completed in advance of the FORWARD(F:L) operation.

```
for(j=F; j<=L; j++)
 {
   /* forward operation for column j */

   H         = In[j] * G[j][j];
   In[j]     = H;

   for(i=j+1; i<=d; i++)
   {
     In[i] -= H * G[i][j];
   }
 }
```

For purposes of notation, we define an operation BACKWARD(L:F) illustrated in the following C code as the backward operations on the modified [In] vector using columns L to F inclusive of the dxd [G] matrix result that was produced by the DECOMP(F:L) operation. If there are any columns to the right of column L, then operations for those columns must be completed in advance of the BACKWARD(L:F) operation.

```
 for(j=L; j>=F; j--)
 {
   /* backward operation for column j */

   H     = In[j];

   for(i=d; i>j; i--)
   {
     H  -= G[i][j] * In[i];
   }

   In[j] = H * G[j][j];
 }
```

VII. References

[1] A. Pothen, H.D. Simon, and K.-P. Liou, "Partitioning sparse matrices with eigenvectors of graphs", SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430-452.
Available (spaces in URL are underscores):
http://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19970011963_1997016998.pdf

[2] A. Pothen, H.D. Simon, and L. Wang, "Spectral nested dissection", Tech. Rep. CS-92-01, Computer Science, Pennsylvania State University, University Park, PA, 1992.
Available:
http://www.nersc.gov/homes/simon/Papers/NASA/rnr-92-003.pdf

[3] Alex Pothen, Edward Rothberg, Horst Simon, and Lie Wang, "Parallel Sparse Cholesky Factorization with Spectral Nested Dissection Ordering", Report RNR-94-011, May 1994, NAS Systems Division, Applied Research Branch, NASA Ames Research Center
Available (spaces in URL are underscores):
http://reference.kfupm.edu.sa/content/p/a/parallel_sparse_cholesky_factorization_w_1540750.pdf

[4] J. W. H. Liu, "Modification of the minimum degree algorithm by multiple elimination", ACM Trans. on Math. Software, 11 (1985), pp. 141-153.

[5] E. C. H. Chu, A. George, J. W. H. Liu, and E. G. Y. Ng, "User's guide for Sparspak-A: Waterloo sparse linear equations package", Tech. Rep. CS-84-36, Computer Science, University of Waterloo, Ontario, Canada, 1989.

[6] Colin Adamson and N.G. Hingorani, *High Voltage Direct Current Power Transmission*, London: Garraway, 1960, p. 44.

VIII. Biographies

Trevor Maguire graduated from the University of Manitoba with B.Sc.EE, LL.B., M.Sc.EE and Ph.D. degrees in 1975, 1979, 1986, and 1992 respectively. Relevant employment experience includes time with Manitoba Hydro (1975-76), Manitoba HVDC Research Centre (1986-1994), and RTDS Technologies, Inc. (1994-present). He is a founding principal of RTDS Technologies, Inc. with a special interest in real time simulation model development and also real-time simulation digital hardware development. He participated in creating the world's first commercial real-time digital power system simulator.