Tree Structured Symmetrical Systems of Linear Equations and their Graphical Solution

Jaime Cerda and Mario Graff

Abstract

Tree-structured symmetrical systems of linear equations are symmetrical systems of linear equations whose underlaying matrix topology can be represented by a tree. The solution to a large kind of problems can be addressed by solving a tree-structured symmetrical system of linear equations, in particular, electrical distribution networks. In general these systems are solved using matrix approaches which when taken to the sparse solution do not take advantage of the tree structure of the system. In this document we explore the graphical representation of this systems as well as proposing the graphical solution to such kind of systems. Furthermore, the method presented, for this kind of systems, does not generate elements at all in the elimination process, something which leads to remarkably efficient algorithms in time and memory usage.

***Index Terms*—Systems of Linear Equations, Matrices, Graphs, Gaussian Elimination, Trees, Sparsity.**

I. Introduction

TREE-structured symmetrical systems of linear equations are symmetrical systems of linear equations whose underlaying matrix topology can be represented by a tree. The solution to a large kind of problems can be addressed by solving a tree-structured symmetrical system of linear equations, in particular, electrical distribution networks. In general these systems are solved using matrix approaches which when taken to the sparse solution do not take advantage of the tree structure of the system. In this document we explore the graphical representation of this systems as well as proposing the graphical solution to such kind of systems. Furthermore, the method presented, for this kind of system, does not generate elements at all i the elimination process, something which leads to remarkably efficient algorithms in time and memory usage. To this end the organization of this document is as follows: Section II presents a graphical representation for a Symmetrical System of Linear Equations (SSLE). Next section III presents a closer look to special SLEs whose structure are represented by a tree (TSSSLE). Section IV presents the graphical interpretation for the Gaussian elimination. Following, a graph-based solution for tree structured symmetric systems of linear equations is given in section V. Finally section VII provides some concluding remarks.

Manuscript received July 26, 2014; revised August 5, 2014. This work was supported by the Scientific Research Coordination, Universidad Michoacana de San Nicolas de Hidalgo. J. Cerda (jcerda@umich.mx) and M. Graff (mgraffg@gmail.com) are with the Electrical Engineering School, Universidad Michoacana de San Nicolas de Hidalgo, Morelia, Michoacan, Mexico

II. Symmetrical Systems of Linear Equations and Its Graphical Representation.

An n-order system of linear equations represent systems where the number of equations is equal to the number of variables have the form of equation 1.

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{1}$$

where $\boldsymbol{A} \in \mathbb{R}^{n\times n}$, $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{b} \in \mathbb{R}^n$. The non-zero elements in $\boldsymbol{A}$, excluding those in the diagonal,

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

define the topology of the graph which represents the SLE. If $\boldsymbol{A}$ is very sparse then the graph will have very few interconnections. It turns out that many physical systems solved using linear systems are very sparse. In particular, in electrical power systems, the matrices which represent transmission networks are very sparse. This is the main reason why sparsity techniques have been improved by research whose goal is to solve efficiently the actual state of the power system; such as the best elimination ordering [6], [9]. Sparsity techniques have been around since at least 1970 using a technique known as bifactorization [10]. The main principles used in matrix bifactorization are strongly directed toward exploiting the underlying matrix graph.

Therefore in order to approach the solution using its graph representation, first an appropriate model has to be derived. This model has to be able to represent the complete SSLE elements (i.e. $\boldsymbol{A}, \boldsymbol{x}$, and $\boldsymbol{b}$). The model proposed in this document is based on a per equation basis. This representation is given in figure 1.

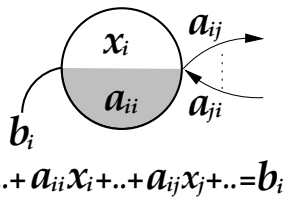


Figura 1. Conversion from a linear system of equations to its graph model

In this model an equation is represented by two components: a node and a set of links. The node is a well defined component which consists of two subcomponents: a circle consisting of two half parts and an arc. The upper part of the circle represents the variable related to this equation which has to be solved by the system (i.e. x_i) and the lower part represents the coefficient related to this variable in equation i (i.e. a_{ii}). The arc represents the $i-th$ component in $\boldsymbol{b}$ (i.e. b_i). The second part depends on the SLE topology and is represented by links which connect the nodes. These links will be denoted as (i, j) where i and j represent the row and the column number respectively. There can be zero or more links which connect the node with some other nodes in the graph. Each link has an associated value for the coefficient located in the row i column j (i.e. a_{ij}). Perhaps it is a little absurd to consider the case where there are no external links. However, as will be shown later, this is the basic configuration which will always be pursued in order to solve the SLE.

III. Tree Structured Symmetric Systems of Linear Equations and Its Graphical Representation.

Tree structured symmetrical systems of linear equations (TSSSLE) are SLE where the graph representing the SLE is a tree. Based on this structure, efficient algorithms can be derived in order to solve this kind of systems. These algorithms emerge naturally, just by exploiting the properties of trees. This kind of graphs have been applied to solve electrical distribution networks whose main characteristic is its radial shape (i.e. no loops exists in the network). Therefore a tree structure can be derived for the SLE representing these systems. Algorithms to solve different problems with different degree of complexity have been proposed for distribution networks based on this structure in [5], [4], [2], [7].

A tree-shaped graph has to be free of loops. This work does not deal with how to identify and remove loops from graphs. Let us instantiate equation 1 with equation 2.

$$\begin{pmatrix} 0{,}5 & 0 & -1 & 0 \\ 0 & 2 & -1 & 0 \\ -1 & -1 & 1 & -1 \\ 0 & 0 & -1 & 4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -5 \\ 0 \end{pmatrix} \quad (2)$$

whose solution for the TSSSLE denoted by equation 2 is given by 3

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 40/7 \\ 10/7 \\ 20/7 \\ 5/7 \end{pmatrix} \quad (3)$$

Applying the model defined in figure 1, leads to the graph shown in figure 2 which which complains with the TSSSLE properties.

This example will be used throughout this chapter. The system will be solved using different strategies which have to lead to the same solution. To this end

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

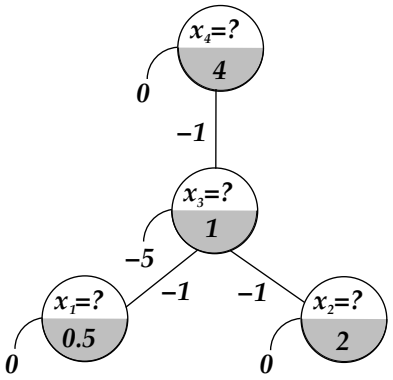


Figura 2. Graph corresponding to the system defined by equation 2

Gaussian elimination will be used as the main tool to solve the system.

IV. Gaussian Elimination and Its graphical Interpretation

Gaussian elimination is a general method to solve a SLE. It consists of the iterative application of elementary row operations which lead the system to an echelon form. This is achieved by modifying each of the elements which do not belong to the column and row to the equation under reduction. These elements are modified using expression 4.

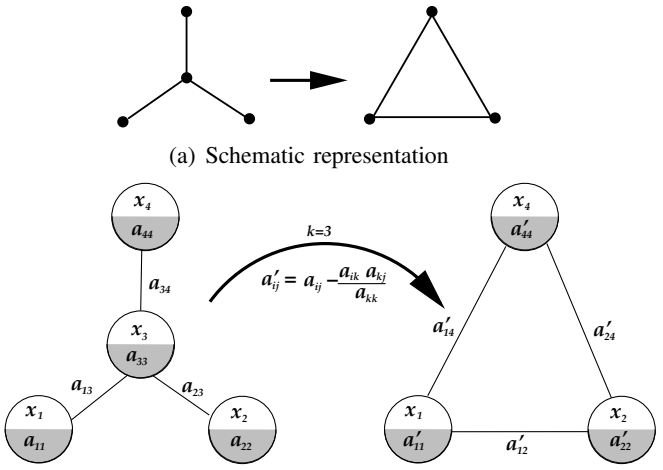
$$a_{ij}^{'} = a_{ij} - \frac{a_{ik}a_{kj}}{a_{kk}} \tag{4}$$

Gaussian elimination can be regarded as a matrix transformation from $\mathbb{R}^{n\times n} \rightarrow \mathbb{R}^{(n-1)\times(n-1)}$. The resulting system has all the information needed to solve the subsystem resulting from the transformation. Let us define Γ_k as the set of nodes adjacent to node k. Then, the maximum number of elements generated in the elimination is given by equation 5 .

$$N_k = \frac{|\Gamma_k|\left(|\Gamma_k| - 1\right)}{2} \tag{5}$$

IV-A. Special Cases for Gaussian Elimination

As mentioned previously, Gaussian elimination has two main costs: updating and link creation. The first one is unavoidable but the second one can be optimised if an elimination order is achieved such that the number of links created are minimal, or in the ideal case no links are created. There are three special cases which deserve special attention. The first one is shown in figure 3. This is the well known as star-delta transformation in electrical engineering. Here $|\Gamma_k| = 3$ and by equation 5 at most three new links will be generated.

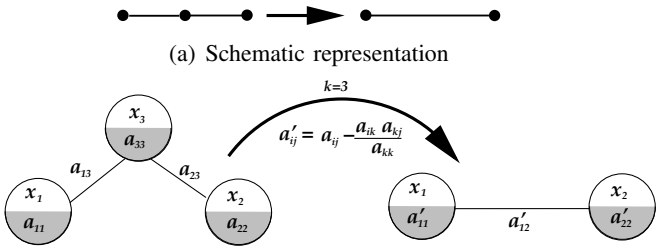


(a) Schematic representation

(b) Complete transformation

Figura 3. Gaussian elimination for $|\Gamma_k| = 3$.

The second one is shown in figure 4. This is the well known series reduction in electrical engineering. Here $|\Gamma_k| = 2$ and by equation 5 at most one new link will be generated.



(a) Schematic representation

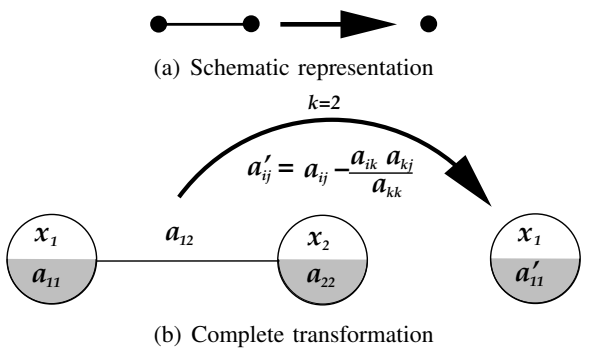
(b) Complete transformation

Figura 4. Gaussian elimination for $|\Gamma_k| = 2$.

The third and most important case regarding this document , as it will be shown along this section, is shown in figure 5. This configuration can be regarded as a dangling node which can be reduced into the node where it is dangling from. Here $|\Gamma_k| = 1$ and by equation 5 there will be no new links generated!! This basic fact will be used in order to reduce the

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TSSSLE as the leafs can be regarded as dangling nodes. Once they are reduced, then the nodes where they were dangling from can be reduced, and so on.



(a) Schematic representation

(b) Complete transformation

Figura 5. Gaussian elimination for $|\Gamma_k| = 1$.

V. Graph-based Solution for Tree Structured Symmetric Systems of Linear Equations

A tree is a special kind of graph whose main characteristic is the absence of loops. A standard representation for a tree is presented in figure 6. A tree has n layers, $n \geq 1$, numbered from 0 to $n-1$. The number of layers represents its depth. A function $layer(x)$ can be defined to express the layer where node x is located, for instance $layer(E) = 2$. Another useful function is $parent(x)$ which returns the node where node x is hanging from or nil if it has no parent, for instance $parent(E) = B$ and $parent(A) = nil$.

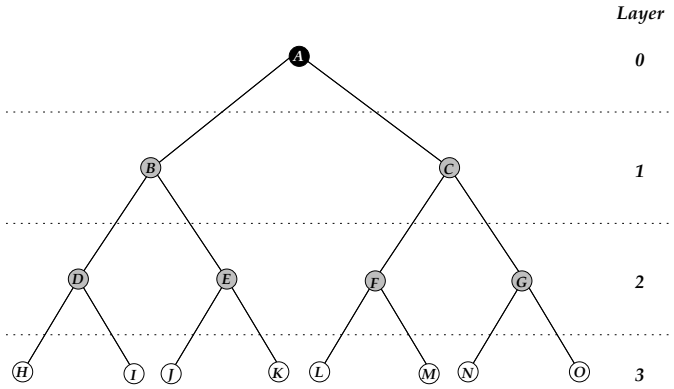


Figura 6. An example of a tree.

Let us denote Υ_i as the set of indices j which corresponds to variables x_j which are hanging from x_i as denoted by equation 6 (i.e. the *children* of x_i).

$$\Upsilon_i = \{j : \exists(i,j), parent(i) \neq j\} \tag{6}$$

For instance $\Upsilon_A = \{B, C\}$, and $\Upsilon_G = \{N, O\}$ in figure 6. Based on these definitions, the nodes in a tree can be classified in three types

- *root* node: There is just one of them in the graph such that $layer(root) = 0$ and $parent(root) = nil$. For instance the root node in figure 6 is A (in black),
- *internal* node: are those nodes x such that $\Upsilon_x \neq \emptyset$ and $x \neq root$. The set of internal nodes, I, in figure 6 is $\{B, C, D, E, F, G\}$ (in gray),
- *terminal* node: also called *leaf* node, are those nodes x where $\Upsilon_x = \emptyset$. Therefore the set L representing the leaf nodes in figure 6 is $\{H, I, J, K, L, M, N, N, O\}$ (in white).

This kind of graphs are one of the most useful structures in computer science. Their application spans from context-free grammar analysers in compiler theory, XML representation in internet technologies, and so on. A common characteristic about those applications is that only the leaf nodes have real information. However, when applied to TSSSLEs, the internal nodes will have some information which need to be processed in order to solve the SLE.

It is very important to preserve the tree structure of the system in the reduction process. The best way to reduce these graphs is by finding an order which does not generate new elements at all. From figures 3, 4 and 5 a basic fact can be asserted. The only reductions which do not generate new elements are those which are applied to nodes k where $|\Gamma_k| = 1$. Therefore those are the ideal candidates to apply the elimination process.

Gaussian elimination when applied to all nodes j where $j \in \Upsilon_i$, is expressed in equations 7 and 8 for the coefficient corresponding to variable x_i and the independent term respectively.

$$a'_{ii} = a_{ii} - \sum_{j \in \Upsilon_i} \frac{a_{ij}^2}{a_{jj}} \tag{7}$$

$$b'_i = b_i - \sum_{j \in \Upsilon_i} \frac{b_j a_{ij}}{a_{jj}} \tag{8}$$

The solution for the SLE once the reduction process has been applied is straightforward. To this

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

end, advantage is taken from the modifications the reduction process in its way up has made to the tree. It has updated the value of b_i and a_{ii} when their dependencies with the lower layer were eliminated. Let us suppose $parent(i) = k$. Then, the equation to be solved for x_i is

$$a_{ii}x_i + a_{ik}x_k = b_i \tag{9}$$

which is solved as

$$x_i = \frac{b_i - a_{ik}x_k}{a_{ii}} \tag{10}$$

clearly for this equation if node i is the tree root, then there will be no more layers up so this equation becomes

$$x_i = \frac{b_i}{a_{ii}} \tag{11}$$

Therefore, a top down propagation approach can be applied in order to solve the nodes in the lower layers. This can be based on a Breadth-first (BF) or a Depth-first (DF) algorithm [3]. Nevertheless, if the shape of the graph is known apriori, then more efficient algorithms can be proposed in order to speed up the computations avoiding recursivity or the handling of other auxiliary data structures.

VI. Applying the Graph-based Solution for TSSSLE to our case

If we take node 4 as the root node the the elimination and solution process is described in figure 7

The second elimination order shown in 7 is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$. As previously stated, no new links are created. However, the initial configuration for nodes 3 and 4 has been modified.

VI-A. About the Importance of the Initial Configuration

In the previous example, the modification to the initial configuration was mentioned. This is important to preserve or at least try to preserve it as much as possible as there are iterative algorithms which will be using this configuration in order to reach the solution. If the algorithm which solves the graph modifies this configuration at every iteration, then this will have to be reinstantiated in each of them.

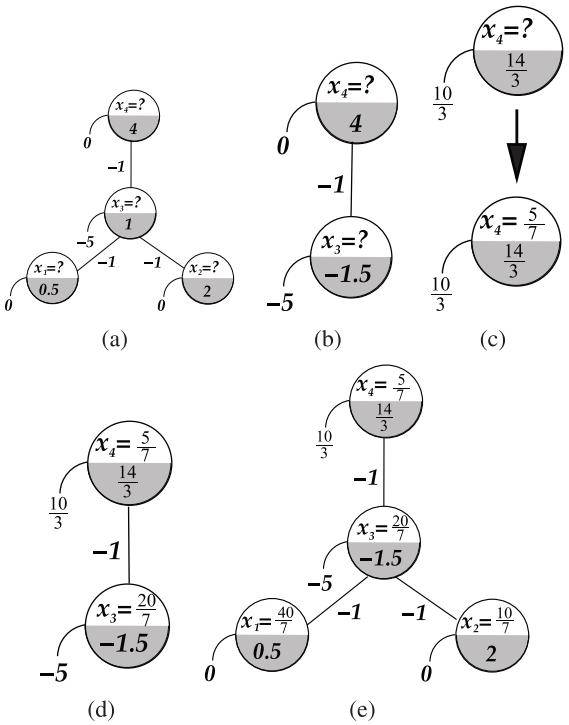


Figura 7. The graph solution taken node 4 as the root node

VII. Concluding Remarks

In this work the graph representation for TSSSLE have been presented. Then Gaussian elimination and its graphical interpretation has been presented.

When solving a TSSSLE, the objective is to find the values for variables represented by vector $\boldsymbol{x}$. The basic tool to find those values is based on the Gaussian elimination. The processing task for this graph will address the previous features so no links are created at all. A TSSSLE is a SLE which can be represented with a tree. Finally, the solution to TSSSLE does not require the creation of links, therefore is a very efficient structure which every solution algorithm must try to derive.

References

[1] Anne Berry and Pinar Heggernes. The minimum degree heuristic and the minimal triangulation process. In *Proceedings of WG 2003*, pages 58–70. Springer Verlag, 2003.

[2] G.J. Chen, K.K. Li, T.S. Chung, and G.Q. Tang. An efficient two-stage load flow method for meshed distribution networks. In *Proceedings of the 5th Conference on Advan-*

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

ces in Power System, Control, Operation and Management, APSCOM 2000, pages 537–542, October 2000.

[3] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Cliff Stein. *Introduction to Algorithms*. MIT Press, 2001.

[4] D. Das, H. S. Nagi, and D. P. Kothari. Novel method for solving radial distribution networks. *IEE Proceedings on Generation, Transmission and Distribution*, 141(4):291–298, July 1994.

[5] S. K. Goswami and S.K. Basu. Direct solution of distribution systems. *IEE Proceedings on Generation, Transmission and Distribution*, 138:78–85, 1991.

[6] Harry M. Markowitz. The elimination form of the inverse and its application to linear programming. *Management Science*, 3(3):255–269, April 1957.

[7] S.F. Mekhamer, S.A. Soliman, M.A. Moustafa, and M.E. El-Hawary. Load flow solution of radial distribution feeders: A new contribution. *Electrical power and Energy Systems*, 24:70–707, 2002.

[8] Gilbert Strang. *Linear Algebra and Its Applications*. Brooks Cole, 4 edition, July 2005.

[9] W. F. Tinney and J. W. Walker. Direct solutions of sparse network equations by optimally ordered triangular factorization. *Proceedings of the IEEE*, 55(11):1801–1809, 1967.

[10] K. Zollenkopf. Bifactorization: Basic computational algorithm and programming techniques. In Oxford, editor, *Conference on Large Sets of Sparse Linear Equations*, pages 76–96, 1970.

ISBN: 978-988-19252-0-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)