Application of graph theory to the nonlinear analysis of large space structures

Magdy Ibrahim Hindawy
Iowa State University

Follow this and additional works at: https://lib.dr.iastate.edu/rtd
Part of the Aerospace Engineering Commons

Recommended Citation
Hindawy, Magdy Ibrahim, "Application of graph theory to the nonlinear analysis of large space structures " (1986). Retrospective Theses and Dissertations. 8007.
https://lib.dr.iastate.edu/rtd/8007

This Dissertation is brought to you for free and open access by the Iowa State University Capstones, Theses and Dissertations at Iowa State University Digital Repository. It has been accepted for inclusion in Retrospective Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.
INFORMATION TO USERS

This reproduction was made from a copy of a manuscript sent to us for publication and microfilming. While the most advanced technology has been used to photograph and reproduce this manuscript, the quality of the reproduction is heavily dependent upon the quality of the material submitted. Pages in any manuscript may have indistinct print. In all cases the best available copy has been filmed.

The following explanation of techniques is provided to help clarify notations which may appear on this reproduction.

1. Manuscripts may not always be complete. When it is not possible to obtain missing pages, a note appears to indicate this.

2. When copyrighted materials are removed from the manuscript, a note appears to indicate this.

3. Oversize materials (maps, drawings, and charts) are photographed by sectioning the original, beginning at the upper left hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is also filmed as one exposure and is available, for an additional charge, as a standard 35mm slide or in black and white paper format.*

4. Most photographs reproduce acceptably on positive microfilm or microfiche but lack clarity on xerographic copies made from the microfilm. For an additional charge, all photographs are available in black and white standard 35mm slide format.*

*For more information about black and white slides or enlarged paper reproductions, please contact the Dissertations Customer Services Department.
APPLICATION OF GRAPH THEORY TO THE NONLINEAR ANALYSIS OF LARGE SPACE STRUCTURES

Iowa State University  PH.D.  1986

University Microfilms International  300 N. Zeeb Road, Ann Arbor, MI 48106
Application of graph theory to the nonlinear analysis of large space structures

by

Magdy Ibrahim Hindawy

A Dissertation Submitted to the Graduate Faculty in Partial Fulfillment of the Requirements for the Degree of DOCTOR OF PHILOSOPHY

Major: Aerospace Engineering

Approved:

Signature was redacted for privacy.

In Charge of Major Work

Signature was redacted for privacy.

For the Major Department

Signature was redacted for privacy.

For the Graduate College

Iowa State University
Ames, Iowa
1986
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td></td>
</tr>
<tr>
<td>1.1. Large Space Structures (LSS)</td>
<td>1</td>
</tr>
<tr>
<td>1.2. Application of Graph Theory in LSS Analysis</td>
<td>3</td>
</tr>
<tr>
<td>1.3. DOC (Dynamic Ordering and Condensation)</td>
<td>3</td>
</tr>
<tr>
<td>2. NONLINEAR ANALYSIS OF LARGE SPACE STRUCTURES</td>
<td>5</td>
</tr>
<tr>
<td>2.1. Basic Structural Definitions for LSS</td>
<td>5</td>
</tr>
<tr>
<td>2.2. Formulation of the Nonlinear Model for LSS</td>
<td>9</td>
</tr>
<tr>
<td>2.2.1. Nonlinear finite element methods</td>
<td>9</td>
</tr>
<tr>
<td>2.2.2. Large displacement formulations</td>
<td>10</td>
</tr>
<tr>
<td>2.2.3. LSS global static equilibrium equation</td>
<td>12</td>
</tr>
<tr>
<td>2.3. LSS Nonlinear Analysis Procedure</td>
<td>12</td>
</tr>
<tr>
<td>2.3.1. The margin of linearity</td>
<td>13</td>
</tr>
<tr>
<td>2.3.2. Nonlinear checking priority</td>
<td>14</td>
</tr>
<tr>
<td>2.3.3. Analysis procedure</td>
<td>15</td>
</tr>
<tr>
<td>3. GRAPH REPRESENTATION OF LSS WITH NONLINEAR REGIONS</td>
<td>17</td>
</tr>
<tr>
<td>3.1. Basic Definitions and Theorems of Graph</td>
<td>17</td>
</tr>
<tr>
<td>3.2. Graph Representation of LSS</td>
<td>20</td>
</tr>
<tr>
<td>3.2.1. Nodes, members, loads</td>
<td>20</td>
</tr>
<tr>
<td>3.2.2. Models, structures, substructures</td>
<td>22</td>
</tr>
</tbody>
</table>
3.3. Rearranging Algorithms

3.3.1. SR (Structure Rearranging) algorithm

3.3.2. MR (Model Rearranging) algorithm

3.4. Linear and Nonlinear Models in LSS

4. ORDERING OF THE STRUCTURE GRAPH, G(α)

4.1. The Multi-Level Structure of LSS, MLS(α)

4.2. The Linear Interval

4.3. The Transition Interval

4.3.1. The super graph, \( G^*(α) \)

4.3.2. The block graph, \( \mathcal{G}(α) \)

4.3.3. The brick graph, \( \mathcal{G}(I) \)

4.3.3.1. Ordering of a super node, \( \mathcal{G}_K \)

4.3.3.2. Ordering of a substructure \( \mathcal{G}_S \)

4.4. Storage Allocation of Block Stiffness Equation

4.4.1. Block stiffness matrix

4.4.2. Load block vector

4.4.3. The priority set, \( Y_0 \)

4.5 The Nonlinear Interval

5. UPDATING AND REORDERING OF THE STRUCTURE GRAPH, G(α)

5.1. ID (Identification) Process

5.1.1. Nonlinear member ID

5.1.2. Adjacent nodes ID
5.1.3. Substructure level ID 61

5.2. The Basic Configurations 61
5.2.1. Initiation of a new nonlinear substructure 63
5.2.2. Fill in through one transition node 64
5.2.3. Fill in through two transition nodes 65
5.2.4. Connection of two nonlinear substructures 65

5.3. Updating and Reordering of the Brick Graph, $G^a(I)$ 66
5.3.1. Updating a super node 66
5.3.2. Updating a nonlinear substructure 67
5.3.3. Updating a linear substructure 67

5.4. Updating and Reordering of the Super Graph, $G^*(\alpha)$ 68
5.4.1. The linear interval 68
5.4.2. The transition begins 68
5.4.3. The transition interval 69
5.4.3.1. Steady state configuration 70
5.4.3.2. Initiation of a nonlinear substructure 71
5.4.3.3. Connection of two nonlinear substructures 71
5.4.3.4. Elimination of a linear substructure 72
5.4.3.5. Split of a linear substructure 72

5.4.4. The transition ends 73

5.4.5. The nonlinear interval 73

5.5. Updating and Reordering of the Block Graph, \( \mathcal{G}(\alpha) \) 73

5.5.1. Updating the structure level matrix, \([C^\alpha]\) 74

5.5.2. Updating the set \( N_\gamma \) 75

5.5.3. Updating of the block matrix \([B^\alpha]\) 75

5.6. Updating the Priority Set, \( Y^\alpha \) 75

6. CONDENSATION OF LINEAR SUBSTRUCTURES 76

6.1. Condensation of a Brick Graph, \( \mathcal{G}^\alpha(I) \) 76

6.2. Condensation of the Block Graph, \( \mathcal{G}(\alpha) \) 78

6.2.1. Condensation of a linear substructure, \( \alpha S_L \) 78

6.2.2. The condensed nonlinear graph, \( \mathcal{G}^C(\alpha) \) 80

6.2.3. The condensed nonlinear stiffness matrix, \([K^C]\) 84

6.3. Reverse Condensation 86

6.4. Updating the Condensed Graph, \( \mathcal{G}^C(\alpha) \) 86

7. CONCLUSION AND FUTURE PROJECTIONS 89

8. BIBLIOGRAPHY 91

9. ACKNOWLEDGMENTS 95
1. INTRODUCTION

1.1. Large Space Structures (LSS)

A number of future space missions being considered by NASA require ultra-low-mass, large space structures (LSS). Missions such as solar power collection, communications, and earth resource surveillance will be considered. Although the purposes of these missions vary widely, there appears to be a high degree of structural commonality. Large space structures are characterized by their large dimensions and small rigidity. Some projects require structures that are larger in area than the largest existing earth-bound structures. The prospect of transporting into orbit structures of this magnitude provides a unique challenge to the aerospace design community to develop extremely efficient structural concepts. The feasibility of these missions will depend on compatibility of the structural concept with available transportation systems. Therefore, it is necessary to identify and develop new efficient ways to design, fabricate, assemble, control and maintain these large space structures in their orbit.

Within the decade, large space structures will be assembled in low earth orbit (LEO). In some instances, these structures will consist of large areas of assembled members, such as antennas and solar arrays mounted on structural platforms. These lightweight and very flexible structures
will be transferred from (LEO) to much higher orbits, such as geosynchronous equatorial orbit (GEO), using specially designed propulsive systems. The propulsive thrust forces will have great influence on the overall structural configuration and structural design methods. One alternative to keep the weight of LSS in the feasible region and below current estimates is to allow regions of the structure to respond nonlinearly during these maneuvers. The existence of these local nonlinear regions in LSS increases the difficulty of its structural analysis and design. Such an analysis has been a very complicated and expensive process, and it has forced many structural designers to use simpler methods which are inadequate.

These methods can be divided into three main categories.

1—Reduced order methods.

2—Fixed region analysis (linear-nonlinear interaction regions with fixed boundaries)

3—Global-local (Ritz-like) approximate methods.

The validity of the results of these methods for a large space structure is questionable. Since the alternative is to perform a full scale nonlinear analysis for all the DOF of the LSS, these simpler methods have been used.

In this research, a new method is developed to reduce the order of the nonlinear problem to include only the nonlinear DOF and condense out all the linear DOF. It also updates the
order reduction during the loading process. This method is based on graph theory concepts.

1.2. Application of Graph Theory in LSS Analysis

Graph-theoretic approaches have found many applications in sparse matrix studies [1-3]. Although few results from graph theory are directly applicable to the study of sparse matrices, the graph representation is nonetheless a powerful tool to characterize the structure of a sparse matrix. Application of graph theory, in data structure techniques to develop better algorithms is well-established [4-7]. Recently, it has proved its unique efficiency in the linear and nonlinear analysis of large truss structures using the network theories [8-12].

In this research, the graph theory will be used to represent LSS and establish a new algorithm, to order and reduce the discretized model of the LSS with nonlinear regions of variable configuration, and update this process during the loading interval.

1.3. DOC (Dynamic Ordering and Condensation) Algorithm

DOC is a multi-level structured algorithm to order the nodes and the members of LSS at a certain load amplitude. A multi-level graph can be introduced to represent the LSS. This graph restructures its levels as the nonlinear regions change their configurations during the change of the load.
amplitude. The DOC algorithm also condenses out all the linear members and updates the condensed stiffness matrix at every load step. The final result is to reduce the size of the nonlinear problem without assuming an artificial reduced space for the solution vector of the global stiffness equation. This algorithm will be introduced in Chapters 3-6.
2. NONLINEAR ANALYSIS OF LARGE SPACE STRUCTURES

The nonlinear structural analysis process for LSS can be divided into five separate tasks:

1. Formulation of the discretized model.
2. Storage allocation.
3. Ordering such that the system of equations will have a desirable structure.
5. Solution to compute the solution vector at every load step.

This research focuses on the third task. This task involves the ordering of a system of equilibrium equations resulting from finite element discretization of the structure. It will be assumed that nonlinear regions exist within the LSS.

2.1. Basic Structural Definitions for LSS

A node is a point defined by its coordinates in the global axis system of the structure. Associated with each node is a set of variables corresponding to the node DOF (degrees of freedom defined at that node). This set of DOF, in turn, corresponds to a submatrix in the global stiffness matrix. During assembly and solution this submatrix can conveniently be treated as one unit for analysis purposes.

A Boundary node, \( n_i \), is a node that exists on the boundary of a member.
A Member, $m$, (Figure 2.1) is the smallest substructure that is considered in this study to be either a linear or a nonlinear substructure and is defined by:

a) two boundary nodes $n_i$, $n_j$, 

b) compatibility conditions at the boundary nodes, 

c) an elastic line, connecting the two nodes $n_i$, $n_j$, defined by the equation $f_m(r) = 0$, $f_m \in C^1$, where $f_m : [0,1] \rightarrow \mathbb{R}^3$, e.g., for a 3-D Cartesian global axis. The function $f_m$ is introduced in the parametric form, 

$$
\begin{align*}
  f_m^T &= (x = x(r), y = y(r), z = z(r)), \\
  n_i &= (x(0), y(0), z(0)), n_j = (x(1), y(1), z(1)). 
\end{align*}
$$

For members with linear configuration, the parametric representation will have the form 

$$
\begin{align*}
  x &= x(0) + [x(1) - x(0)] \times r \\
  y &= y(0) + [y(1) - y(0)] \times r \\
  z &= z(0) + [z(1) - z(0)] \times r 
\end{align*}
$$

d) Intermediate nodes, rigidly connecting the Finite Elements to each other, inside the member. Their coordinates are defined by the parameter $r$.

$$(r_{k-1} < r_k < r_{k+1}) \quad k = 1, \ldots, NK_i, \quad NK_i = NE_i - 1$$

$NE_i$, $NK_i$ are the number of Finite Elements, and intermediate nodes in the member $m_i$, respectively. In our analysis, we will consider that the intermediate nodes are internal nodes in the member.
Figure 2.1. Configuration of a large space structure member
e) Finite Elements, are defined by their assigned order number \( k \) inside the member \( m^i \), where \( k = 1, \ldots, NE_i \), material properties, and cross sectional geometric characteristics in the local axis system.

f) A local axis system where the longitudinal axis is defined as the axis from node \( n_i \) to \( n_j \).

It is clear that the stiffness matrix of a member is either linear or nonlinear. The DOF of the member that will be influenced due to that distinction are the DOF of its boundary nodes, DOF of its intermediate nodes, and DOF of the internal nodes in the elements which belong to that member. This set of DOF also corresponds to a submatrix in the global stiffness matrix.

The collection of all the members, and their connectivity through the boundary nodes, is defined as the physical Model. If all the members in the physical model are interconnected such that they form one elastic continuum, the model is said to be a Structure \( S \), which will be modeled as a discrete system using the Finite Element Method.

The Constraints are a set of \( ND_c \) equations constraining a number of DOF which belong to boundary nodes. The Load will be introduced in the form of a load distribution vector, \( F \), associated with an amplitude, \( \alpha \). Let \( F^T = (f_{1}^T, \ldots, f_{ND_a}^T) \) where \( ND_a \) is the number of active DOF. Hence, \( ND_a = ND - ND_c \). We assume in this study that the load is conservative, and
that the material properties of the finite elements will depend on the load amplitude, $\alpha$.

The present analysis can also be extended to investigate:

1. Curved beam and arch members.
2. The effect of joint types (DOF compatibility conditions) at the boundary nodes.
3. The influence of geometric imperfection of members using perturbation methods.
4. The mechanism of developing plastic hinges in members.
5. The influence of member fracture on the overall structure.
6. Thermal loading effects on the material properties.

2.2. Formulation of the Nonlinear Model for LSS

2.2.1. Nonlinear finite element methods

In surveying the literature, it is noted that the methods which have been adopted in deriving the governing nonlinear equations are either based on the finite element concept [13-15] or a flexibility approach which is a direct consideration of forces [16,17]. Recently, advances in finite element research have combined these two methods in a hybrid formulation to provide the most powerful tool in developing general purpose nonlinear structural analysis programs.
Most finite element techniques for nonlinear analysis are based on using either asymptotic (perturbation) method or the incremental/iterative (step-by-step) approach. The former method is an adaptation of Koiter’s perturbation procedure for the study of the immediate postbuckling response and the sensitivity of the structure to initial imperfections [18]. At this time, the conventional incremental/iterative method appears to be computationally more rewarding, but it has its difficulties in predicting the branching points in the solution. In this research, an integrated method is suggested which is in fact a combination of both methods.

2.2.2. Large displacement formulations

For a large displacement formulation, describing the motion of a structural body, essentially two distinct approaches exist. The Lagrangian (material) description in which the initial configuration is taken as the reference state, and the Eulerian (spatial) description, in which the final deformed configuration is taken as a moving reference state are used. Computationally, an Eulerian formulation is strictly an updated Lagrangian approach where the initial position becomes the current equilibrium state prior to some incremental change [19]. Therefore, the formulations are termed total Lagrangian and updated Lagrangian formulations. Under consistent physical laws both formulations are
theoretically equivalent because the same equilibrium
principles are used. Therefore, the choice between the
formulations depends on the relative ease of application and
efficiency of the formulations.

Even though the total Lagrangian formulation (TL) has
successfully been used for both geometrical and material
nonlinearities while the updated Lagrangian formulation (UL)
has been less successful until recent years. It is now
recognized that the updated formulation often presents more
simplicity than the total formulation especially for large
displacement—small strain problems.

Because of the complex nature of nonlinear structural
problems various simplifying assumptions are imposed on the
magnitude of deformations, rotations and strain, the material
properties, the type of loadings, etc., which limits the
practical usage of the solution method developed. Also, the
limitations on computer hardware resources restrict the size
of the problem and the effectiveness of the solution schemes
implemented.

By considering large displacements, large rotations, and
small strains only geometric nonlinearities are taken into
account. However, provided with proper constitutive material
laws and time integration schemes the analysis may be extended
to include material nonlinearities and dynamic loadings.
2.2.3. **LSS global static equilibrium equation**

In any formulation, the static equilibrium equation at the load amplitude \( \alpha \), in the global axis system, can be stated in the form:

\[
[K] \cdot U = \alpha F 
\]  

(2.3.3)

where

\[
[K] = [K_L] + [K_N] 
\]

\([K_N]\) is the nonlinear part of the global stiffness matrix, \([K]\), due to geometric, initial strain, temperature and material nonlinearities.

\([K_L]\) is the linear part of the global stiffness matrix, \([K]\).

\(U\) is the DOF solution vector.

Without loss of generality, the load amplitude, \( \alpha \), is assumed to be a monotone variable associated with a constant load distribution vector \( F \), and that \([K_N] = [K_N(\alpha)]\). Equation (2.3.3) can be reformulated as a perturbation or an incremental equation without any influence on the analysis.

2.3. **LSS Nonlinear Analysis Procedure**

The size of the LSS model will not allow a nonlinear analysis to be performed, in a feasible way, without a reduced order formulation. It is also important to note that the nonlinear problem of LSS is characterized by the initiation and growth (spreading) of the nonlinear regions as a function
of the load. This implies that the nonlinear part of the stiffness matrix, \([K_N]\), is experiencing a (fill-in) in its entries as the load amplitude \(\alpha\) increases. Taking advantage of the sparsity of \([K_N]\), allows a condensation of the linear equations of the structure and requires a solution to the reduced order nonlinear equations for a given load amplitude increment, \(\Delta \alpha\). The next step is to perturb the solution at the new load amplitude to include the effect of the additional nonlinear members. This procedure will increase the dimension of the nonlinear problem as additional nonlinear members are included.

2.3.1. **The margin of linearity**

Condensation of the linear members in LSS models depends mainly on the definition of the word linear. If we adopt the exact definition of the word, all the members will be nonlinear for \(\alpha \neq 0\), i.e., we will have only nonlinear structural problems. Since the linear analysis is a reasonable and acceptable approximation for small displacement formulations, we will define a margin in which the member performance can be assumed to be linear. The linear performance index of the member, \(m_j\), at the load amplitude \(\alpha\), \(LP\alpha\), is defined as the absolute value of the ratio of the nonlinear strain energy to the linear strain energy stored in that member at the prescribed load amplitude.
where

$U_{m_i}$ is the DOF solution vector for the member $m_i$ in global axis system

$[K_{N_i}]$ is the nonlinear part of the member stiffness matrix in global axis system

$[K_{L_i}]$ is the linear part of the member stiffness matrix in global axis system.

The choice of an upper limit for $LP^\alpha_i$, which will be denoted by $\chi$, is a question with no definite answer. Its answer is somewhere between the two extremes. If $\chi$ is too small, the computational effort will increase, whereas if $\chi$ is too large this will decrease the accuracy of the analysis. A reasonable value ($\chi = 0.1$) is suggested since it was acceptable to assume a linear analysis with valid engineering results for several cases with such value for $\chi$. The member, $m_i$, is said to be linear if ($0 \leq LP^\alpha_i \leq \chi$), otherwise, it is said to be nonlinear. The margin, $[0, \chi]$, is said to be the Margin of Linearity, and is denoted by (ML).

2.3.2. Nonlinear checking priority

Checking the nonlinearity of the members is carried out according to a certain order (priority). Before the first incremental analysis, we can expect that certain members will
have higher tendencies of becoming nonlinear. The list will include:

1. Members with direct applied loads.
2. Members with constrained boundary nodes.
3. Members with longer spans.
4. Members with unfavorable material properties or cross-sectional geometric characteristics.

After the first incremental analysis, the linear members can be ordered in a descending order of their linear performance index. This order is said to be the **Priority**.

### 2.3.3. Analysis procedure

**Step 1.** A linear analysis for the LSS is carried out.

**Step 2.** A nonlinearity check which will lead to establishing the priority, and estimating the load step that will change the first member to be nonlinear.

**Step 3.** Beginning from the load amplitude that will change the first member to be nonlinear, and in load increments, $\Delta \alpha$.

1. Reduce the order of the nonlinear problem by condensing out all the linear members.
2. Using an incremental method, solve the condensed nonlinear problem, for the nonlinear solution vector, $U_N$. 
3. Find the linear solution vector, \( U_L \), using the nonlinear solution vector, \( U_N \), in a back substitution process.

4. Update the linearity performance index, \( LP_{\alpha_1} \), according to the priority and check for new nonlinear members at that load amplitude.

5. Perturb the static equilibrium equation to include the effect of the new nonlinear members, which will increase, in turn, the dimension of the condensed static equilibrium equation.

6. Update the condensed equation and take another incremental load step.

Each one of these steps is a collection of topics for research by itself. In the present work, the emphasis will be to address the problem of condensation of the linear members and to update that condensation as a function of the load amplitude. The main concern during this process is to minimize the fill-ins in the condensed stiffness matrix \([R_\alpha]\).
3. GRAPH REPRESENTATION OF LSS WITH NONLINEAR REGIONS

The graph of LSS, $G(S)$, is a useful tool for linear analysis of large complicated structural continuum. Whereas, graph theory representation becomes a necessity when nonlinear regions develop in this continuum during loading.

3.1. Basic Definitions and Theorems of Graph Theory

A finite undirected graph $G = (N, H)$ consists of a finite set $N$, called nodes, and a set $H$ of unordered pairs of distinct nodes $(n_i, n_j)$, called edges. For any node $n_i$ in $G$, the set of nodes adjacent to $n_i$, $\text{adj}(n_i)$ is defined as $\text{adj}(n_i) = \{n_j \in N \mid (n_i, n_j) \in H\}$. The number of nodes in $\text{adj}(n_i)$ is denoted by $|\text{adj}(n_i)|$. The deficiency of $n_i$, $\text{def}(n_i)$, is the set of distinct pairs of nodes in $\text{adj}(n_i)$ which are not themselves adjacent. A graph is complete if every pair of nodes is adjacent. A subgraph $G' = (N', H')$ of $G$ is a graph for which $N' \subseteq N$ and $H' \subseteq H$. A complete subgraph is called a clique.

A section graph $G(N')$ is a subgraph $G = (N', H(N'))$ induced by a node set $N'$, where $H(N') = \{(n_i, n_j) \in H \mid n_i, n_j \in N'\}$.

If a set of nodes $N''$ is deleted from a graph $G = (N, H)$, the section graph $G(N \setminus N'')$ is obtained from $G$ by removing the nodes $N''$ together with their incident edges. A connection is a progression $\{n_1, h_1, n_2, h_2, \ldots, h_{k-1}, n_k\}$ where $h_i = (n_i,$
If the nodes are distinct, the connection is referred to as a path $p(n_i, n_k)$. The number of edges in a connection or path is called the length of the connection or path. The path is the shortest connection.

A graph is said to be connected if there is at least one path between every pair of distinct nodes in $G$; otherwise, $G$ is disconnected. A connected subgraph is called a component and a maximal component is called a strong component. The nodes of any graph can be partitioned into mutually exclusive strong components. However, not every edge of the graph necessarily lies in a strong component.

The distance $d(n_i, n_j)$ between two nodes $n_i$ and $n_j$ in a connected graph is the number of edges in the shortest path joining nodes $n_i$ and $n_j$. The eccentricity $e(n_i)$ of a node $n_i$ is then given by $e(n_i) = \max \{d(n_i, n_j) / n_j \in N\}$. The diameter $\rho(G)$ of the graph $G = (N, H)$ is defined as $\rho(G) = \max \{e(n_i), n_i \in N\}$. The node $n_i$ is said to be a peripheral node if $e(n_i) = \rho(G)$, $P(G)$ is the set of all peripheral nodes of a graph $G$.

A circuit of length $k$ in the graph $G$ is called the path which has $k$ edges and in which the first and the last node coincide. The graph $G$ is called acyclic if it includes no circuit. Let $G = (N, H)$, then its subgraph $T = (N_T, H_T)$ will be called a tree of graph $G$, if $G$ as well as $T$ have the same number of nodes ($|N_T| = |N|$) and if $T$ is acyclic. The
edges of tree $T$ will be called tree branches and the other edges chords $F$. Hence, $|F| + |H| = |H_T|$. For a given tree $T$ of a graph $G$, a basic circle, $k \in K$, may be assigned to each chord, $f \in F$, so that $|K| = |F|$. In general, the graph $K$ can contain several trees. Their number may be found in a purely algebraic way.

For a graph $G = (N, H)$ with $N_n$ nodes, $N_m$ edges

i) A node ordering of $N$ is a bijection (a one-to-one onto mapping).

$$\beta : \{1, 2, \ldots, N_n\} \mapsto N.$$ 

The integer, ranging from 1 to $N_n$, assigned to a node by the node ordering is called a number of that node. The node ordered graph of $G$ is denoted by $G_\beta = (N, H, \beta)$.

ii) An edge ordering of $H$ is a bijection.

$$\gamma : \{1, 2, \ldots, N_m\} \mapsto H.$$ 

The integer, ranging from 1 to $N_m$, assigned to an edge by the edge ordering is called a label of that edge.

iii) If the graph $G$ is both numbered and labelled, it is denoted by $G_{\beta \gamma} = (N, H, \beta, \gamma)$.

A level structure $LS$ of a graph $G$ is an arrangement of the nodes of $G$ into $k$ levels, $LS_1, \ldots, LS_k$ such that nodes at a given level $LS_i$ are connected to no nodes other than $LS_{i-1}$, $LS_i$, $LS_{i+1}$. For a node $n_i \in N$, there corresponds a level structure rooted at $n_i$. The levels of a rooted level structure are determined by:
i) Assigning \( L_{S1} = \{n_i\} \).

ii) For \( i > 1 \), assigning to \( L_{Si} \) the set of nodes adjacent to nodes in \( L_{Si-1} \) which have not yet been included in any previous levels.

The level structure is commonly used to describe the properties of many ordering schemes.

3.2. Graph Representation of LSS

3.2.1. Nodes, members, loads

The boundary nodes, \( N \), is a set of numbers, \( N = \{n_i, i=1, \ldots, |N|\} \), where \( n_i \) is assigned to the \( i^{th} \) boundary node. The member nodes, \( M \), is a set of numbers, \( M = \{m_i, i=1, \ldots, |M| / m_i \in N\} \), where \( m_i \) is assigned to the \( i^{th} \) member, it is assigned for every \( m_i \) a distinct unordered pair \((n_i, n_j), n_i, n_j \in N\).

A graph of a boundary node, \( G(n_i) \), is a graph with the DOF of the boundary node \( n_i \), as its nodes. We will consider that all the nodes in this graph are pairwise adjacent, i.e., the graph of a boundary node is a clique. This graph corresponds to a submatrix in the global stiffness matrix of a structure; this submatrix is considered to be an entity.

The size of a node \( n_i, |z(n_i)| \), is the number of all DOF in the set \( z(n_i) \), \( z(n_i) \) is the set of all DOF of the \( i^{th} \) boundary node.
A **graph of a member**, $G(m_i)$, is a path which can be identified by the progression $(n_i, e_i, n_{i+1}, \ldots, e_{j-1}, n_j)$, where $(n_i, \ldots, n_j)$ are the nodes, $(e_i, \ldots, e_{j-1})$ are the edges that are assigned to the elements of the member. $n_i, n_j$ are the peripheral nodes that are assigned to the boundary nodes of the member $m_i$. The other nodes are assigned one-to-one to the intermediate nodes in their order. Without loss of generality, we will assume that there exists only one **member node**, $(m_i)$, that contain all the intermediate nodes in the member; there is a one-to-one correspondence between a member and its member node. The **graph of the member node**, $m_i$, is the graph with the DOF of all the intermediate nodes, $n_{i+1}, \ldots, n_{j-1}$, as its nodes. We will assume in this study that the graph of a member node is a clique that corresponds to a submatrix in the global stiffness matrix of a structure. This submatrix is considered to be an entity. The **size of a member** $m_i$, $|z(m_i)|$, is the number of all DOF in the set $z(m_i)$, $z(m_i)$ is the set of all DOF of the intermediate nodes in the $i_{th}$ member.

An **influence line**, $w_{ij}$, is an imaginary line passing between a boundary node, $n_i$, and a member node, $m_j$, $n_i \in \text{adj}(m_j)$.

A **load brick**, $f_i$, is a vector with the same size as $n_i(m_i)$ and is associated with $n_i(m_i)$. It is also considered as an entity.
3.2.2. Models, structures, substructures

A model, \( S, S = R(S) \cup D(S) \), where \( D(S) \subseteq M \), 
\[ R(S) = \{ n_i \in N / n_i \in \text{adj}(m_j), m_j \in D(S) \} \], where \( |D(S)| \) is 
said to be the degree of the model \( S \), \( |R(S)| \) is said to be 
the expanded size of the model \( S \). The graph of a model \( S \), 
\[ G(S) = (S,H) \] with the model \( S \), as nodes, and 
\[ H = (h_i, i=1, \ldots, |H|) \], as edges, where \( h_i \) is a distinct 
unordered pair \( (n_i,m_j), n_i \in R(S), m_j \in D(S) \) \( h_i \) is assigned to 
the \( i^{th} \) connection between the node \( n_i \), the member \( m_j \). If \( G(S) \) 
is connected, the model \( S \) is defined to be a structure \( S \), 
\[ G(S) \equiv G(S) \].

The valency of a node \( n_i \), \( V(n_i) \), is the number of edges 
incident at \( n_i, n_i \in R(S) \). The valency of a member \( m_j \), \( V(m_j) \), 
is the number of edges incident at \( m_j, m_j \in D(S), V(m_j) = 2 \). 
The internal influence of a structure \( S \), \( W_0(S) \), 
\[ W_0(S) = (w_{ij} / n_i, m_j \in S) \].

A structure matrix of a model \( S \), \( [X(S)] \), is a 
\( (|S| \times |S|) \) Boolean symmetric matrix, \( |S| \) is the number 
of nodes in the graph \( G(S) \), where

i) \( X_{ii} = X_{jj} = X_{ij} = X_{ji} = 1 \) if there exist an edge 
\( h_i = (n_i, m_j) \) which is connecting the boundary node 
\( n_i \), to the member node \( m_j, n_i \in R(S), m_j \in D(S) \).

ii) \( X_{ij} = 0 \), otherwise.
The **connectivity matrix** of a model $S$, $[Q(S)]$, is a $(|H| \times |H|)$ Boolean matrix, $|H|$ is the number of edges in the graph $G(S)$, where

i) $Q_{ij} = Q_{ji} = 1$ if there exist a node in the graph $G(S)$ such that it is connecting topologically $h_i, h_j$.

ii) $Q_{ij} = 0$, otherwise.

A **graph of a matrix** $[A](N \times N)$, $G[A]$, is a graph with $N$ nodes $d_1, \ldots, d_N$ in which there is an edge from $d_i$ to $d_j$ if the entry $A_{ij} \neq 0$, $i \neq j$. The **connectivity graph**, $G(S)$, is the graph of the connectivity matrix of the model $S$. This graph represents topologically the connectivity of the edges in $G(S)$. The nodes in this graph correspond to $h_i \in H$,

$G(S) = (S, H)$. The edges describe the interconnections between the edges.

$S_i$ is defined to be a **substructure** of the structure $S$, if $G(S_i)$ is a connected subgraph of $G(S)$. $C_i$ is the **complementary model** of the substructure $S_i$, if $S_i \cup C_i = S$, $S_i \cap C_i = \emptyset$. If $S_i$ is a substructure of the structure $S$, $G(S_i) = (S_i, H_i)$,

$G(C_i) = (C_i, H_i)$ and

i) $n_i \in S_i$, $n_i \in \text{adj}(m_j)$, $m_j \in D(C_i)$, $n_i$ is defined as an **internal node** of the substructure $S_i$. $Z(S_i)$ is the set of all internal nodes of $S_i$.

ii) $n_i \in R(S)$, $n_i \in \text{adj}(m_j)$, $m_i \in D(S_i)$,

$n_i \in \text{adj}(m_j)$, $m_j \in \text{b}(C_i)$, $n_i$ is defined as a
peripheral node, $P(S_i)$ is the set of all peripheral nodes of $S_i$.

The size of a substructure $S_i$, $|Z(S_i)|$, is the number of nodes in the set $Z(S_i)$.

The substructures $S_1$, $S_2$, $S_1 \subseteq S_1$, $S_2 \subseteq S$, are said to be disjoint if $S_1 \cap S_2 = \emptyset$. If there exists a node $n_j$, such that $n_j \in P(S_1)$, $n_j \notin P(S_2)$, $S_1$, $S_2$ are said to be connected. If not, $S_1$, $S_2$ are said to be disconnected.

If $S_1$, $S_2$, ..., $S_k$ are substructures of the structure $S$, $G(S_i) = (S_i, H_i)$, $i=1, \ldots, k$, $S = \cup_i S_i$, $H = \cup_i H_i$,

$G(S) = (S, H)$, then $S^*_i$ is a strong substructure of $S$, if $G(S^*_i)$ is a strong component of $G(S)$. $S$ can be rearranged into strong substructures $S^*_i$, $(i=1, \ldots, k)$,

$S^* = (S^*_1, S^*_2, \ldots, S^*_k)$, where $k = |S^*|$, $|S^*|$ is the number of the strong substructures in the model $S$. This process of rearranging the substructures into strong substructures will prove to be useful in our analysis.

A substructure ordering of $S^* = (S^*_i, i=1, \ldots, k)$ is a bijection. $\delta : \{1, 2, \ldots, k\} \leftrightarrow S^*$, where $k$ is the number of strong substructures in the model $S$. The integer, ranging from 1 to $k$, assigned to a substructure by this order is called its index. Denote the ordered set of $S^*$ by $S^*_\beta = (S^*, \beta)$. 
3.3 Rearranging Algorithms

One of the basic procedures in the topological representation of a model $S$ is to rearrange it into its strong substructures, using the graph theory, and study their interaction with each other. By definition, the graph of each strong substructure is in fact a strong component in the graph of the model $S$, $G(S)$, which will lead us to develop the rearranging algorithms.

3.3.1. SR (Structure Rearranging) algorithm

The purpose of this algorithm is to find the substructures corresponding to the strong components of a graph of a model $S$.

**Step 1:** Establish the structure matrix of the node ordered model $S_{\zeta}$, $[X] = [X(S_{\zeta})]$ $(N \times N)$; $N$ is the number of nodes from its graph $G_{\zeta}(S)$,

$\zeta : (1, 2, \ldots, N) \leftrightarrow S$.

**Step 2:** Find the matrix $[X^i]$ by Boolean multiplication where $i$ is either determined by $[X^i] = [X^{i-1}]$ (i.e., $1+1=1$, $1+0=1$, $0+0=0$ rules) or $i = N-1$. This matrix is called the reachability matrix (or path matrix) and is denoted by $[A]$ [20].

**Step 3:** Let $\zeta_1$ be the collection of all the indices, $i$, such that there appears the number "1" in both the $i^{th}$ place of the first row of $[A]$ and of the first column of
\( \{A\} \). These indices name the nodes of \( G_\xi(S) \) in the same strong component as the node with the first index. Delete from the matrix, \( [A] \), all rows and columns whose indices are in the set, \( \zeta_1 \), but preserve the original row and column indices from \( [A] \) in the resulting sub-matrix of \( [A] \).

**Step 4:** Repeat Step 3 on this resulting sub-matrix, obtaining the collection of sets of indices \( \zeta_1, \zeta_2, \ldots, \zeta_k \). Then, clearly \( \zeta_1, \zeta_2, \ldots, \zeta_k \) are pairwise disjoint sets of indices whose union is \( \{1, 2, \ldots, N\} \) and each set, \( \zeta_j \), contains all the indices which belong to the nodes in one strong component of \( G_\xi(S) \).

**Step 5:** Rearrange \( [X] \) into a \( k \) block diagonal matrix, each block represent the structure matrix of a strong substructure.

**Step 6:** Order each set \( \zeta_j \).

The numerical effort involved in reducing the original matrix is nominal as the matrices involved are Boolean. The multiplications can be done row by column wise using an assembly level language. The memory involved is also nominal as every element can be stored in a bit instead of a word. Thus, this procedure is very attractive for large dimensional systems.
3.3.2. MR (Model Rearranging) algorithm

Given $S^*$, $S^*$, $S$, $S^*$ is arranged in its strong substructure form, $S^* = (S_{i_1}^*, i=1, \ldots, |S^*|)$, $m_i \in D(S)$, $m_i$ is assigned to the $i$th member which is connecting the boundary nodes $n_1, n_2$.

Algorithm A: - If $m_i \notin S_{i_1}^*$, $S_{i_1}^* \notin S^*$ and $m_i$ is added to the model $S^*$ to form $\bar{S}$, $D(\bar{S}) = D(S^*) \cup m_i$. Then, $\bar{S}$ can be rearranged into its strong substructures, $\bar{S}^* = (S_{i_1}^*, i=1,2, \ldots, |\bar{S}^*|)$, using the following algorithm.

Step 0: If $S^*$ is an empty set, then $|S^*| = 0$, go to step 4.

Step 1: If $n_1 \notin P(S_{i_1}^*)$, $n_2 \notin P(S_{j_1}^*)$, $(i,j=1,2, \ldots, |S^*|)$. Then, the number of strong substructures will not be changed, $|\bar{S}^*| = |S^*|$. $G(S_{i_1}^*)$ can be updated by adding two edges $(n_1,m_i)$, $(m_i,n_2)$ to the graph to form $G(\bar{S}_{i_1}^*)$, $D(\bar{S}_{i_1}^*) = D(S_{i_1}^*) \cup m_i$, $|D(\bar{S}_{i_1}^*)| = |D(S_{i_1}^*)| + 1$. An ordering procedure will follow to renumber the members of $S_{i_1}^*$.

Step 2: If $n_1 \notin P(S_{i_1}^*)$, $n_2 \notin P(S_{j_1}^*)$, $(i=1, \ldots, |S^*|, i \neq j)$. Then, the number of strong substructures will be decreased by 1, $|\bar{S}^*| = |S^*| - 1$. $G(S_{i_1}^*)$, and $G(S_{j_1}^*)$ will be connected together by two new edges $(n_1,m_i)$, $(m_i,n_2)$ to
form $G(S^*_i), D(S^*_i) = D(S^*_i) \cup D(S^*_j) \cup m_i$.

$\mid D(S^*_i) \mid = \mid D(S^*_i) \mid + \mid D(S^*_j) \mid + 1$. An ordering procedure will follow to renumber:

i) The strong substructures $S^*_i$.

ii) The members of $S^*_i$.

**Step 3:** If $n_1, n_2 \in P(S_i), (i=1, \ldots, \mid S^*_i \mid)$, $D(S^*_i) = D(S^*_i) \cup m_i$, $\mid D(S^*_i) \mid = \mid D(S^*_i) \mid + 1$.

Ordering of members of $S^*_i$ will follow.

**Step 4:** If $n_1 \in P(S^*_i), n_2 \in P(S^*_j), (i,j=1, \ldots, k)$, $k = \mid S^*_i \mid$. Then, a new strong substructure will be formed $S^*_{k+1}, D(S^*_{k+1}) = \{m_i\}$, its graph $G(S^*_{k+1})$ has two nodes $n_1, n_2$ and two edges connecting $n_1$ to $n_2$, $(n_1, m_i)$, $(m_i, n_2)$. $\mid S^*_i \mid = \mid S^*_i \mid + 1, \mid D(S^*_{k+1}) \mid = 1$. An ordering procedure will follow to renumber the strong substructures. Member $m_i$ will be labelled 1 in $S^*_{k+1}$.

**Algorithm B:**

If $m_i \in S^*_i, S^*_i \in S^*, m_i$ is taken off the model $S^*$ to form $S, D(S) = (D(S^*) \setminus m_i)$, then $S$ can be rearranged using the following algorithm.

**Step 1:** Find the strong substructure $S^*_i$, where $m_i \in D(S^*_i)$, define $S_i, D(S_i) = (D(S^*_i) \setminus m_i)$. If $\mid D(S_i) \mid = 0$, then $S^*_i = \emptyset, \mid S^*_i \mid = \mid S^*_i \mid - 1$.

**Step 2:** Check $G(S_i)$, where $G(S_i)$ can be constructed by deleting the edges that are connecting $n_1, n_2$ to $m_i$. 


a) If \( G(S_i) \) is a connected graph, then \( S^*_i = S_i \), reorder the members of \( S_i \). 
\[ |D(S_i^*)| = |D(S_i)| - 1, \]
\[ |S^*| = |S_i^*|. \]

b) If \( G(S_i) \) is a disconnected graph, \( G(S_i) \) can form at most two strong subgraphs \( G(S'_i), G(S''_i) \). Where \( S'_i, S''_i \) are two strong substructures of the model \( (S_i) \). The number of strong substructure will be increased by 1.
\[ |S^*| = |S_i^*| + 1, \]
\[ S^* = (S_i^* \setminus S_i^*) \cup S'_i \cup S''_i. \]

The connectivity of the graph \( G(S_i) \) can be checked using its structure matrix and the SR algorithm which will establish not only \( G(S'_i), G(S''_i) \), but also \( S'_i, S''_i \) and their structure matrices. An ordering procedure will follow to renumber:

i) The strong substructures, \( S^* \).

ii) The members of \( (S'_i, S''_i) \), or \( S_i^* \).

If the rearranging algorithm is carried out for more than one member \( m_i \), then Algorithm A (B) will be executed for every \( m_i \), and the ordering procedures is carried out later.

3.4. Linear and Nonlinear Models in LSS

Assign a Boolean variable \( t_i \) to the member \( m_i \), \( m_i \in D(S) \), \( i=1, \ldots, |D(S)| \). A member \( m_i \) is said to be a \textbf{linear} \( \textbf{(nonlinear)} \) \textit{member} if \( t_i = 0 \) \( (t_i = 1) \).
A node which is connected only to nonlinear (linear) members is said to be a \textit{nonlinear (linear) node}, and all DOF associated with it \textit{nonlinear (linear) DOF}. Otherwise, they are said to be \textit{transition node (DOF)}.

If $t : \mathbb{R} \rightarrow \text{Boolean}$, $t_i = t_i(\alpha)$, $\alpha$ is said to be the \textit{load amplitude}, $\alpha_{\text{min}}$ is the minimum load amplitude such that the structure $S$ has at least one nonlinear member. $\alpha_{\text{max}}$ is the minimum load amplitude such that the structure $S$ has no linear members. $(\alpha_{\text{cr}})_i$ is the least load amplitude at which $m_i$ is nonlinear. The \textit{linear valency of a node $n_i$} at the load amplitude $\alpha$, $\nu^L(n_i)$, is the number of linear members that are connected at $n_i$, $n_i \in \mathbb{R}(S)$.

A structure $S$, at the load amplitude $\alpha$, is defined to be \textit{linear (nonlinear) structure} if $t_i = 0 (t_i = 1) \ orall m_i \in \mathbb{D}(S)$.

The \textit{linear model} $S^L$ and the \textit{nonlinear model} $S^N$ of a structure, $S$, at the load amplitude $\alpha$, are defined as:

$D(S^L) = \{m_i \in \mathbb{D}(S) / t_i = 0\}$, $D(S^N) = \{m_i \in \mathbb{D}(S) / t_i = 1\}$;

$Z(S^L) = \{n_i \in \mathbb{R}(S) / n_i \text{ is linear}\}$, $Z(S^N) = \{n_i \in \mathbb{R}(S) / n_i \text{ is nonlinear}\}$, $S^L = D(S^L) \cup Z(S^L)$, $S^N = D(S^N) \cup Z(S^N)$.

i) For $\alpha < \alpha_{\text{min}} : S^N = \emptyset$, $S^L = S$.

ii) For $\alpha \geq \alpha_{\text{max}} : S^N = S$, $S^L = \emptyset$.

iii) For $\alpha_{\text{min}} \leq \alpha < \alpha_{\text{max}} : S^N \neq \emptyset$, $S^L \neq \emptyset$.

$S^N$, $S^L$ can be arranged into two sets of strong substructures.

Nonlinear substructures, $S^N_j$, $S^N_j = (aS^N_j, J=1, \ldots, |S^N_j|)$;

linear substructures, $S^L_I$, $S^L_I = (aS^L_I, I=1, \ldots, |S^L_I|)$. It
follows from the definitions that any nonlinear strong substructure, \( \alpha S_{N_j} \), is connected only to linear strong substructures, and vice versa. Define the set, \( S^\alpha \),
\[ S^\alpha = S_{N}^\alpha \cup S_{L}^\alpha , \quad |S^\alpha| = |S_{N}^\alpha| + |S_{L}^\alpha| , \]
\[ S^\alpha = (S_{N}^\alpha, I=1, \ldots, |S^\alpha|) . \] The ordered set \( S^\alpha_\beta = (S^\alpha, \beta) \),
\[ \beta : \{1,2,\ldots, |S^\alpha|\} \leftrightarrow S^\alpha . \]

The expanded substructure, \( S_{EI}^\alpha (S_{LEI}^\alpha; S_{NEI}^\alpha) \),
\[ aS_{EI}^\alpha = aS^\alpha_{I} \cup P(aS^\alpha_{I}), \quad aS^\alpha_{I} \in S^\alpha_{I}, \quad I \in \beta \] is said to be the expanded size of \( aS^\alpha_{I} \).

We can prove that all peripheral nodes for a linear (nonlinear) substructure are transition nodes, all peripheral nodes for a linear (nonlinear) expanded substructure are also transition nodes, and all linear (nonlinear) substructures are disconnected.

A super node, \( N^\alpha_{IJ} \), is the set of all transition nodes that are connecting a linear and a nonlinear strong substructures at a certain load amplitude \( \alpha \).
\[ N^\alpha_{IJ} = N^\alpha_{JI} = \{n_i \in N / n_i \in P(aS^\alpha_{I}), n_i \in P(aS^\alpha_{J})\} . \]
The super node \( N^\alpha_{IJ} \) can be represented by a distinct unordered pair \( (I,J) \), \( I,J \in \beta \). Define the super nodes set
\[ N^\alpha = (N^\alpha_{IJ} / I, J \in \beta) , \] the ordered set \( N^\alpha_\gamma = (N^\alpha, \gamma) \),
\[ \gamma : \{1,2,\ldots, |N^\alpha|\} \leftrightarrow N^\alpha . \]

The valency of a super node, \( N^\alpha_{K} \), \( V(N^\alpha_{K}) \), is the number of strong substructures \( aS^\alpha_{I} \) that are connected to \( N^\alpha_{K} \), where
\[ m_i \in aS^\alpha_{I} , \quad m_i = \text{adj}(n_k) , \quad n_k \in N^\alpha_{K} . \] It is obvious that \( V(N^\alpha_{K}) = 2 \).
The valency of a substructure, $\alpha_S$, $V(\alpha_S)$, is the number of super nodes $N^a_K$ that are connected to $\alpha_S$, where $n_k \in N^a_K$, $n_k = \text{adj}(m_i)$, $m_i \in \alpha_S$. For $\alpha'' > \alpha'$, $S''_N \cap S''_N \neq \emptyset$, where $S''_N = S''_N$, $S''_N = S''_N$, we can rearrange the linear and the nonlinear models using the MR algorithm.

The block structure, $\alpha_S$, $S = S \cup N^a$, is the set of all substructures and supernodes of the structure, at the load amplitude $\alpha$. The ordered set $\alpha_S = (\alpha_S, \delta)$, $\delta : \{1, 2, \ldots, | \alpha_S | \} \leftrightarrow \alpha_S$.

The ordered set $\alpha_S(I) = (\alpha_S, \zeta(I))$, $\alpha_S(I) \in S, I \in \delta$, $\zeta(I) : \{1, 2, \ldots, | \alpha_S | \} \leftrightarrow \alpha_S$, is the set of all ordered nodes, $n_i$, members $m_i$, and $n_i, m_i \in \alpha_S$. The ordered set $N^a_K = (N^a_K, \zeta(K))$, $N^a_K \in N^a, K \in \delta$, $\zeta(K) : \{1, 2, \ldots, | N^a_K | \} \leftrightarrow N^a_K$ is the set of all ordered nodes, $n_k, n_k \in N^a_K$.

Internal influence of a substructure $\alpha_S$, $w_{\alpha_0}(I), I \in \delta$, $w_{\alpha_0}(I) = \{w_{ij} / n_i, m_j \in \alpha_S, \alpha_S \in \alpha_S\}, \alpha_0 = \cup_I w_{\alpha_0}(I)$.

A cut, $w_{kj}$, is an influence line passing between a boundary node $n_k \in N^a_K$, and a member node $m_j \in \alpha_S, I, K \in \delta$, $m_i \in \text{adj}(n_k)$.

A front, $w^a(K,I), w^a(I,K) = w^a(K,I) = \{w_{kj} / n_k \in \alpha^a_S, m_j \in \alpha^a_S, \alpha^a_S \in \alpha^a_S\}, \alpha_S \in N^a, \alpha_S \in \alpha_S, I, K \in \delta$, can be represented by a distinct unordered pair $(K,I), K, I \in \delta$.

There exist two fronts for every $N^a_K \in N^a, K \in \delta$. A passive front, $w^a(K,J), w^a(K,J) = w^a(K,J), \alpha_S \equiv \alpha_S_N$. 


\[ \alpha_{SN_j} \in S^\alpha_N, \ k \in \delta. \] And an active front, \( \mathcal{W}^\alpha(K,J), \)
\[ \mathcal{W}^\alpha(K,I) \equiv \mathcal{W}^\alpha(K,I), \ a_S^I \equiv a_S^{L_I}, a_S^L \in S^L. \]

The set of all passive fronts for a nonlinear substructure \( \alpha_{SN_j}, \ a_{SN_j} \in S^\alpha_N, \ j \in \delta, \) is denoted by \( \mathcal{W}^\alpha(J), \)
\[ \mathcal{W}^\alpha(J) = \{ \mathcal{W}^\alpha(J,K) / K \in \delta \}, \ | \mathcal{W}^\alpha(J) | = V(a_{SN_j}). \] Define
\[ \mathcal{W}^\alpha = (\mathcal{W}^\alpha(J) / J \in \delta) \] to be the set of all passive fronts in a structure, \( S \) at the load amplitude \( \alpha. \) The set of all active fronts for a linear substructure \( \alpha_{SL_I}, \ a_{SL_I} \in S^\alpha_L, \ I \in \delta, \) is denoted by \( \mathcal{W}^\alpha(I), \)
\[ \mathcal{W}^\alpha(I) = \{ \mathcal{W}^\alpha(K,I) / K \in \delta \}, \ | \mathcal{W}^\alpha(I) | = V(a_{SL_I}). \] Define \( \mathcal{W}^\alpha = (\mathcal{W}^\alpha(I) / I \in \delta) \) to be the set of all active fronts in a structure, \( S \) at the load amplitude \( \alpha. \) The set of all super-nodes for a substructure \( \alpha_S^I, \ a_S^I \in S^\alpha_I, I \in \delta, \) is denoted by \( \mathcal{W}^\alpha(I), \)
\[ \mathcal{W}^\alpha(I) = \{ \mathcal{W}^\alpha_{N,K} / I \in \delta, n_k \in \mathcal{W}^\alpha_{N,K}, n_k \in \mathcal{P}(a_{SI}) \}. \] The set of all fronts in a structure, \( \mathcal{W}^\alpha, \)
\[ \mathcal{W}^\alpha = \mathcal{W}^\alpha_{I} \cup \mathcal{W}^\alpha_{J}, \mathcal{W}^\alpha_{I} = \bigcup \mathcal{W}^\alpha_{I}, \mathcal{W}^\alpha_{J} = \bigcup \mathcal{W}^\alpha_{J}. \] Define \( \mathcal{W}^\alpha(I) = \bigcup K \mathcal{W}^\alpha(I,K), \)
\[ \mathcal{W}^\alpha_{+}(I) = \mathcal{W}^\alpha_{0}(I) \cup \mathcal{W}^\alpha(I). \]
4. ORDERING OF THE STRUCTURE GRAPH, $G(\alpha)$

The graph of the structure, $S$, at the load amplitude $\alpha$, $G(\alpha) = (S^\alpha, H)$, consists of the finite set $S^\alpha$, as nodes, where $S^\alpha = D(S^\alpha) \cup R(S^\alpha)$, $D(S^\alpha) = D(S_L^\alpha) \cup D(S_N^\alpha)$, $R(S^\alpha) = R(S_L^\alpha) \cup R(S_N^\alpha)$, and $H = \{h_i, i = 1, 2, \ldots, |H|\}$, as edges, where $h_i$ is a distinct unordered pair $(n_i, m_j)$, $n_i \in R(S_N^\alpha)$, $m_j \in D(S^\alpha)$, $h_i$ is the $i^{th}$ connection between the node $n_i$, the member $m_j$.

To demonstrate the procedure an example is discussed throughout this and the following chapters. This example is for a large space structure shown in Figure 4.1. This structure represents a 3-hoop/column antenna, subjected to a load distribution $F$ with a given amplitude, $\alpha$.

The main carrying structure of the reflective surface of the antenna is divided into three equal segments. Each represents a linear substructure (2-4) Figure 4.2, whereas its hoops have been fractured, in several sections, leaving the nonlinear substructures (1), (6,10), (5,7,9,11) along the 3 hoops, respectively. The column and the upper control stringers form the nonlinear substructure (8) as shown in Figure 4.3. The topological relations between the substructures and each other are presented in Figure 4.4.
Figure 4.1. 3-hoop/column antenna cross section
Figure 4.2. Nodes and members of the linear substructure $\alpha_{S_L^7}$
Figure 4.3. Linear and nonlinear substructures in the antenna structure
Figure 4.4. Topological representation of the connectivity of the substructures
4.1. The Multi-Level Structure of LSS, MLS(\(\alpha\))

The structure graph, \(G(\alpha)\), has a special structure of its own, its main advantage is its flexibility to change as the load amplitude, \(\alpha\), changes. This special structure denoted by MLS(\(\alpha\)), will allow us to analyze the structure, \(S\), in different perspectives.

1. The overall relation between the strong substructure, linear and nonlinear, can be studied through the Super Graph, \(G^*(\alpha)\), and its Structure Levels, \(LS^*(\alpha)\).

2. The interaction between two adjacent substructures connected by a super node can be examined through the Block Graph, \(G(\alpha)\), and its Substructure Levels, \(LS(\alpha)\).

3. The relations between nodes and members in the \(I^{th}\) substructure are considered through the Brick Graph of the substructure, \(G^a(I)\), and its Node Sublevels, \(LS^a(I)\).

4. The influence of the boundary node DOF or member node DOF, \(z(n^I)\) or \(z(m^I)\) on each other can only be analyzed through the graph of the boundary node \(G(n^I)\) or the graph of the member node \(G(m^I)\).

These four categories of graph levels must be examined simultaneously for a nonlinear analysis with a variable load amplitude. In the present work, it is assumed that \(G(n^I)\) and \(G(m^I)\) are clique graphs and they will be considered as basic units in the preceding third category.
4.2. The Linear Interval

For $\alpha < \alpha_{\text{min}}$, the structure, $S$, is a linear elastic continuum.

Several algorithms can be used to order the members, $m_i \in D(S)$, and the nodes, $n_i \in R(S)$ of the structure. The most used algorithms in conjunction with the finite element method are:

1. Cuthill-McKee and reverse Cuthill-McKee (RMC) ordering algorithms [21].
2. Minimum Degree ordering algorithm [22].
3. Two-step approach [23].
4. Frontal algorithms [24].

One of these methods will be used to establish an order for the nodes and members to be used for the linear analysis in this interval. Taking into consideration that each node, $n_i$, (member, $m_i$) in the structure, $S$, will be associated with one number ($i$), $i \in \zeta$, the ordered set $S_\zeta = (S, \zeta)$, $\zeta : \{1, 2, \ldots, |S|\} \leftrightarrow S$.

During the linear analysis, the priority set, $Y^0_\theta$, will be initiated and, if necessary, updated to check the nonlinearity of the members in the order of their likelihood to be nonlinear. In this interval, $Y^0 = Y^\alpha = \{i \in \zeta / m_i \in D(S)\}$. The ordered set $Y^0_\theta = Y^0_\theta = (Y^0, \theta^0)$, where $\theta^0 : \{1, 2, \ldots, |D(S)|\} \leftrightarrow Y$. The order of the set $Y$ is determined using a reasonable criterion (Section 2.3.2). The
linear interval will end with the first member to be nonlinear at \( \alpha = \alpha_{\text{min}} \).

4.3. The Transition Interval

At the load amplitude, \( \alpha, \alpha_{\text{min}} \leq \alpha < \alpha_{\text{max}} \), the structure \( S \) is composed of linear and nonlinear regions changing their configuration as \( \alpha \) changes. The multi-level structure of the structure, \( S \), at the load amplitude, \( \alpha \), MLS(\( \alpha \)), can be established in its main categories:

1. The structure levels, \( LS^*(\alpha) \), of the super graph, \( G^*(\alpha) \).
2. The substructure levels, \( LS(\alpha) \), of the block graph, \( G(\alpha) \).
3. The nodes sublevels, \( LS^\alpha(I) \), of the brick graph, \( G^\alpha(I), I \in S \).

4.3.1. The super graph, \( G^*(\alpha) \)

At any load amplitude, \( \alpha \), the Super Graph of a Structure, \( S^\alpha = G(S^\alpha, N^\alpha) \), consists of the finite set \( S^\alpha \), the set of all strong substructures in the structure \( S \), as nodes and the set \( N^\alpha \) of super nodes as edges. We can easily prove that \( G^*(\alpha) \) has a special level structure \( LS^*(\alpha) \) such that the nodes at a given level \( LS^*_i(\alpha) \) are not interconnected. The levels of \( LS^*(\alpha) \) will represent nonlinear and linear strong substructures alternatively.
Construction and ordering of LS*(α): (Figure 4.5a).

For α < α_min; G*(α) is just one node. SL*0(α) = (S).

For α_min ≤ α < α_max; G*(α) will be ordered as follows:

**Level 1, LS*1(α):** the first $aS_{N_j}$ to be formed. Index this node (substructure) as '1'.

**Level 2, LS*2(α):** All $aS_{L_I}$ which are adjacent to LS*1(α). Index these nodes (substructures) in descending order of their valency $V(aS_{L_I})$. If two nodes have the same valency, order them in descending order of their degree $|D(aS_{L_I})|$. 

**Level 3, LS*3(α):** All $aS_{N_j}$ which are adjacent to nodes in LS*2(α), and have not yet been included in any previous levels. Index these nodes in lexicographic order of their connection to the nodes in the previous level. Follow the same rule in all preceding levels. And so on.

For α ≥ α_max, G*(α) is just one node.

If two nodes in a nonlinear level will have the same lexicographic order, give a smaller index to the one that was formed first. If these two nodes exist in a linear level, index them in descending order of their valency, then to their degree.

The node ordered graph of G*(α) is denoted by $G^*_β(α) = (S^α, N^α, β)$.

The Super Tree of the Structure S at the load amplitude α, $T^*_β(α)$ can easily be constructed from LS*(α), following the
Figure 4.5a. Structure levels of the super graph $G^*(\alpha)$
simple rule that: "If $S_i$ at the LS* is connected to more than one substructure at LS*$, all the edges will be deleted, except the one which is connected to the substructure with the smallest index at that level." The super tree of a structure is used to span the substructures (width first) [25].

The Structure Level Matrix at the load amplitude $\alpha$, $[C^\alpha]$, is a $(|S^\alpha| \times |S^\alpha|)$ Boolean symmetric matrix, $|S^\alpha|$ is the number of substructures, $|S^\alpha| = |S^L_\alpha| + |S^N_\alpha|$, where:

i) $C^\alpha_{ij} = C^\alpha_{ji}$ if there exists a super node $N^\alpha_{ij}$ connecting the substructures $S_i$, $S_j$, $i, j \in \beta$.

ii) $C^\alpha_{ij} = 0$ otherwise (Figure 4.5b).

The super node $N^\alpha_{ij} \in N^\alpha$ connecting $S_i$, $S_j$, $i, j \in \beta$, $i < j$, will be ordered with (I) as a first priority, (J) as a second one. The ordered set of $N^\alpha$ is denoted by $N^\alpha_\gamma = (N^\alpha_\gamma, \gamma)$, $\gamma : \{1, 2, \ldots, |N^\alpha|\} \leftrightarrow N^\alpha$, $N^\alpha_\gamma = (N^\alpha_k, k = 1, \ldots, |N^\alpha|)$.

4.3.2. The block graph, $\tilde{G}(\alpha)$

At any load amplitude $\alpha$, the Block Graph of a structure $S$, $\tilde{G}(\alpha) = G('S^\alpha', 'W^\alpha)$, consists of the finite set $'S^\alpha = S^\alpha \cup N^\alpha$, as nodes, and the set $'W^\alpha$, as edges. $\tilde{G}(\alpha)$ has a special level structure $LS(\alpha)$ such that the nodes at a given level are not connected. The levels of $LS(\alpha)$ will represent substructures and supernodes alternatively.
Figure 4.5b. Structure level matrix \([C^\alpha]\)
Construction and ordering of $\text{LS}(\alpha)$: (Figure 4.6a).

For $\alpha_{\text{min}} \leq \alpha < \alpha_{\text{max}}$, $G(\alpha)$ will be ordered as follows:

**Level 1**, $\text{LS}_1(\alpha)$: $\text{LS}_1(\alpha) = \text{LS}^*_1(\alpha) = \{\alpha_{S_{N_1}}\}$ with index $J = 1$.

**Level 2**, $\text{LS}_2(\alpha)$: All $\alpha_{N_K}$ which are adjacent to $\text{LS}_1(\alpha)$. Index the nodes according to their order $K$, $K \in \gamma$.

**Level 3**, $\text{LS}_3(\alpha)$: All $\alpha_{S_{I_1}} \in \text{LS}^*_2(\alpha)$ with their order $(I)$, $I \in \beta$.

**Level 4**, $\text{LS}_4(\alpha)$: All $\alpha_{N_K}$ which are adjacent to $\text{LS}_3(\alpha)$ and have not yet been included in any previous level with their order.

**Level 5**, All $\alpha_{S_N} \in \text{LS}^*_3(\alpha)$ with their order $J \in \beta$. And so on. The number of levels $|\text{LS}(\alpha)|$ is odd. The ordered graph of $G(\alpha)$ is denoted by $\tilde{G}_8(\alpha) = (\tilde{S}^\alpha, \tilde{W}^\alpha, \tilde{S})$.

For $\alpha < \alpha_{\text{min}}$, or $\alpha > \alpha_{\text{max}}$, $\tilde{G}(\alpha)$ is just one node.

It is obvious that any two substructures at a certain level can not be connected to the same super node at an adjacent level.

The **Structure Block Matrix** at the load amplitude $\alpha$, $[B^\alpha]$, is a $(|\tilde{S}^\alpha| \times |\tilde{S}^\alpha|)$ Boolean symmetric matrix, $|\tilde{S}^\alpha| = |S^\alpha| + |N^\alpha|$, where:

i) $B_{IK}^\alpha = B_{KI}^\alpha = B_{II}^\alpha = B_{KK}^\alpha = 1$ if there exist a front $W^\alpha(K,I)$ between a super node $\alpha_{N_K}$, and a substructure $\alpha_{S_I}$ $I,K \in \delta$.

ii) $B_{IK}^\alpha = 0$, otherwise (Figure 4.6b).
Figure 4.6a. Block graph $\mathcal{G}(\alpha)$
Figure 4.6b. Structure block matrix [B^a]
Each one of the entries $B_{IK}^{a} \neq 0$, $I,K \in \delta$ represents a Block Stiffness Matrix $[K_{IK}^{a}]$ in the global stiffness matrix $[K]$.

Each $[K_{IK}^{a}]$ is considered to be an entity in the discussion of matrix interpretation of the graph operations in $G(\alpha)$.

4.3.3. **The brick graph, $G^{a}(I)$**

4.3.3.1. **Ordering of a super node, $N_{K}^{a}$**

The nodes $n_{K} \in N_{K}^{a}$, $K \in \delta$, will be subindexed in the order of their presence in the set $N_{K}^{a}$. This order is not restrictive, but it is easy to establish. The ordered set $N_{K}^{a}$ is denoted by $N_{K}^{a} = (N_{K}^{a}, \zeta(K))$, where $\zeta(K) : \{1,2, \ldots \mid N_{K}^{a} \mid \} \leftrightarrow N_{K}^{a}$.

Each node, $n_{K} \in N_{K}^{a}$, will be associated with an ordered pair $(K,k)$, $K \in \delta$, $k \in \zeta(K)$.

The Brick Graph of the super node $N_{K}^{a}$, $G^{a}(K)$, is a null graph which consists only of the isolated nodes $n_{K} \in N_{K}^{a}$,

$L^{a}(K) \equiv L^{a}_{1}(K) \equiv N_{K}^{a}$ (Figure 4.7).

4.3.3.2. **Ordering of a substructure $S_{I}^{a}$**

At any load amplitude $\alpha$, the Brick Graph of a substructure $S_{I}^{a}$,

$G^{a}(I) = G(S_{I}^{a}, W^{a}_{0}(I))$, consists of the finite set $S_{I}^{a} = D(S_{I}^{a}) \cup Z(S_{I}^{a})$, $S_{I}^{a} \in S^{a}$, $I \in \delta$, as nodes, and the set $W^{a}_{0}(I)$, as edges. $G^{a}(I)$ will have also a special level structure $L^{a}(I)$, $I \in \delta$, such that the nodes at a given level $L^{a}_{i}(I)$ are not connected. The levels of $L^{a}(I)$ will represent boundary nodes, and member nodes, alternatively.
Figure 4.7. Brick graph of super nodes
The most suitable method, which is also natural, for a dynamic ordering algorithm is using the active (passive) fronts to order the linear (nonlinear) substructures using a modified Reverse Wave Front (RWF) method.

Construction and ordering of \( L^S(I) \): (Figure 4.8).

**Step 0:** Using the block stiffness matrix \([B^a]\), find all the fronts \( W^a(K,I), W^a(K,I) \in W^a(I) \), then find all the super nodes \( N^a_K \in N^a(I) \). Define the Natural Order of \( n_k, n_k \in P(\alpha S_I) \), to be the order of their ordered pair \((K,k), K \in \delta, k \in \zeta(K)\).

**Step 1:** Establish the sublevels of the substructure, \( \alpha S_I, I \in \delta \), and order \( S^a(I) \) as follows:

- **Starting level,** \( L^S_0(I) \): \( n_k \in P(\alpha S_I) \) in their natural order.
- **Level 1,** \( L^S_1(I) \): \( m_i \in D(\alpha S_I), m_i \in \text{adj}(n_k), n_k \in L^S_0(I) \), ordered according to the lexicographic order of their connectivity to \( n_k \in L^S_0(I) \).
- **Level 2,** \( L^S_2(I) \): \( n_i \in Z(\alpha S_I), n_i \in \text{adj}(m_i), m_i \in L^S_1(I) \), ordered according to the lexicographic order of their connectivity to \( n_i \in L^S_1(I) \).

If two member nodes will have the same lexicographic order, then order them in a descending order of their size. Move the front (wave front) to the next level. And so on.
Figure 4.8. Brick graph of the substructure $\alpha_{S_{L,7}}$. 
Step 2: Reverse the order of the levels keeping the order of the nodes inside each level.

Step 3: Subindex all \( m_j, n_i \in \alpha S_i \), \( I \in \delta \). The ordered set \( \alpha S_i \) is denoted by \( \alpha S_i(\zeta(I)) = (\alpha S_i, \zeta(I)) \), where \( \zeta(I) : (1, 2, \ldots, |\alpha S_i|) \leftrightarrow \alpha S_i \).

It is obvious that the last level of \( G^\alpha(I) \), denoted by \( L^{\alpha}_{\max}(I) \), will only have members that are adjacent to transition nodes.

The motivation behind reversing the order, in Step 2, is the fact that the changes for a substructure usually takes place at its fronts, i.e., the changes in the subindices, as \( \alpha \) changes, will be minimal.

The Expanded Brick Graph of a substructure \( \alpha S_i \) (Figure 4.9). \( G^\alpha_+(I) = G(\alpha S_{E_i}, \alpha W_+(I)) \) is a graph with the set \( \alpha S_{E_i} \) as nodes, and the set \( \alpha W_+(I) \), as edges. It is obvious that \( G^\alpha(I) \) is a section graph of \( G^\alpha_+(I) \).

Ordering of \( G^\alpha_+(I) \): \( G^\alpha_+(I) \) is obtained directly from \( G^\alpha_+(I) \) by adding the starting level of nodes, \( n_k \in P(\alpha S_i) \) \([n_k \in L^{\alpha}_{\max}(I)]\), that are ordered \( n_k \in \alpha S_\zeta(K), K \in \delta, k \in \zeta(K) \), \( K \) is the index, \( k \) is the subindex. Using the RWF method to previously order \( \alpha S_i \) will help to assign a Dummy Index to these nodes keeping the same order they originally had.

Finally, at any load amplitude \( \alpha \), \( \alpha_{\min} \leq \alpha < \alpha_{\max} \), each node \( n_i \) and member \( m_j \), where \( n_i, m_j \in \alpha S_i \), will be associated with an ordered pair \( (I, i) \) or \( (I, j) \), \( I \in \delta, i, j \in \zeta(I) \), and
Figure 4.9. Expanded graph of the substructure $^0S_{1,7}$
each influence line \( w_{ij} \), and cut \( w_{kj}, n_k \in N\alpha_K, K \in \delta, k \in \zeta(K) \), will be associated with an ordered double-pair \([(I,i),(I,j)]\) and \([(K,k),(I,j)]\), respectively.

4.4. Storage Allocation of Block Stiffness Equation

4.4.1. Block stiffness matrix

1. The block stiffness matrix of \( \hat{N}\alpha_K, [K_{KK}], \) is a Brick diagonal matrix. Each brick, \( [K_{kk}], K \in \delta, k \in \zeta(K) \), is a symmetric matrix (Figure 4.10).

2. The block stiffness matrix of \( \alpha_S_j, [K_{jj}], \) is a banded symmetric matrix that can be partitioned to four submatrices with square submatrices on the diagonal (Figure 4.11).

\[
[K_{JJ}] = \begin{bmatrix}
[K_{bb}] & [K_{bp}] \\
- & - & - & - \\
[K_{pb}] & [K_{pp}] \\
- & - & - & -
\end{bmatrix}
\]

\[
[K_{J}] = [K_{bb}]_J + [K_{bp}]_J^T
\]

i) \( [K_{bb}]_J \), which belong to \( m_j \in L\alpha_{max}(J), J \in \delta, j \in \zeta(J) \), \( L\alpha_{max}(J) \) is the last sublevel of \( \alpha_S_j \), is a brick diagonal matrix. Each brick, \( [K_{jj}]_J \), is a symmetric matrix,

ii) \( [K_{bb}]_J \) is a symmetric banded brick matrix that belongs to \( n_j \in \alpha_S_j \), or \( m_j \in \alpha_S_j \), \( m_j \in L\alpha_{max}(J) \) each diagonal brick, \( [K_{jj}]_J, J \in \delta, j \in \zeta(J) \), is a symmetric matrix of dimension \( |Z(n_j)| \), or \( |Z(m_j)| \).
Figure 4.10. The brick matrix of the substructure $\alpha_{SL_7}$ and its super nodes
Figure 4.11. Brick matrix of the substructure $\alpha_{SL_7}$
iii) \( [K^\alpha_{ij}]_J \) is a sparse brick matrix, each brick, \( [K^\alpha_{ij}]_J \), represents an existing influence line between \( n_j \in S\), \( m_j \in L^{\text{max}}(J), J \in \delta, j \in \zeta(J). \)

3. The block stiffness matrix \( [K^\alpha_{IK}] \) which belongs to the front \( W^\alpha(I,K) \) is naturally divided into two submatrices, one of them is a null matrix, while the other one is a sparse brick matrix. Each brick, \( [K^\alpha_{IK}]_{IK} \), is a
\[
(1 \times 1) \text{ matrix and belongs to the cut } W_{ij}
\]
between \( m_i \in \alpha(S), n_k \in \alpha(K), I,K \in \delta, i \in \zeta(I), k \in \zeta(K). \) The number of these brick matrices is at most \( 2 \times |\alpha(K)| \) in \( [K^\alpha_{IK}] \) (Figure 4.10).

Each brick stiffness matrix for a node, \( n_i, n_k \in Z(S), \) a member, \( m_j \in D(S), \) a cut, \( w_{kj} \in W^\alpha(I), \) and an influence line \( w_{ij} \in W^\alpha_0(I), I \in \delta, \) will be stored and recalled as a Unit associated with an ordered address \( (I,i); (J,j), (K,k,I,j) \) and \( (I,i,I,j), \) respectively. All the basic matrix operations will treat the units as an entity which can be performed on an assembly language level, we will need to store the inverse of each of the brick matrices on the diagonal of the global stiffness matrix.

4.4.2. Load block vector

Each diagonal block in the block stiffness matrix \( [K^\alpha_{II}] \) is associated with a Load Block vector, \( F_I, \)
\[
F_I^T = (f_i^T, i = 1, \ldots, |F_I|), \text{ where } f_i \text{ is a load brick.}
\]
Each one of these load bricks is associated with a node $n_i$ and a member $m_i$, the order of the load bricks in the load block vector is identically the same as its corresponding node, i.e., the load brick, $f_i$, $f_i \in F_i$, will be identified by the ordered pair $(I, i)$, $I \in \delta$, $i \in \zeta(I)$.

### 4.4.3. The priority set, $Y^\alpha$

In the transition interval, the set $Y$ will change its elements into ordered pairs $Y^\alpha = ((I, i) / m_i \in aS_L, I \in \delta, i \in \zeta(I))$, $|Y^\alpha| = |D(S_L^\alpha)|$. $Y_{\theta_0}^\alpha = (Y_0^\alpha, \theta_0)$, $\theta_0 : \{1, 2, ..., |Y^\alpha| \} \rightarrow Y^\alpha$. Needless to say that the number of elements in the set $Y^\alpha$ will change with the load amplitude $\alpha$.

### 4.5. The Nonlinear Interval

For $\alpha \geq \alpha_{\text{max}}$, all the members $m_i$, $m_i \in S$, are nonlinear. And the structure, $S$, is one nonlinear continuum.

$$\text{MLS}^* = \text{MLS}(\alpha) = \text{L}S'(\alpha) = \text{L}S(\alpha) = \text{L}S_1'(\alpha) = \text{L}S_1(\alpha)$$

During this interval, the nodes and members of the structure $S$, will be kept in the same order they had after the last member changed to be nonlinear.

The priority set, $Y_\theta^\alpha = \phi$, the ordering, updating, and condensation procedures will stop and other feasible approximate methods must be used to solve the nonlinear LSS.
5. UPDATING AND REORDERING OF THE STRUCTURE GRAPH, 
\( G(\alpha) \)

Without loss of generality, we will assume that the load amplitude, \( \alpha \), is a monotone variable 
\( (\alpha_0 < \alpha_1 < \alpha_{\text{min}} < \alpha' < \alpha'' < \alpha_{\text{max}}), \alpha_0 = 0. \)

Given the multi-level structure of the structure \( S \) at the load amplitude \( \alpha' \), MLS (\( \alpha' \)), define MLS\(_i\)(\( \alpha' \)) to be a new multi-level structure that has developed due to the change of only one member \( m_i \) to be nonlinear changing the linear substructure \( 'S_L' \), to the model \( "S_L", S_L = (S_L \setminus m_i), m_i \in 'S_L', S \in S_L', I \in \delta', i \in \zeta'(I). \)

We can rearrange the node levels, \( LS'_i(I) \), substructure levels, \( LS_i(\alpha') \), and the structure levels, \( LS^*_i(\alpha') \), to form MLS\(_i\)(\( \alpha' \)). The updating process for all new nonlinear members at a load amplitude \( \alpha'' \) is a recursive process.

\[
\mathcal{R}_\alpha[\text{MLS}(\alpha')] = \{\text{MLS}_0(\alpha') = \text{MLS}(\alpha'), \text{MLS}_1(\alpha'), \ldots, \\
\text{MLS}(\alpha'') = \text{MLS}_k(\alpha') \}
\]

where \( k \) is the number of new nonlinear members at \( \alpha'' \).

Dividing the load amplitude range \( (\alpha_{\text{min}}, \alpha_{\text{max}}) \) into load steps, \( \Delta \alpha \), and repeating \( \mathcal{R}_\alpha[\text{MLS}(\alpha)] \) will develop not only the topological configuration of the linear and nonlinear substructures, but also the order of the nodes and members in the structure \( S \), at any load amplitude \( \alpha \).
5.1. ID (Identification) Process

The first step in updating $G(\alpha')$ is to identify the order of $m_i, n_1$ and $n_2$, $m_i \in S_{L}^i$, $n_1, n_2 \in \text{adj}(m_i)$.

5.1.1. Nonlinear member ID

During the nonlinearity check, if $m_j$ is nonlinear ($t_j = 1$), $m_j \in S_{L}^j$, $j \in \theta'$, the nonlinear member $m_j$ can be identified as $m_i \in S_{L}, I \in \delta', i \in \zeta'(I)$. Using the ordered priority set, $Y_0$, where $Y_j = (I, i)$.

5.1.2. Adjacent nodes ID

From the expanded graph $G_+^e(I)$, we can identify $n_1, n_2$, such that $n_1, n_2 \in \text{adj}(m_i)$, and find their order. The order of $n_1, n_2$ will lead to four distinct basic cases to be studied separately.

The linear valency of the nodes $V_L(n_1), V_L(n_2)$, will also be determined at this stage.

5.1.3. Substructure level ID

The substructure level $LS_L(\alpha'), LS(\alpha') \in LS(\alpha')$, will be identified such that, $S_L \in L_3^L(\alpha')$, then we can easily define $LS_{L}^i(\alpha')$, such that $S_L \in LS_{L}^i(\alpha')$.

5.2. The Basic Configurations

We can isolate and study four distinct basic cases for the previously defined $m_i, n_1, n_2$.

A. $m_i \in L_3^e_{\text{max}}(I)$; (initiation of a new nonlinear sub-
structure)
1. \( n_1, n_2 \in Z(S_L) \)

B. \( m_i \in LS'_{\max}(I) \); (fill in of a nonlinear substructure)
2. \( n_1 \in Z(S_L), n_2 \in \hat{N}'_K[n_2 \in P(S_{N^j})] \).

(Fill in through one transition node.)
3. \( n_1, n_2 \in \hat{N}'_K[n_1, n_2 \in P(S_{N^j})] \).

(Fill in through two transition nodes.)
4. \( n_1 \in \hat{N}'_K[n_1 \in P(S_{N^j})], \n_2 \in \hat{N}'_K[n_2 \in P(S_{N^j})] \)

(Connect of two nonlinear substructures.)
\( \hat{N}\)"_K1, \( \hat{N}\)"_K2 can be in one or two substructure levels.

And consider the following distinct variations for each case.
   a) \( V'_L(n_1) = 1, V'_L(n_2) = 1 \)
   b) \( V'_L(n_1) = 1, V'_L(n_2) > 1 \)
   c) \( V'_L(n_1) > 1, V'_L(n_2) = 1 \)
   d) \( V'_L(n_1) > 1, V'_L(n_2) > 1 \)

The last variation may lead to the two alternatives.
   i) "S_{L1} connected, \( D(\"S_{L1}\) = D(\"S_{L1}\)
   ii) "S_{L1} disconnected, \( D(\"S_{L1}\) = D(\"S_{L1}\) \cup D(\"S_{L12}\)

and each new substructure, "S_{L11}, "S_{L12}"s a strong substructure. Since "S_L can be arranged, at most, into two strong substructures.
5.2.1. **Initiation of a new nonlinear substructure**

The member $m_i$, will form a new unordered nonlinear substructure, $S_N$, and a new unordered super node, $N''$, where

$$D(S_N) = (m_i), \quad Z(S_{LI}) = Z(S_{LI} \setminus (n_1 \cup n_2)).$$

a) $V'_L(n_1) = V'_L(n_2) = 1$

$$S_N = S_{LI} = S = (m_i, n_1, n_2).$$

The structure was one linear member that changed to be nonlinear.

b,c) $V'_L(n_1) = 1, V'_L(n_2) > 1$

$$Z(S_N) = (n_1), \quad N = (n_2),$$

$$|S''_N| = |S'_N| + 1, \quad |N''| = |N'| + 1. \text{ Order change from } L_S^L+1(\alpha).$$

d) $V'_L(n_1) > 1, V'_L(n_2) > 1$

$$Z(S_N) = \emptyset.$$

i) $S_{LI}$ connected,

$$D(S_{LI}) = D(S_L),$$

$$|S''_L| = |S'_L|, \quad N'' = (n_1, n_2),$$

$$|N''| = |N'| + 1. \text{ Order change from } L_S^L+1(\alpha).$$

ii) $S_{LI}$ disconnected,

$$D(S_{LI}) = D(S_{LI1}) \cup D(S_{LI2}), \quad n_1 \in P(S_{LI1}),$$

$$n_2 \in P(S_{LI2}), \quad N'_1 = (n_1), \quad N'_2 = (n_2),$$

$$|S''_L| = |S'_L| + 1. \text{ Order change from } L_S^L-1(\alpha).$$
5.2.2. **Fill in through one transition node**

Update, \( D('S_{N_j}) = D('S_{N_j}) \cup m_i \).

a) \( V'_L(n_2) = 1, V'_L(n_1) = 1 \):

\[ 'S_{L_1} = (m_1), 'S_{L_1} \equiv N'_k \equiv \phi, \]

\[ | S''_L | = | S'_L | - 1, | N'' | = | N' | - 1, \]

\[ Z('S_{N_j}) = Z('S_{N_j}) \cup (n_1 \cup n_2). \]

Order change from \( LS_{L-1}(\alpha) \).

b) \( V'_L(n_2) = 1, V'_L(n_1) > 1 \):

\[ 'S_{L_1} \text{ is connected}, Z('S_{N_j}) = Z('S_{N_j}) \cup n_2, \]

\[ | N''_K | = | N'_K |, N''_K = (N'_K \setminus n_2) \cup n_1. \]

No reorder is necessary.

c) \( V'_L(n_2) > 1, V'_L(n_1) = 1 \):

\[ N''_K = N'_K, Z('S_{N_j}) = Z('S_{N_j}) \cup n_1. \]

No reorder is necessary.

d) \( V'_L(n_2) > 1, V'_L(n_1) > 1 \):

\[ N''_K = N'_K \cup n_1, Z('S_{N_j}) = Z('S_{N_j}). \]

i) \( 'S_{L_1} \text{ connected}, D('S_{L_1}) = D('S_{L_1}), \)

\[ | S''_L | = | S'_L |. \]

No reorder is necessary.

ii) \( 'S_{L_1} \text{ disconnected}, n_1 \in P('S_{L_1}), \)

\[ n_2 \in P('S_{L_2}), | S''_L | = | S'_L | + 1, \]

Order change from \( LS_{L-1}(\alpha) \).
5.2.3. **Fill in through two transition nodes**

Update, $D(S_{N_j}) = D(S_{N_j}) \cup m_i$.

a) $V'_{L}(n_1) = 1, V'_{L}(n_2) = 1$:

\[ S_{L_1} = (m_1), \quad S_{L_1} = N''_K \equiv \Phi, \]
\[ | S_L' | = | S_L | - 1, \quad | N'' | = | N' | - 1, \]
\[ Z(S_{N_j}) = Z(S_{N_j}) \cup (n_1 \cup n_2). \]

Order change from $L_{S_{L-1}}(\alpha)$.

b,c) $V'_{L}(n_1) = 1, V'_{L}(n_2) > 1$:

\[ S_{L_1} \text{ is connected}, \quad D(S_{L_1}) \equiv D(S_{L_1}), \]
\[ Z(S_{N_j}) = Z(S_{N_j}) \cup n_1, \quad N''_K = (N''_K \setminus n_1), \]
\[ | N'' | = | N' |. \]

No reorder is necessary.

d) $V'_{L}(n_1) > 1, V'_{L}(n_2) > 1$:

i) $S_{L_1}$ connected, $D(S_{L_1}) = D(S_{L_1})$.

\[ N''_K \equiv N''_{K'}, \quad | N'' | = | N' |, \quad | S''_L | = | S''_L |. \]

No reorder is necessary.

ii) $S_{L_1}$ disconnected,

\[ D(S_{L_1}) = D(S_{L_1}) \cup D(S_{L_2}), \]
\[ n_1 \in P(S_{L_1}), \quad n_2 \in P(S_{L_2}), \]
\[ | S''_L | = | S''_L | + 1. \]

Order change from $L_{S_{L-1}}(\alpha)$.

5.2.4. **Connection of two nonlinear substructures**

Update, $D(S_{N_j}) = D(S_{N_{j1}}) \cup D(S_{N_{j2}}) \cup m_i$.

\[ | S''_N | = | S''_N | - 1, \text{ order change from } L_{S_{L-2}}(\alpha). \]
5.3. Updating and Reordering of the Brick Graph, $G^a(I)$

5.3.1. Updating a super node

a) $\hat{N}''_K = (\hat{N}'_K \setminus n_1)$ remove the $n_1$ from the set $\hat{N}''_K$ and use the previous order, $n_k \in \hat{N}'_K$, $k \in \zeta(K)$ to subindex the nodes. If $\hat{N}'_K = (n_1)$, $(\hat{N}''_K \equiv \phi)$, the super node will be deleted.

b) $\hat{N}''_K = \hat{N}'_K \cup n_1$, $n_1$ will be ordered as the last element in the set $\hat{N}''_K$, i.e., the subindex $k$, $k = |\hat{N}'_K| + 1$, will be assigned to $n_1$. If $\hat{N}''_K = (n_1)$, $(\hat{N}'_K \equiv \phi)$, a new super node will be initiated.
5.3.2. Updating a nonlinear substructure

For $S_{Nj}, S_{Nj1}, S_{Nj2} \in S'_N$, we will have the following cases

a) Initiation of "S_{Nj}, $S_{Nj} \in S'_N$ [$D(S_{Nj}) = (m_i)$]

b) Fill in of 'S_{Nj} [$D(S_{Nj}) = D(S_{Nj}) \cup m_i$]

c) Connection of 'S_{Nj1}, 'S_{Nj2}, $D(S_{Nj}) = D(S_{Nj}) \cup D(S_{Nj}) \cup m_i$

The member, $m_i$, will be added to the expanded graph, $G'_+(J)$, connected, at least, to a node at the old starting level.

Establish the new starting level, $n_i \in P(S_{Nj})$, from Section 5.3.1, and use the same steps in Section 4.3.2 to order "S_{Nj}.

Update the sets $N'_K \in N'(J)$ by rearranging the set $P(S_{Nj})$, the last level in $G'_+(J)$, into its disjoint subsets $N'_I \in N'(J), N'_I = (n_K \in P(S_{Nj}) / n_K \in P(S_{L_I}))$.

5.3.3. Updating a linear substructure

$m_i \in 'S_{L_I}, 'S_{L_I} \in S'_L, D(S_{L_I}) = D(S_{L_I}) \setminus m_i$. 

a) If $'S_{L_I} = (m_i) \text{ i.e., } (S_{L_I} = \emptyset)$, delete "S_{L_I}.

b) If the case requires a connectivity check, use the SR algorithm to find the new sets "S_{L_I1}, "S_{L_I2}, $P(S_{L_I1}), P(S_{L_I2})$, and $G'_+(I1), G'_+(I2)$. Otherwise, reorder $G'_+(I)$ by taking out the member, $m_i$, from $L'_\text{max}(I)$, establish the new starting level and order $G''_+(I)$.
Update the sets, $N'_k \in N'(I)$, for "$S_{LI}^I$", and "$S_{LI}^I$" or "$S_{LI}^I$" by rearranging the set $P("S_{LI}^I")$, the last level in $\mathcal{G}'_+(I)$, into the disjoint sets, $N''_{Ij} \in N''(I)$,

$$N''_{Ij} = (n_k \in P("S_{LI}^I") / n_k \in P("S_{NJ}^I")).$$

5.4. Updating and Reordering of the Super Graph, $G^*(\alpha)$

During the change of the load amplitude, $\alpha$, the super graph, $G^*(\alpha)$, will go through five phases to complete the transition from a linear structure to a nonlinear structure.

5.4.1. The linear interval ($\alpha < \alpha_{\text{min}}$)

The structure, $S$, is one linear continuum we can apply the linear methods for the structural analysis, meanwhile we can update the priority set, $Y^Q$, when we perform the nonlinearity check. The structure is represented by a node (0) in the only level $LS^*_0(\alpha)$. In this interval, no updating for the order is required.

5.4.2. The transition begins ($\alpha = \alpha_{\text{min}}$)

At least one member will change to be nonlinear, and the first nonlinear substructure will be formed with its super nodes connecting it to one or two linear substructures. Forming the first and the second structure levels $LS^*_1(\alpha)$, $LS^*_2(\alpha)$ which can be viewed as the "top" of the structure levels "pyramid."
5.4.3. The transition interval \( \alpha_{\text{min}} < \alpha < \alpha_{\text{max}} \)

In this load amplitude interval, all the significant topological changes will happen to the linear (nonlinear) substructures. The interval can be analyzed as four overlapped loading periods.

At the first period, the initiation and growth of the nonlinear topological regions will develop the upper part of the pyramid-like structure levels, \( LS^*(\alpha) \). Its levels are nonlinear and linear substructure alternately.

At the second period, we will notice "splitting" phenomena, a linear region splitting into two small regions, increasing the width and depth of \( LS^*(\alpha) \). The pyramid structure reaches its maximum width and depth at the end of this period which is denoted by \( \alpha_{\text{opt}} \).

The optimum load amplitude, \( \alpha_{\text{opt}} \) is defined to be the maximum load amplitude which correspond to maximum number of structure levels and maximum number of substructures in the last level, \( | LS^*(\alpha_{\text{opt}}) | = | LS^*(\alpha) |_{\text{max}} \),

\[
| \alpha_{\text{opt}} |_{\text{max}} = | \alpha_{\text{opt}} |_{\text{max}}, \alpha_{\text{opt}} \in LS^*_{\text{max}}(\alpha_{\text{opt}}).
\]

At the third period, we will find "collision" phenomena, two nonlinear regions joining on their boundary to form one larger region, decreasing the width and depth of \( LS^*(\alpha) \) pyramid.

At the fourth period, the collapse of \( LS^*(\alpha) \) pyramid levels into fewer levels will take place. At the end of this
period, we will have only two levels, the linear level containing at most two linear substructures while the nonlinear level has only one nonlinear substructure. The super nodes number will increase during the first two periods, and decrease during the last two.

The "dynamics" of $G^*(\alpha)$ can be studied in view of five basic cases. Their set, the basic mechanisms, is denoted by $C_b = (C_1, C_2, C_3, C_4, C_5)$. Any change process is considered as a combination of one or more of these basic cases, taking into consideration that a change of one member, $m_i$, to be nonlinear can be viewed as a combination of at most two basic cases, and it should be one of the feasible operation set,

$$C_f = \{(C_1), (C_2), (C_3), (C_4), (C_5), (C_2C_4), (C_2C_5), (C_3C_4), (C_3C_5)\}$$

The DOC algorithm is most rewarding to use, in the nonlinear analysis of LSS, in the first and second period of the transition interval of the load amplitude, $\alpha$.

5.4.3.1. Steady state configuration, $(C_1)$

$D(S_{nj}) = D(S_{nj}) \cup m_i, m_i \in S_{Li}$. This case is accompanied only by a "fill-in" situation without any changes on the structure levels $LS^*(\alpha)$ or substructure levels $LS(\alpha)$. No reorder is necessary in this case.
5.4.3.2. **Initiation of a nonlinear substructure.**

\[(C_2) \quad D(S_{ij}) = (m_i), m_i \in \mathcal{S}_{L^i}, S_{L^i} \in LS^*_{i}(\alpha'), I,J \in \delta\]

A new nonlinear substruction, \(S_{ij}\), is formed at a higher structure level, \(LS^*_{i+1}(\alpha')\), and a new super node, \(N''_K\), is formed. If no other nonlinear substructures exist in this level, a new structure level \(LS^*_{i+1}(\alpha')\) is formed. Reorder from \(LS^*_{i+1}(\alpha')\) to get \(\beta'\).

5.4.3.3. **Connection of two nonlinear substructures.**

\[(C_3) \quad D(S_{ij}) = D(S_{nj_1}) \cup D(S_{nj_2}) \cup m_i, m_i \in \mathcal{S}_{L^i}, S_{L^i} \in LS^*_{i}(\alpha'), I,J_1,J_2 \in \delta'.\]

We will consider that one nonlinear substructure, \(S_{nj_2}\), has been deleted. The other one, \(S_{nj_1}\), will be the new nonlinear substructure, \(S_{nj}\), and its super nodes set, \(\mathbb{N}''(J) = \mathbb{N}'(J_1) \oplus \mathbb{N}'(J_2)\); \(\oplus\) is defined as follows:

"For every \(N''_K \in \mathbb{N}''(J)\), \(N''_K \equiv N''_{IJ}, N''_{IJ} = N''_{IJ_1} \cup N''_{IJ_2}, \) there exist \(N'_{IJ_1} \in \mathbb{N}(J_1)\), \(N'_{IJ_2} \in \mathbb{N}'(J_2)\), \(I \in \delta'\)."

a) For \(S_{nj_1}\), \(S_{nj_2} \in LS^*_{j}(\alpha')\), \((j = i + 1, \) or \(j = i - 1)\). \(S_{nj} \in LS^*_{j}(\alpha')\). Reorder from \(LS^*_{j}(\alpha')\) to get \(\beta'\).

b) \(S_{nj_1} \in LS^*_{i-1}(\alpha')\), \(S_{nj_2} \in LS^*_{i+1}(\alpha')\).

For all \(S_{L^i}, S_{L^i} \in LS^*_{i+2}(\alpha'), S_{L^i} \in \text{adj}('S_{nj_2}), I \in \delta, S_{L^i} \in LS^*_{i}(\alpha')\). This process is repeated for all the nonlinear substructures, which are connected to those linear substructures, to be shifted two structure levels, and so on.
If $LS_{i+1}(\alpha) = \emptyset$, delete this level. Reorder from $LS_i(\alpha')$, to get $\beta'$.

### 5.4.3.4. Elimination of a linear substructure.

(C4) \[ D\left(S_L^i\right) = \emptyset, \text{ i.e., } D\left(S_L^i\right) = (m_i), \text{ } m_i \in S_L^i, \]

The linear substructure, $S_L^i$, its super nodes, $S'k \in S'(I), I \in \delta'$, will be deleted. If $LS^*_i(\alpha') = (S_L^i)$, delete this level. Reorder from $LS^*_{i+1}(\alpha')$, if it exists, if not, no reorder is necessary.

### 5.4.3.5. Split of a linear substructure, (C5)

\[ D\left(S_L^i\right) \cup D\left(S_L^{i+1}\right) = (D\left(S_L^i\right) \setminus m_i), \text{ } S_L^i \in LS^*_i(\alpha'), \]

We will consider that a new linear substructure, $S_L^{i+1}$, is formed, the original one, $S_L^i$, will be denoted $S_L^{i+1}$, $I \in \delta'$, and a new super node set $S'(I)$ will be formed from the splitting of the old sets $S'(I)$, as follows:

i) **Split rule:**

"For every $S''_K \in S''(I), there exist $S''_{K1} \in S''(I1), S''_{K2} \in S''(I2), I, K \in \delta'."

ii) **Check rule:**

"For every $S''_{K1} \in S''(I1), S''_{K2} \in S''(I2), eliminate the empty sets, i.e., if $S''_{K1} = \emptyset$, then $S''_K$ will be deleted."

iii) **Move rule:**
"If there exists at least one super node, $N^*_K$, connecting $S_{Li}$ to $LS^*_i-1(a')$ , $S_{Li} \in LS^*_i(\alpha')$, then it remains in the same structure level, $S_{Li} \in LS^*_i(\alpha')$. Otherwise, it will 'move' to a two level higher structure level, $S_{Li} \in LS^*_{i+2}(\alpha')$.

If $LS^*_{i+2}(\alpha') = \emptyset$, $LS^*_{i+2}(\alpha") = \{S_{Li}\}$, a new structure level, $LS^*_{i+2}(\alpha")$, is formed. Reorder from $LS^*_{i+2}(\alpha")$ if one linear substructure will "move," otherwise, reorder from $LS^*_i(\alpha")$.

5.4.4. The transition ends ($\alpha = \alpha_{\text{max}}$)

The last two levels will collapse to form just one point. This indicates that the whole structure has completed the transition to be a nonlinear structure.

5.4.5. The nonlinear interval ($\alpha > \alpha_{\text{max}}$)

The structure, $S$, is one nonlinear structural continuum. Nonlinear structural methods must be applied for the analysis of the structure.

5.5. Updating and Reordering of the Block Graph, $\mathcal{G}(\alpha)$

This process will follow automatically any changes in the $i^{\text{th}}$ structure level $LS^*_i(\alpha')$ that need a reordering process, and it will influence all the substructures $S_{I} \in LS^*_j(\alpha')$, $j \geq i$, $I \in \delta$. 
5.5.1. **Updating the structure level matrix, \([C^\alpha]\)**

The changes in the super graph, \(G^*(\alpha)\), due to one or more processes of the feasible operations, are translated into matrix operations in the structure level matrix \([C']\).

a) **Elimination of a diagonal block, \('S_L^I, "S_N^I'\):**

This process is executed by elimination of the row and column of that block completely.

b) **Initiation of a diagonal block, \("S_N^I\):**

A new row and column will be added with a diagonal block and an off-diagonal set of blocks (super nodes), such that they properly represent the connectivity of \("S_N\).

c) **Splitting of a diagonal block, \('S_L^I \rightarrow \{"S_L^I_2, "S_L^I_1\}:**

A new row and column will be added as a copy of the ones that represent \('S_L^I\), and the check in Section 5.4.3.5 is performed to establish the existence of each off-diagonal block and rearrange the matrix.

d) **Connection of two diagonal blocks:**

The two rows and column will be added using the Boolean multiplication (1 + 1 = 1, 1 + 0 = 1, 0 + 0 = 0) to establish the existence of the new sets of super nodes.

e) **Reordering of the block matrix \([B']\):**

This process is equivalent to a symmetric permutation \([T][B'][T^T]\) where \([T]\) is an \((|S'| \times |S'|)\) permutation matrix.
5.5.2. Updating the set $\mathcal{N}_{\gamma}$

The procedure used to reorder the super nodes is the one used in Section 4.2, except that the reordering will only begin from $LS_i^*(\alpha')$, which means that it will influence only substructure levels beginning from $LS_j(\alpha')$, $j = 2(i - 1)$. The new substructure levels, $LS_j(\alpha'')$, and the new ordered set $\mathcal{N}_{\gamma''}$ will be the result of this procedure.

5.5.3. Updating of the block matrix $[B^\alpha]$ 

It is possible to establish the new block matrix $[B'']$ by following the same rules used in Section 4.2 beginning from $LS_j(\alpha'')$, $j = 2(i - 1)$.

5.6. Updating the Priority Set, $Y^\alpha_\theta$

The priority set, $Y^\prime_\theta$, will be updated by the elimination of the pair $(I,i)$ associated with, $m_i$, and updated the order for the remaining linear members. It is obvious that the updating process will end at the load amplitude $\alpha_{\text{max}}$. 
6. CONDENSATION OF LINEAR SUBSTRUCTURES

The second main purpose of the DOC algorithm is to condense out all the linear substructures during the transition interval, \( a_{S_I} \), \( I \in \mathcal{S} \), and update the Condensed Graph of the structure \( S \), \( G^c(\alpha) \), as \( \alpha \) changes.

The condensation process takes place only in the transition interval. For the linear and the nonlinear intervals, the structure graph, \( G^s(\alpha) \), and the block graph, \( G(\alpha) \), are both null graphs of one node, \( G^s(\alpha) \equiv G(\alpha) \equiv (S) \).

6.1. Condensation of a Brick Graph, \( G^a(I) \)

The condensation of the internal nodes and members, in a linear substructure \( a_{S_I} \), will be performed in two stages.

**Stage 1:** \( \{ n_i \in Z(a_{S_I}), m_i \in D(a_{S_I}), m_i \notin L^\alpha_{\max}(I) \} \)

The condensation of a node is executed according to its order inside the linear substructure, going from a lower level to a higher one, to decrease the number of fill-in edges (Figure 6.1).

**Elimination Graph** \( G^a_i(I) \) [26,27].

**Step 1:** Delete the \( i \)th node \( n_i(m_i) \) together with its incident edges.

**Step 2:** Add auxiliary (fill-in) edges such that all adjacent nodes of \( n_i \) form a clique, that is

\[
G^a_i(I) = (a_{S_I} \setminus n_i), \quad W^a_0(a_{S_I} \setminus n_i) \cup \text{def}(n_i).
\]
Figure 6.1. Condensation of a Brick Graph, $\overline{G}^a(I)$ (Stage 1)
$G^a_I(I)$ is a strongly connected graph. At the end of this stage, the only remaining nodes in the substructure will be the member nodes in the last level, $m_i \in LS^a_{\text{max}}(I)$.

**Stage 2:** $[m_i \in LS^a_{\text{max}}(I)]$ (Figure 6.2).

All the members in the last sublevel of the substructure will be condensed out through the expanded graph $G^a_I(I)$ to the peripheral nodes, $n_i \in P(S^a_{\text{IL}})$, which are in the surrounding super nodes, $N^a(I)$, i.e., the fill-in will happen only inside the super nodes $N^a(I)$, and the new fronts that will connect them.

This process will reduce the number of fill-in effectively, and decrease the CPU computer time to the minimum.

The process of elimination of a node, $n_i(m_i)$, from the graph, $G^a(I)$, is associated with a Gauss elimination for the brick matrix of this node, where the inverse of the brick matrix $[K^a_{ij}]_I$ will be used and also with the elimination of the load brick, $f_i$, and the update of the block load vector, $F_I$.

### 6.2. Condensation of the Block Graph, $\tilde{G}(a)$

#### 6.2.1. Condensation of a linear substructure, $^{aS_L}$

The process of condensation of a linear substructure $^{aS_L}_I$, $I \in \delta$, from the structure $S$ during the solution process is topologically represented by the elimination process of the
Figure 6.2. Condensation of a Brick Graph, $\tilde{G}(I)$ (Stage 2)
Ith node from the block graph, $G(a)$, to obtain the Elimination Graph $G_I(a)$ (Figure 6.3a).

**Step 1:** Delete the Ith node, $a_{SL_I}$, together with its incident edges.

**Step 2:** Add auxiliary (fill in) edges such that all adjacent super nodes, $N(I)$, form a clique, that is $G_I(a) = (\{a^* \setminus a_{SL_I}\}, \{\mathcal{W} \setminus a_{SL_I}\} \cup \text{def}(a_{SL_I}))$, $I \in \delta$, $G_I(a)$ is a strongly connected graph.

Each one of these elimination graphs, $G_I(a)$, is associated with a condensation of $G(a)$ and an update of the block load vectors, $F_J$, $J \neq I$, $J \in \delta$ (Figure 6.3b).

6.2.2. The condensed nonlinear graph, $G_C(a)$

The condensation of all linear substructures in the structure, $S$, at the load amplitude $\alpha$, is a recursive process defined by:

$G_C(a) = \{G_0(a) = G(a), G_2(a), \ldots, G_C(a) = G_{ix}(a)\}$ where $(ix)$ is the max index for a linear substructure.

The Condensed Nonlinear Graph $G_C(a)$ is the final result of this process, where its nodes are nonlinear substructures and super nodes placed in the nonlinear levels, LS$^C(a)$, which has a special level structure, shown in Figure 6.4a, characterized by one nonlinear substructure level followed by two super node levels with interconnected super nodes.
Figure 6.3a. Elimination graph $G_7(\alpha)$
Figure 6.3b. Block matrix of the elimination graph $G_7(\alpha)$
Figure 6.4a. Condensed nonlinear graph $G^c(\alpha)$
If \( G(\alpha) \) has \((2k-1)\) structure levels \((LS_1(\alpha), LS_2(\alpha), ..., LS_{2k-1}(\alpha))\) then, \( G^c(\alpha) \) will have \([(3k/2) - 1]\) nonlinear levels, for even \( k \), and \([(3k-5)/2]\) for odd \( k \), \((k \neq 1)\).

### 6.2.3. The condensed nonlinear stiffness matrix, \([F^a]\)

After the condensation of the block matrices that correspond to linear substructures, the remaining blocks will represent nonlinear substructures, passive fronts and supernodes. The change, will take place in the block matrices of the super nodes and a set of new formed off-diagonal blocks between them. These blocks will experience a fill-in of brick matrices that didn't exist before, and a change in the diagonal ones (Figure 6.4b).

The fill-in rule is simple. If there exists a path in the expanded graph of a condensed linear substructure, \( G^a(I), aS_{LI} \in S^a, I \in \delta \), between two peripheral nodes, \( n_i, n_j \in P(aS_{LI}) \), with the corresponding brick matrices \([K^a_{ii}], [K^a_{jj}]\), respectively. Then:

1. The diagonal brick matrices \([K^a_{ii}], [K^a_{jj}]\) will change.
2. A new off-diagonal brick matrix \([K^a_{ij}]\), will be formed.

These brick matrices are calculated using simple matrix partitioning rules.
Figure 6.4b. Block matrix of the condensed nonlinear graph
6.3. Reverse Condensation

After the solution of the nonlinear problem at the load amplitude $\alpha'$ (Section 2.3), the nonlinear solution vector, $U_N$, will be available. To proceed in developing the linear solution vector, $U_L$, we will propagate the solution of the DOF of the nodes, $n_k \in \mathcal{N}_K$, through the expanded graphs $\mathcal{G}_+^\alpha(I)$, $\mathcal{G}_{L_1}^\alpha \in \mathcal{S}_L$, $I \in \delta$, in a reverse order of the condensation process using their condensed stiffness matrices. This Reverse Condensation process is equivalent to a back substitution phase in our solution.

The following step in the analysis is to check the linear members for nonlinearity according to their order in the priority set. That will help to update the priority set and to find the new nonlinear members which will be added to the condensed graph, $G_C^\alpha$.

6.4. Updating the Condensed Graph, $G_C^\alpha$

Without loss of generality, we will assume that at the load amplitude $\alpha'$, only one new linear member, $m_i \in \mathcal{S}_L$, $I \in \delta$, has changed to be nonlinear, and its boundary nodes are $n_j, n_k \in \text{adj}(m_i)$. The position of $m_i$, $n_j$ and $n_k$ will lead to these cases:

1. $m_i \in \mathcal{L}_{\text{max}}^\alpha(I)$, $n_j \in \mathcal{P}(\mathcal{S}_{N_j})$, $n_k \in \mathcal{P}(\mathcal{S}_{N_k})$; $I, J, K \in \delta$.

Only one brick row and column, that correspond to $m_i$, will be added in its new order, with a diagonal brick matrix.
and the off-diagonal brick matrices \([K'_{ij}], [K'_{ik}]\) that correspond to two influence lines between \((m_i, n_j)\) and \((m_i, n_k)\), respectively. The other changes will take place in \([K'_{jj}], [K'_{jk}], [K'_{kk}]\)

\[
[K'_{jj}] = [K'_{jj}] + [K'_{ji}] [K'_{ii}]^{-1} [K'_{ij}]
\]

\[
[K'_{jk}] = [K'_{jk}] + [K'_{ji}] [K'_{ii}]^{-1} [K'_{ik}]
\]

\[
[K'_{kk}] = [K'_{kk}] + [K'_{ki}] [K'_{ii}]^{-1} [K'_{ik}]
\]

(6.4.1)

2. \(m_i \in L\bar{S}'_{\text{max}}(I), n_j \in P(S_{N_j}), n_k \notin P(S_{N_k}); I, J, K \in \delta\). Two brick rows and columns, that correspond to \(m_i\) and \(n_k\), will be added in their new order. To update the condensation process in this case, delete the condensation process of the last level in \(\bar{G}'(I)\), \(L\bar{S}'_{\text{max}}(I)\), and repeat the condensation for that level without \(m_i\) or \(n_k\).

3. \(m_i \notin L\bar{S}'_{\text{max}}(I), m_i \in L\bar{S}'_i(I)\).

Three brick rows and columns, that correspond to \(m_i\), \(n_j\) and \(n_k\), will be added in their new order. To update the condensation process, delete the process up to the level \(L\bar{S}'_{i-1}(I)\) and repeat the condensation from this level without \(m_i\) or its boundary nodes \(n_j, n_k\).

A perturbation analysis will follow to expand the nonlinear solution to include the nonlinearity of the new nonlinear member, \(m_i\), and increase the dimension of the solution space by the new nonlinear DOF.
Finally, the updated condensed nonlinear stiffness matrix, \([K''\],\) and the updated condensed load vector, \(F''\), will be available for a nonlinear structural analysis phase.
7. CONCLUSION AND FUTURE PROJECTIONS

This research introduced a graph theoretic approach to the nonlinear analysis of large space structures (LSS). This approach is applicable during the transition of LSS models from linear to nonlinear. Using (SR) and (MR) algorithms, developed within this research, the members of the LSS can be arranged into linear and nonlinear substructures connected at their boundaries. The LSS model is first represented by a multi-level graph. Then, during the loading process the graph changes its structured levels with changes in the configuration of the nonlinear regions. This gives a unique representation of the transition of a linear LSS into a nonlinear one.

The new DOC (dynamic ordering and condensation) algorithm is used to order the nodes and members of the LSS and update this order with any change in the nonlinear regions. In addition, the DOC algorithm condenses all the linear degrees of freedom of the LSS model to reduce the dimension of the nonlinear stiffness matrix. The condensed stiffness equation is updated during the transition loading interval to increase its dimension as more members become nonlinear. The number of fill-ins in the condensed stiffness matrix is controlled by the dynamic ordering aspect of the DOC algorithm.
The DOC algorithm can also be extended to account for plastic hinge formation and fracture in members. The plastic hinge can be represented by a new boundary node that split the member into two connected members. Whereas, the fracture of any member can be represented by two coinciding boundary nodes which split the member into two disconnected members.
8. BIBLIOGRAPHY


9. ACKNOWLEDGMENTS

The author thanks Dr. T. J. McDaniel for his encouragement, motivation, and support throughout the course of this study. The author expresses his gratitude to Dr. A. M. Fink for his inspiration and many creative suggestions. Special thanks are extended to the author's wife, Hala Galal, whose support and patience have been invaluable. Thanks are also due to the author's parents and in-laws for their moral support.