Image Cover Sheet

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/92/462

MODIFICATIONS TO THE FINITE ELEMENT MODELLER PROGRAM "REFINE"

by
M.F. Palmeter - M.W. Chernuka

MARTEC Limited
1888 Brunswick Street, Suite 400
Halifax, Nova Scotia, Canada
B3J 3J8

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canadä

PLEASE DIRECT ENQUIRIES TO:

THE CHIEF,
DEFENCE RESEARCH ESTABLISHMENT ATLANTIC,
PO BOX 1012,
DARTMOUTH, NOVA SCOTIA,
CANADA.

UNLIMITED DISTRIBUTION

National Defence
Research and Development Branch

Défense nationale
Bureau de recherche et développement

DREA CR/92/462

MODIFICATIONS TO THE FINITE ELEMENT MODELLER PROGRAM "REFINE"

by
M.F. Palmeter - M.W. Chernuka

MARTEC Limited
1888 Brunswick Street, Suite 400
Halifax, Nova Scotia, Canada
B3J 3J8

Scientific Authority
Neil Pegg for D.R. Smith

W7707-6-7749/01-OAE
Contract Number

August 1990

CONTRACTOR REPORT

Prepared for

Defence Research Establishment Atlantic

Centre de Recherches pour la Défense Atlantique

DEFENSIO PER SCIENTIAM

Canada

ABSTRACT

Program REFINE was initially developed as part of an adaptive finite element modelling package for VAST where from estimates of discretization errors based on element residuals, the finite element model would be selectively refined (by element subdivision) until acceptable levels of accuracy (according to computed error estimates) were attained [1]. However, increasingly the REFINE program finds application as a general modelling tool which complements the data generation capabilities of PVAST and HVAST. Under these circumstances, the PVAST and HVAST programs need only concentrate on achieving accurate geometric models, since subsequent refinement of the model to attain acceptable finite element accuracy can be achieved with REFINE. This is particularly useful when the initial model is defined in a pointwise fashion with a graphics table. In this context as well, selective model refinement would generally be employed but the experience of the modeller would dictate where the model refinements would be.

RÉSUMÉ

Le logiciel REFINE a été conçu dans le cadre d'un progiciel adaptatif de modélisation aux éléments finis pour VAST. Le programme raffine sélectivement (par subdivision des éléments) le modèle aux éléments finis à partir d'estimations des erreurs de discrétization basées sur les résidus des éléments, jusqu'à ce que le niveau de précision indiqué par les estimations d'erreur calculées soit atteint [1]. Toutefois, les applications du programme REFINE se multiplient en tant qu'outil général de données de PVAST et de HVAST. Dans ces circonstances, les programmes PVAST et HVAST peuvent n'être utilisés que pour produire des modèles géométriques précis, REFINE permettant de raffiner ultérieurement le modèle pour obtenir des éléments finis de précision acceptable. Cela est particulièrement utile lorsque le modèle initial est défini point par point à l'aide d'une tablette graphique. Dans ce contexte également, le modélisateur aurait recours au raffinement sélectif du modèle, mais son experience lui dicterait où apporter les raffinements.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

CHAPTER 1
INTRODUCTION

1.1 Introduction

Program REFINE was initially developed as part of an adaptive finite element modelling package for VAST where from estimates of discretization errors based on element residuals, the finite element model would be selectively refined (by element subdivision) until acceptable levels of accuracy (according to computed error estimates) was attained [1]. However, increasingly the REFINE program finds application as a general modelling tool which complements the data generation capabilities of PVAST and HVAST. Under these circumstances, the PVAST and HVAST programs need only concentrate on achieving accurate geometric models, since subsequent refinement of the model to attain acceptable finite element accuracy can be achieved with REFINE. This is particularly useful when the initial model is defined in a pointwise fashion with a graphics tablet. In this context as well, selective model refinement would generally be employed but the experience of the modeller would dictate where the model refinements would be.

This report describes recent work completed to extend the capabilities of the program REFINE. Chapter 2 describes the program modifications which enable the refinement of additional element types. Almost all element types are operational now. Chapter 3 describes the introduction of appropriate cubic multi-point constraint equations for refinement of stiffened or unstiffened panels. Chapter 4 summarizes deficiencies in refinement of substructured models and discusses how one of them was addressed by automatically defining newly generated boundary nodes (on superelement boundaries) as either slave nodes or master nodes in a manner that preserves maximum structural flexibility. Chapter 5 describes a restart capability for REFINE suitable for either substructured or unsubstructured finite element models. Chapter 6 outlines other improvements in the superelement refinement algorithm. Chapter 7 describes several miscellaneous changes to the REFINE program not called for in the contract statement of work but which were determined to be desirable in recent applications of the program.

CHAPTER 2
REFINEMENT OF ADDITIONAL ELEMENT TYPES

2.1 Introduction

The element refinement algorithm of the program REFINE has been extended to several new elements from the VAST element library including the curved beam element (IEC=7), the axisymmetric shell (IEC=19) and the second degenerate form of the brick, namely the square based pyramid (IEC=2). In general, element refinement involves the generation of new nodes, the removal of the connectivity of the parent element to be refined, the generation of connectivities for new elements produced by the refinement process and also the generation and manipulation of constraint equations for irregular nodes on the interface between refined elements and unrefined elements. The program modifications for each of the above mentioned element types will be discussed separately in the following sections.

2.2 Square Based Pyramid (IEC=2)

The square based pyramid is a special case of the solid element and is defined by appropriate repetition of node numbers in the element connectivity list (see Figure 2.1).

The generation of new nodes for the square based pyramid followed an approach similar to that taken for the refinement of the tetrahedron elements and posed no great difficulties. The generation and manipulation of the constraint equations also did not create any unusual problems. The generation of the connectivities produced by the element refinement however, is a complex procedure. The philosopy of the REFINE program has been that the refinement of an element normally results in the generation of only smaller elements of the same type. The one exception to this is the refinement of the transition element which results in the creation of transition and shell elements as will be discussed in the following section. The refinement of the square based pyramid is also an exception to this rule since the refinement results in the formation of square based pyramids and tetrahedron elements. Therefore, a new element group is necessarily produced. The treatment of this new group is similar to the new group formed by refinement of the transition element. The refinement of the square based pyramid poses an additional problem since the total number of elements produced does not follow the general equation:

$$NELMF = NOR^{IDIEC} \tag{2.1}$$

where

NELMF = total number of elements produced by refinement
NOR = order of refinement
IDIEC = spatial dimension of the element type (1,2 or 3)

In this case, IDIEC is 3 since the square based pyramid is a three-dimensional element. The number of pyramid elements produced is given by the following equation:

$$NELMP = \frac{2}{3} * [NOR^{IDIEC} + \frac{NOR}{2}] \tag{2.2}$$

where

NELMP = total number of pyramid elements produced by refinement
NOR = order of refinement
IDIEC = 3

The number of tetrahedron elements produced is given by the following equation:

$$NELMT = \frac{2}{3} \; [NOR^{IDIEC} - NOR] \tag{2.3}$$

where

NELMT = total number of tetrahedron elements produced by refinement

NOR = order of refinement

IDIEC = 3

The refinement of the square based pyramid does produce an equivalent number of elements to Equation 2.1 if half the number of tetrahedrons given by Equation 2.3 are added to the total number of pyramids given by Equation 2.2. This manipulation is somewhat justified with the observation that each pyramid may be broken into 2 component tetrahedrons. Table 1 summarizes element information for several different orders of refinement. It can be readily seen that combining Equation 2.2 with half of Equation 2.3 always yields Equation 2.1.

Now that the number and type of elements produced have been established, the next consideration is the orientation of the elements. The orientation of elements is important when generating the load data for the refined model. The new pyramid elements are therefore generated with their face orientation the same as the parent element. The new tetrahedron elements produced on each of the faces have consistently one orientation. The logic required in the program to generate the element connectivities is quite complex and being sufficiently unique, it forms a separate section in subroutine REFINE. Load refinement for square based pyramids has not been performed under this contract.

2.3 Transition Element (IEC=6)

Refinement of the transition element presents some unique difficulties and some further development is still required. Presently, a transition element refinement of degree n generates n transition elements and (n-1)xn shell elements. The total number of elements produced is governed by Equation 2.1. As illustrated in Figure 2.2 for a refinement degree of 2, the face adjoining a solid element is divided into n elements with no element divisions through the element thickness. Element refinements of the adjacent solid element with refinement degree n, on the otherhand, results in n element divisions through the thickness and thus a refinement inconsistency on the common interface. In fact, depending whether the transition or solid element is refined first, difficulties are sometimes experienced in generating correct multi-point constraint equations on the transition-to-solid element interface.

Although a coding error was previously suspected, as the cause of the difficulty in generating correct MPC equations, this has now been ruled out. After considerable study, it became apparent that for reliable MPC equations the MPC on the transition-to-solid element interface modification of the MPC equation generation algorithm for the solid element was unavoidable. Different MPC equations would be generated on the faces of a solid element depending on whether a transition element or another solid element attached to it. In order to achieve this requires considerable changes to the databases on which REFINE operates. Information related to the neighbours of each element in the model would have to be generated and manipulated during refinement of a model. It is not difficult to appreciate the difficulty of the task especially when one contemplates the refinement of a model which has been refined previously and includes MPC equations on the interface between refined and unrefined sections. The philosopy of the REFINE program has been that each element selected by the user can be refined and appropriate MPC equations generated without regard to its neighbours.

An alternative refinement strategy involving the division of a transition element into smaller transition elements and wedge elements as illustrated in Figure 2.3 has therefore been considered. Although this has the advantages that any given interface is always divided consistently and that the solid element zone moves slightly into the shell structure, some deterioration of element proportioning develops if extensive refinements in the original transition element are required. However, if the maximum refinement order is no more than five or six and the maximum aspect ratio (element length to thickness) is no greater than ten, the aspect ratio will be acceptable.

The work performed under this contract consisted of a review of the existing coding to determine the most appropriate treatment of the transition element. It can be concluded that refinement of transition elements cannot be provided within the framework of REFINE without compromising the assumptions on which the program presently operates. The first refinement strategy involving changes to the program data structures represents a major undertaking and should not be adopted unless adequate manpower can be budgeted.

2.4 Curved Beam Element (IEC=7) and Axisymmetric Shell (IEC=19)

The curved beam element (superparametric beam element) (IEC=7) is suitable as a stiffener for the thick/thin shell element (IEC=1) and if used as such, the displacement node numbers of the shell to which the beam is connected are to be identified. These nodes must be identified in the REFINE program since any constraint equations generated must relate to the displacement nodes and not to the geometric nodes. The same procedure for identifying the displacement nodes is also necessary for the axisymmetric shell element (IEC=19) since the constraint equations relate to those nodes. The refinement of the curved beam elements also required the development of a new subroutine CBEAM to calculate the curved beam properties at new nodes using the shape function of each new nodal point in question.

TABLE 2.1

Number of Elements Produced by
Refinement of a Square Based Pyramid

NOR Order of Refinement	NELMF General IDIEC=3 (Equation 2.1)	NELMP Pyramid Elements (Equation 2.2)	NELMT Tetrahedron Elements (Equation 2.3)
1	1	1	0
2	8	6	4
3	27	19	16
4	64	44	40

```
CUBE MODELLED BY SQUARE BASED PYRAMID ELEMENTS
   0
  27    27     1
   1    10.0000      0.0000     10.0000
   2    10.0000      0.0000      5.0000
   3    10.0000      0.0000      0.0000
   4    10.0000      5.0000     10.0000
   5    10.0000      5.0000      5.0000
   6    10.0000      5.0000      0.0000
   7    10.0000     10.0000     10.0000
   8    10.0000     10.0000      5.0000
   9    10.0000     10.0000      0.0000
  10     5.0000      0.0000     10.0000
  11     5.0000      0.0000      5.0000
  12     5.0000      0.0000      0.0000
  13     5.0000      5.0000     10.0000
  14     5.0000      5.0000      5.0000
  15     5.0000      5.0000      0.0000
  16     5.0000     10.0000     10.0000
  17     5.0000     10.0000      5.0000
  18     5.0000     10.0000      0.0000
  19     0.0000      0.0000     10.0000
  20     0.0000      0.0000      5.0000
  21     0.0000      0.0000      0.0000
  22     0.0000      5.0000     10.0000
  23     0.0000      5.0000      5.0000
  24     0.0000      5.0000      0.0000
  25     0.0000     10.0000     10.0000
  26     0.0000     10.0000      5.0000
  27     0.0000     10.0000      0.0000
    2     3
 0.300E+08 0.300E+00 0.737E-03
   9   19   19    3   14   19   11    6    7   19   19    1   13   19   10    4    8   19   19    2
  27   19   19   21   23   19   20   24    9   19   19    3   14   19   11    6   18   19   19   12
  25   19   19   27   22   19   23   26    7   19   19    9   13   19   14    8   16   19   19   18
```

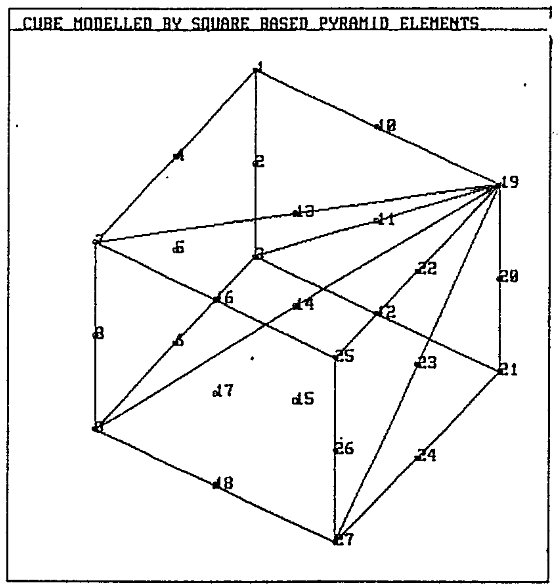


FIGURE 2.1: Typical GOM File for VAST With Square-Based Pyramid Elements.

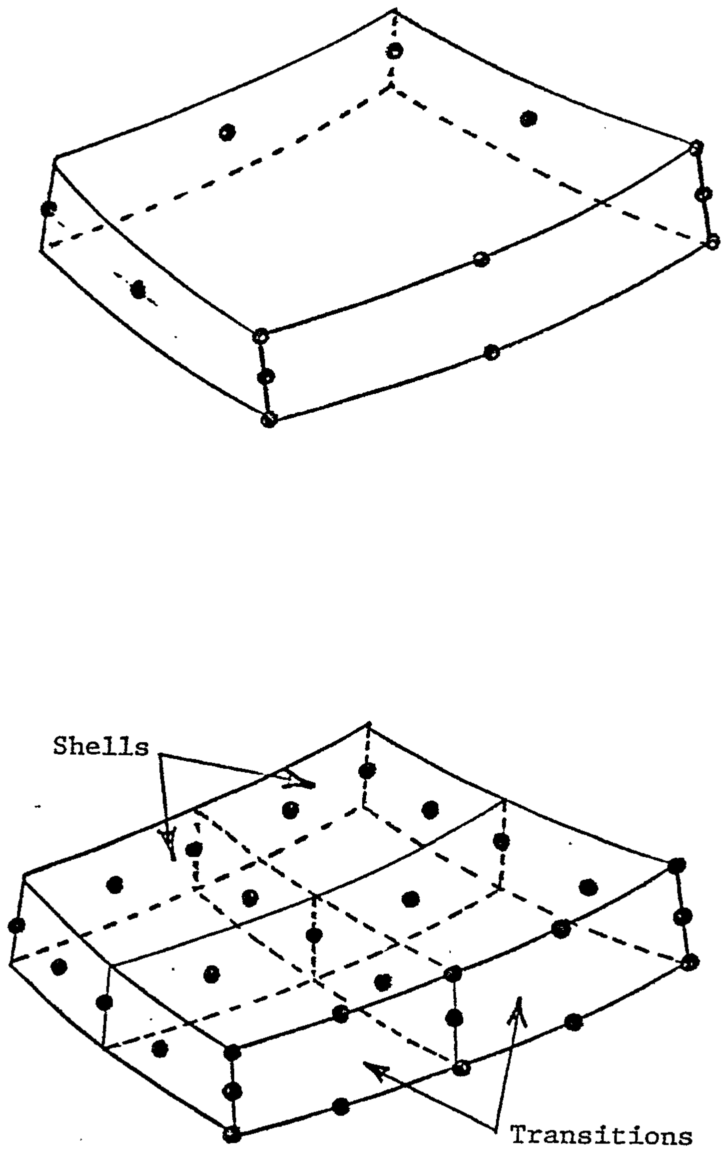


FIGURE 2.2: Transition Element Refinement Strategy as Originally Implemented in REFINE.

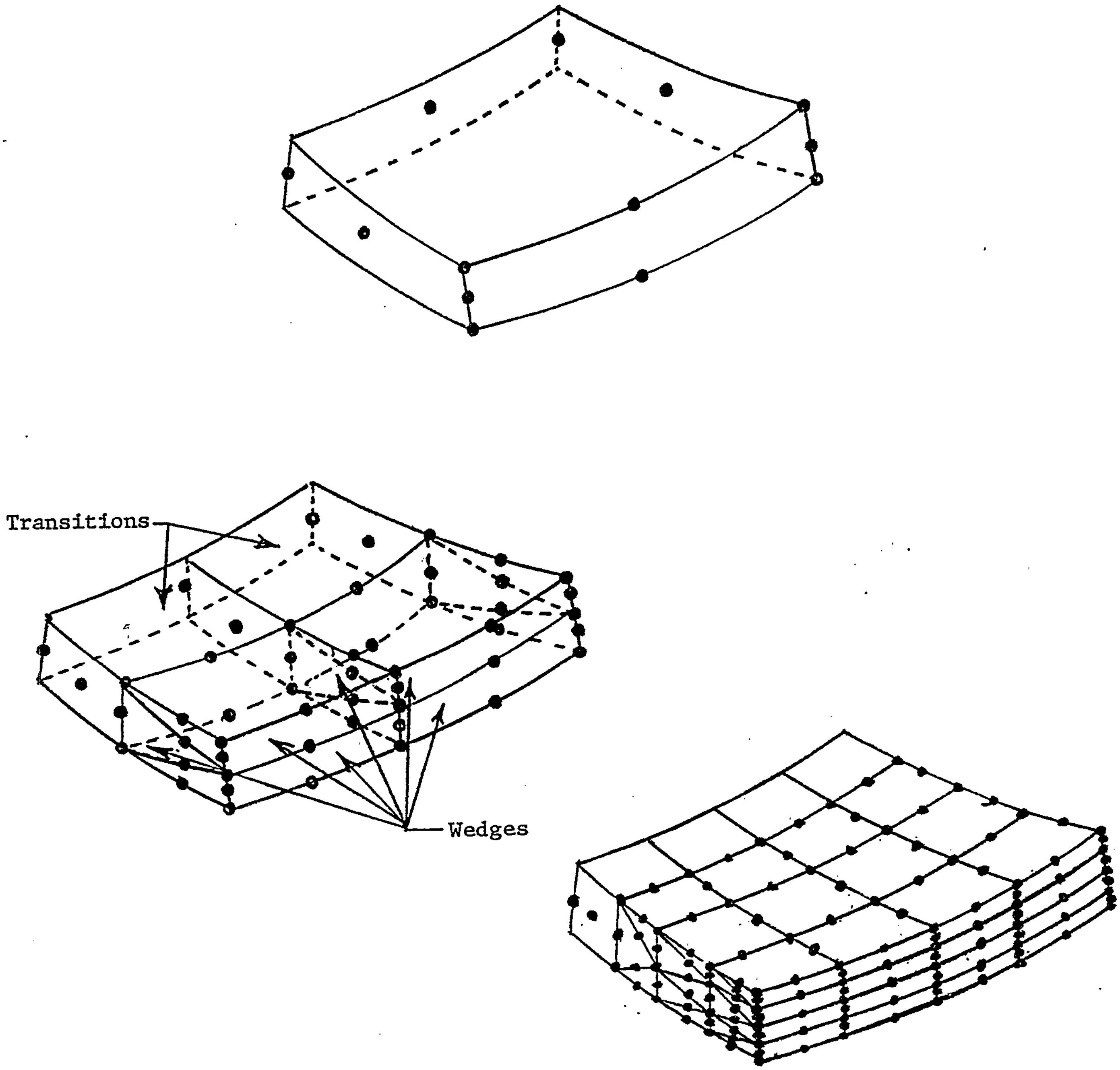


FIGURE 2.3: Alternate Transition Element Refinement Strategy Producing Wedge Elements.

CHAPTER 3
CUBIC MULTIPOINT CONSTRAINT EQUATIONS FOR REFINEMENT OF STIFFENED OR UNSTIFFENED PANELS BY PROGRAM REFINE

3.1 Introduction

If a finite element model is refined nonuniformly, more nodes may be generated on one side of some element interfaces within the original model than on the other side. Nodes not joined to nodes in contiguous elements across such an interface are referred to as irregular nodes. The presence of irregular nodes on an element interface implies lack of displacement compatibility unless displacements at all irregular nodes are related to displacements at regular nodes on such an interface.

Constraint equations are typically of the form:

$$u_{dm} = \sum_{n=1}^{n_t} c_{mn} u_{in}$$

where u_{dm} is the n-th dependent freedom (either translational displacement or rotation component), u_{in} are the independent freedoms on which they depend, and c_{mn} are coefficients relating them. For almost all elements in the VAST library, degree of freedom i at the dependent node m is related to only degree of freedom i at two or more independent nodes n. What this means for a refinement program such as REFINE is that constraint equation coefficients need be stored for only the first degree-of-freedom at each dependent node. At the end of the program REFINE when multi-point constraint data is being written to the USE file for the refined model, constraint equations for all degrees of freedom at dependent nodes can be created by simple repetition. The plate bending element and the general beam stiffener element however are exceptions. Out-of-plane displacements for points along an edge are cubic functions in a linear edge variable (defined by values of out-of-plane displacement and rotations at end points). The present note outlines the required modifications to generalize constraint generation algorithms within program REFINE.

A limited capability for the generation of cubic multi-point constraint equations for the triangular plate bending element and the general beam stiffener elements has now been implemented and tested in program REFINE. It permits nonuniform refinements of stiffened or unstiffened plate panels. The panel must be flat and its normal aligned with one of the coordinate axes. The user is prompted for the orientation of the normal. Although it is relatively straightforward to compute the direction of the normal for a stiffened panel, the program nonetheless prompts the user to manually define its direction cosines. However, it is felt that in facetted plate structures, a single normal may have to be utilized for correct generation and manipulations of the multi-point constraint equations.

What distinguishes constraint equation generation for the plate bending and general beam stiffener element types from others in the VAST library is that typically a different constraint equation is required on each freedom at irregular nodes of a nonuniformly refined model. Development of this capability thus involved two principal steps. The first step consisted of the generalization of procedures with REFINE so that more than one constraint equation could be generated and manipulated for each irregular (constrained) node. The second step, of course, was to implement the appropriate cubic constraint equations required for out-of-plane displacements in the plate bending elements. Similar approach is necessary for the general beam stiffener element.

In the following sections of this note these steps are elaborated on and some program usage restriction and guidelines are also discussed. The future development of this work is also discussed briefly.

3.2 Generalization of Constraint Equation Manipulation

The basic logic implemented within program REFINE for generation, manipulation, and optional elimination of constraint equations (as well as for computation of new nodal coordinates, and for generation of connectivities for new elements) is adequate for processing of cubic constraint equations. The same nodes are related in each constraint equation associated with a given irregular node; different degrees of freedom are generally involved in each constraint equation and frequently more than one degree of freedom is used at every dependent node. Dependent (irregular) nodes for which displacements are defined via constraint equations are stored in array NK and associated independent nodes in the matrix NJ. Enough information can be extracted from the structure of the first constraint equation at every dependent node for the purpose of filling arrays NK and NJ.

It is primarily in the generation and handling of the coefficients for the constraint equation that implementation of cubic constraint equations for the plate bending elements and general beam stiffener element differs from the implementation of appropriate constraint equations for other elements in the VAST library. In this section, the question of modifications to the constraint equation handling is discussed.

The simplest method of extending the program logic to handle multiple constraint equations where conceivably a different constraint equation is associated with each degree of freedom at a dependent node, is to leave the basic program logic related to determining the status of nodes (regular or irregular), to keeping track of all independent nodes associated with each dependent one etc. intact and setup an auxilliary mechanism (data management system) for storage and retrieval of constraint equation coefficient data. Whenever constraint equation manipulations are required, instead of one set of equations being operated on several sets will be. Generally the number of constraint equations to be operated on will be equal to the number of nodal degrees of freedom at each irregular node. Of course, repeating constraint equation computations when all constraint equations associated with an irregular node are equivalent is unnecessary and future refinements of this algorithm could conceivably involve investigating the possibility of performing one set of constraint equation calculations when they are equivalent for each degree of freedom at irregular nodes. For such nodes of course, it will be necessary to expand out the constraint equations and generate the NDF constraint equations from the single set of constraint equation coefficients generated by the program.

Due to efforts to improve speed of program REFINE during an earlier contract [1], dependent node numbers and associated constraint equation coefficients were stored in program arrays to reduce the amount of program I/O and thereby increase computational efficiency of critical sections of the refinement algorithm. To store coefficients for all constraint equations at each irregular (dependent) node will raise program storage requirements quite dramatically particularly since the maximum number of structural nodes has recently been increased to 2500.

CHAPTER 4
AUTOMATIC TREATMENT OF BOUNDARY NODES IN SUPERELEMENT REFINEMENT

4.1 Introduction

Substructuring plays a major role in finite element modelling of ship structures, and any finite element model refinement capability would be incomplete if it could not handle substructured finite element models. Earlier versions of the program REFINE were not fully operational on superelements [1]. A number of restrictions were adopted to facilitate producing a preliminary capability to refine superelements which included:

a. Only level one superelements can be refined.

b. A substructure may only define one superelement.

c. Substructures had to be refined in ascending order.

d. All new superelement interface nodes had to become either master nodes or slave nodes as defined by an interactive prompt.

e. Limitations on refinements of previously refined structures are permitted, notably, each element of the original model could be refined only once.

Considerable effort in this contract has been concentrated on removing the above mentioned restrictions. This chapter describes work on Item d.

4.2 Importance of Automatic Classification of Boundary Nodes

Geometrically, refinement of substructured models is no different than refinement of unsubstructured models. Basically element refinement still involves the generation of new nodes, removal of the connectivity for the parent element, the generation of connectivities for new elements produced by element refinement as well as generation and manipulations of the constraint equations for irregular nodes on the interface between refined and unrefined elements.

Some auxiliary issues related to substructures alone cannot be avoided when they are refined. Firstly, constraint equations are implemented by the slave node concept whereby dependent nodes for constraint equations are master nodes. Consequently, nodes internal to substructures may have to become master nodes in order that constraint equations can be applied. Secondly, assignment of constraint equations for boundary nodes in one substructure (superelement) cannot be done independently of other substructures of the model. If all substructures sharing a given interface are refined consistently all new nodes generated on the interface can become master nodes. Otherwise, those new nodes which do not have counterparts in all connecting substructures must be slaved (have constraint equations retained for them). A limited capability to refine substructure/superelements was implemented in the previous contract [1]. The approach adopted was to prompt the user whether all boundary nodes were to be identified slave or master nodes. The user was thus responsible for ensuring that the refinement of all substructures/ superelements were performed appropriately for the selection made. Figure 4.1 illustrates the two modelling options for new superelement interface nodes.

The restrictions adopted for the initial substructure/superelement refinement capability developed partly from time and budgetary constraints. At the time, it also seemed that the implementation of a general approach for automatically identifying a boundary node to be a master node or slave node would require searching all element connectivities of all superelements within the structure. The large amounts of file searching implied significant increases in CPU requirements and decreased algorithm (program) speed when substructuring was involved. In fact, recent developments in subroutine MPC lead to the proposal that the approach taken to determine when a constrained node can be eliminated could also be used to determine when a constrained boundary node can be eliminated. If the constraints on a boundary node remained, the node would be identified as a slave node, and if the constraints are removed, the node would be identified as a master node.

The main procedures employed in subroutine MPC to determine whether constraints on nodes could be eliminated required a search through all the element connectivities. The extension of this procedure to a substructured model would thus require a search through all the element connectivities of all the superelements. At the start of this contract, a review of the data requirements for determining whether a boundary node should be specified as a slave or master node revealed that sufficient data for a reliable decision could, in fact, be obtained from the superelement data section of the input file. This data provides the master node list for each superelement and thus defines the connectivity of each superelement. A search through this data would eliminate the need for searching through all the individual elements in the structure. The implementation of this procedure within a new subroutine called MPCSE is detailed in this chapter. The overall structure of the REFINE program does not change beyond the addition of a new module as is readily appreciated by examining Figure 4.2.

The capability to refine substructure/superelements has been improved in this contract to automatically identify the boundary nodes as slave or master. A boundary node is defined as a node located on the perimeter of a superelement.

This chapter will review the steps taken to identify nodes requiring multipoint constraint data. Constraint equations for substructures are implemented by using the constrained substructure option of VAST by defining independent nodes as master nodes and dependent nodes as slave nodes. The data for constrained node is manipulated on scratch files using the stiffness modification format and is re-formatted to the super-element format only in the final step. From the point of view of geometry an individual substructure is the same as an unsubstructured model. The treatment of the multipoint constraint data as outlined above allows the REFINE program to operate basically the same way for both a substructured and an unsubstructured model. The only difference is that for a substructured model the constraint equations are again reviewed after all the substructures have been refined.

4.3 Review of Important Terminology

The following definitions will be useful when discussing constraint equation generation and manipulation. The definitions assigned will be used throughout this chapter.

Border Node	A node which is located on the perimeter of a parent element.
Boundary Node	A node which is located in the perimeter of a superelement.
Dependent Node	A node requiring a multipoint constraint equation.
Interface Node	A border node common to two or more elements.
Internal Node	A node which is not located in the perimeter of a superelement.

4.4 Identification of Node Requiring Multipoint Constraint Equations

Multipoint constraint equations are required for irregular nodes created by non-uniform mesh refinement. The REFINE program generates the constraint equations for all "border" nodes created as the result of an element refinement and then may remove the constraint through a series of checks performed at three stages before the generation of the final output file. The constrained node is investigated to ensure the constraint is required by subroutine REFINE, subroutine MPC, and if the model is substructured, subroutine MPCSE. The REFINE subroutine is called as each individual element is refined. Subroutine MPC is called after all the elements in a substructure have been refined and MPCSE is called after all the substructures have been refined. The checks performed by each of the subroutines are described in detail in the following sections.

4.5 Constraint Equation Operations in Subroutine REFINE

Subroutine REFINE generates the new element connectivities for the elements created by the refinement of an element and also generates and investigates the multipoint constraint equations. The constraint equations are calculated by subroutines ELEML, ELEMQ, ELEMC or ELEMP depending upon the element type. Subroutine REFINE does not generate or investigate those nodes that are completely internal to the parent element and generates constraints only for new nodes (not previously in the nodal coordinate array). However, if a node is a border node and is not new it is checked to see if it is already specified as a dependent node. If it is a dependent node then the equation is checked to ensure all the nodes are included in the connectivity of the element being refined. If the nodes are included then the equation is investigated further to determine if the element being refined is the only element defining the independent basis (for the constraint equation). This operation is performed by calling subroutine ELEMR with the ISW parameter set to three. The constraint equation is checked against all the original element connectivities. If the element being refined is the only element defining the independent basis then the constraint is removed. Otherwise the node is an "interface" node and this is indicated by filling the second position of the NIDO array with the number of elements containing the independent basis in their connectivitity. The identification of the nodes as an "interface" node is required for subroutine MPC.

4.6 Constraint Equation Operations in Subroutine MPC

Subroutine MPC determines which multipoint constraint equations are required and eliminates those which are not. The procedure to determine whether a constraint can be removed is described as follows:

1. All the elements connectivities (if superelements are present, then the connectivities for substructure being refined) are read. After each connectivity is read, the independent nodes of each multipoint constraint equation are compared against the connectivity nodes for all elements. If all independent nodes are present in any connectivity list, the equation must be retained. Otherwise, further investigation of the multipoint constraint equation is necessary before it can be removed.

2. If all independent nodes of any multipoint constraint equation are not contained in any element connectivity, the equation is checked to ensure its independent nodes are not in fact dependent (as a result of constraint equations considered previously). If dependent nodes are found and the node is not an "interface" node then the constraint equation remains. If not, and the geometric data is also not substructured, the constraint is marked to be removed.

3. This step is performed only if the file is substructured. The independent nodes are checked to see if they correspond to master nodes and if the master nodes are found only in the substructure being refined, then the constraint is removed. If the nodes do not correspond to master nodes and the node is marked as an interface node then the constraint is also removed. This step also serves to identify "boundary" nodes which are required for subroutine MPCSE

since the constraint may still yet be removed depending upon the refinement of other substructures.

After the above steps are performed for an unsubstructured model, the constraints which remain are condensed and skew effects are taken into account. The resulting final data is written to the output SMD file.

For a substructured file, additional steps are performed and the file data is not condensed. The "boundary" nodes identified in step 3 have their coordinate locations compared to the existing list of boundary nodes to determine their boundary node number. This procedure is performed by subroutine IDBLIS. The master node array for the substructure being refined is then filled with the negative "boundary" node number. Next, the independent nodes are compared to the master node list and if they are not present then the list is incremented. In the case of a boundary node, the boundary node number and the master node number it is dependent upon are written to the NBOR file for processing in subroutine MPCSE. The data for all the constrained nodes is written to the NREB file.

4.7 Constraint Equation Operations in Subroutine MPCSE

After the refinement of all elements in all substructures has been completed, subroutine MPCSE is called. At this point, all the multipoint constraint equations have been computed and stored on the NREB file. These equations are adjusted to take into effect skew conditions but are not condensed. This subroutine makes use of existing array space. The steps to determine whether the constraint is to remain are as follows:

1. A loop over the superelements is performed to read the data on the NBOR file and fill the NJ array with the master node numbers that the boundary node is dependent upon. The NBOR file was generated in subroutine MPC and has the format given in Table 4.1. The substruc-

ture and master node numbers for each substructure are written to a file and a second file containing the boundary node numbers is created.

2. A loop over the superelements is again performed to read the master node numbers in core and compare them against the array generated in step 1 to determine if the superelement contains all the master nodes that the boundary node depends upon. If YES, then another array, IREC, is filled with the superelement number and the counter for the number of superelements containing the independent basis of the equation is incremented.

3. The superelements are again looped over and the boundary nodes are now placed in core and compared against the array generated in step 2. The constraint remains on a node if the independent basis is contained in the superelement as flagged in the IREC array but the superelement does not contain the boundary node as determined by the superelements boundary node list. The one exception is if the independent basis is found only in the superelement then it is further investigated to determine if the node is an interior node. If the node is an interior node, it is neither a slave or master node.

4. New master node numbers are assigned to all the border nodes flagged as having their constraint removed.

5. The superelements are again looped over and the following steps are performed:

   (a) The boundary node numbers flagged as negatives in the master node list for each superelement are substituted with the new master node number calculated in Step 4.

(b) The multipoint constraint data generated by subroutine MPC is read and the data is checked to see which constraints may be removed. If the node was determined to be a master node, then the constraint is removed at this point.

(c) The multipoint constraint equations to remain are condensed.

(d) The data is transformed from the ISTIFM data format to the IELEMB format and written to the output file.

4.8 Sample Problems

In this section, the results of refining an unsubstructured model is first presented to demonstrate how the REFINE program determines which nodes to have multi-point constraint equations applied. A substructured model is then refined to illustrate the treatment of boundary nodes.

Unsubstructured Model Problem

The unrefined model is a square plate composed of nine 4-noded quadrilateral shell elements (IEC=5) arranged in a 3x3 mesh. The central element is first refined to order 2 and then two of the resulting elements are refined again to order 2. The resulting mesh is shown in Figure 4.3 and the following discussion will refer to it.

The refinement of the central element results in new nodes 17 through 21. The border nodes 17, 18, 20 and 21 are specified as constrained in subroutine REFINE.

The refinement of the first new element (with corner nodes 6, 17, 19, 18) results in new nodes 22 through 26. The border nodes 22, 23, 25 and 26 are specified as constrained in subroutine REFINE.

The refinement of the second new element (with corner nodes 17, 7, 20, 19) results in new nodes 27 through 30. The border nodes 27, 29 and 30 are specified as constrained in Subroutine REFINE. Node 25 is a border node but is not new and is determined to have been already specified as a constrained node. The node is specified as an "interface" node since its independent basis is contained in more than one element.

The element refinement specifications are complete and now subroutine MPC is used to determine which constraint equations are required. The constraint equations generated are uncondensed which means a constrained node is related to its two adjacent nodes along the edge being refined. Table 4.2 identifies the independent nodes for each dependent node for the uncondensed equations of this sample problem. Constrained nodes 17, 18, 20 and 21 remain since the comparison against the connectivity list revealed that the independent nodal basis for each is contained in the connectivity of at least one element in the model. Constrained nodes 22, 23, 26, 27, 29 and 30 also remain since the independent nodal basis of each contains dependent nodes which are not *interface* nodes. The constraint equations on node 25 were removed at this point since the node was flagged as an interface node.

The constrained nodes to remain are condensed and the resulting data is written to the input file.

Substructured Model Problem

The unrefined model is a ship transverse bulkhead composed of four structures/superelements shown in Figure 4.4. Both superelements 1 and 2 will be partially refined along their interface and the resulting mesh is shown in Figure 4.5. VASTG graphics has no provision for identification of constrained nodes (slave nodes) for substructures. It is therefore not possible to demonstrate the operation of REFINE for substructures using graphics only and necessary reference will be made to the superelement data (PREFX.SED).

The superelement data for the unrefined model is shown in Table 4.2. Superelement 1 has 64 master nodes which are plotted in Figure 4.6 using the option to plot border nodes labelled with local node numbers and superelement 2 has 80 nodes which are plotted in Figure 4.7. The refinement of superelements 1 and 2 result in the generation of models with nodes as shown in Figures 4.8 and 4.9. The new nodes generated internally to the refined mesh create no changes to the number of slave or master nodes. Table 4.5 presents the refined superelement data for the model. The ten nodes in each refined superelement which are identified as slave nodes in this data have been labelled selectively as shown in Figures 4.10 and 4.11. The two new nodes located along the interface between superelements 1 and 4, namely nodes 160 and 162, remain constrained as do nodes 167 and 146 located between superelements 2 and 3 since superelements 3 and 4 are not refined. These four nodes are interface nodes and are marked as negative in the restart data produced under the REFINE header in the final output file. Chapter 5.0 contains a description of the reason for this procedure.

Nodes 145 and 161 of superelement 1 and nodes 166 and 147 of superelement 2 are identified as slave nodes. These nodes would become master nodes if boundary conditions were applied to them. The remainder of the nodes labelled as slave nodes are internal to each substructure and are constrained since their dependent basis is contained in another element. The new master nodes which are identified as a result of the refinement are the new nodes located along the interface between superelements 1 and 2 and also the four existing nodes required to be identified as master nodes for each refined superelement because these nodes were identified in the dependent basis of a constrained nodes.

TABLE 4.1

Format of File NBOR Generated in Subroutine MPC

```
NSS  [ NGN,NBORD,NDDOF,IDCNK,XO,YO,ZO
     [
     [ NBORD [ BORD NO., NMNP, (MMM(I),I=1,NMNP)
     [       [
     [       [ where: MMM contains master node
     [       [        numbers the boundary node
     [       [        is dependent upon
     [
     [ NSUB,NMN,NBORD
     [
     [ (NMN1(I),I=1,NMN)
     [ (NMN2(I),I=1,NMN)
     [ (SPACE(I),I=1,NBORD)

       where: NMN1 contains substructure node numbers
              NMN2 contains master node numbers and boundary
                   nodes are input as negative number
              SPACE contains boundary node numbers
```

TABLE 4.2

Uncondensed MPC for Sample Problem

Dependent Nodes in MPC Equation	Independent Nodes in MPC Equation
17	6,7
18	6,10
20	7,11
21	10,11
22	6,17
23	6,18
25	17,19
26	18,19
27	7,17
29	7,20
30	19,20

TABLE 4.3

Superelement Data for Unrefined Bulkhead Model

```
  4
  1   64    0    OCPF MAJOR BKHD AT STATION 34.0 BKHD 269,270 & 271 COMPLETES SHAPE
  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
 33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
 49   50   51   52   53   54   55   56   57   69   70   91   92  104  107  110
  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
 33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
 49   50   51   52   53   54   55   56   57   58   59   60   61   62   63   64
   0.000     0.000     0.000
  2   80    0    OCPF MAJOR BKHD AT STATION 34.0 BKHD 134,270 & 271 COMPLETES SHAPE
  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
 33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
 49   50   51   52   53   54   55   56   57   69   70   91   92  101  104  105
106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121
 65   66   67   68   69   70   71   72   73   74   75   76   77   14   78   79
 80   81   82   83   84   85   86   87   88   89   90   91   92   93   94   95
 96   97   98   99  100   38  101  102  103  104  105  106  107  108  109  110
111  112   51  113  114  115  116  117  118   58   59   60   61  119  120  121
122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137
   0.000     0.000     0.000
  3   86    0    OCPF MAJOR BKHD AT STATION 34.00 ,BKHD 134,269 & 271 COMPLETES SHAPE
  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
 33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
 49   50   51   52   53   54   55   56   63   64   65   66   67   68   59   87
 88   89   90   96  101  102  104  105  106  111  112  113  115  118  119  123
125  126  127  128  131  132
  1    2    3    4    5    6    7    8    9   10   11   12   13   14  138  139
140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155
156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171
172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187
188  189  190  191  192  193  194  195  196  197  198  199  200  201  202   64
203  204  205  206  207  208
   0.000     0.000     0.000
  4   86    0    OCPF BKHD AT STATION 34.0 BKHD 270, 269 & 134 COMPLETES SHAPE
  1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16
 17   18   19   20   21   22   23   24   25   26   27   28   29   30   31   32
 33   34   35   36   37   38   39   40   41   42   43   44   45   46   47   48
 49   50   51   52   53   54   55   56   63   64   65   66   67   68   69   87
 88   89   90   96  101  102  104  105  106  111  112  113  115  118  119  123
125  126  127  128  135  136
 65   66   67   68   69   70   71   72   73   74   75   76   77   14  209  210
211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226
227  228  229  230  231  232  233  163  234  235  236  237  238  239  240  241
242  243  244  175  245  246  247  248  249  250  251  252  253  254  255  256
257  258  259  191  192  193  260  261  262  263  198  264  265  266  267  128
268  269  270  271  272  273
   0.000     0.000     0.000
```

TABLE 4.4

Superelement Data for Refined Bulkhead Model

```
   1    74    10
   1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16
  17    18    19    20    21    22    23    24    25    26    27    28    29    30    31    32
  33    34    35    36    37    38    39    40    41    42    43    44    45    46    47    48
  49    50    51    52    53    54    55    56    57    69    70    91    92   104   107   110
  67    72   125   126   127    89    94   134   135   136
   1     2     3     4     5     6     7     8    .9    10    11    12    13    14    15    16
  17    18    19    20    21    22    23    24    25    26    27    28    29    30    31    32
  33    34    35    36    37    38    39    40    41    42    43    44    45    46    47    48
  49    50    51    52    53    54    55    56    57    58    59    60    61    62    63    64
 274   275   282   283   284   276   277   285   286   287
0.000E+00 0.000E+00 0.000E+00
 119     2     6
  36 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  67 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  36 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  67 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  36 0.000E+00 0.000E+00 0.500E+00-0.125E+03 0.125E+01 0.000E+00
  67 0.000E+00 0.000E+00 0.500E+00 0.125E+03-0.125E+01 0.000E+00
  36 0.000E+00 0.000E+00 0.150E-02-0.250E+00 0.750E-02 0.000E+00
  67 0.000E+00 0.000E+00-0.150E-02-0.250E+00 0.750E-02 0.000E+00
  36 0.000E+00 0.000E+00-0.150E-04 0.750E-02 0.500E+00 0.000E+00
  67 0.000E+00 0.000E+00 0.150E-04 0.750E-02 0.500E+00 0.000E+00
  36 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  67 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
 120     2     6
  67 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  72 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  67 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  72 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  67 0.000E+00 0.000E+00 0.500E+00-0.109E+03-0.125E+01 0.000E+00
  72 0.000E+00 0.000E+00 0.500E+00 0.109E+03 0.125E+01 0.000E+00
  67 0.000E+00 0.000E+00 0.172E-02-0.250E+00-0.862E-02 0.000E+00
  72 0.000E+00 0.000E+00-0.172E-02-0.250E+00-0.862E-02 0.000E+00
  67 0.000E+00 0.000E+00 0.198E-04-0.862E-02 0.500E+00 0.000E+00
  72 0.000E+00 0.000E+00-0.198E-04-0.862E-02 0.500E+00 0.000E+00
  67 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  72 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
 121     2     6
  49 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  72 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  49 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  72 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  49 0.000E+00 0.000E+00 0.500E+00 0.109E+03 0.000E+00 0.000E+00
  72 0.000E+00 0.000E+00 0.500E+00-0.109E+03 0.000E+00 0.000E+00
  49 0.000E+00 0.000E+00-0.172E-02-0.250E+00 0.000E+00 0.000E+00
  72 0.000E+00 0.000E+00 0.172E-02-0.250E+00 0.000E+00 0.000E+00
  49 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.500E+00 0.000E+00
  72 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.500E+00 0.000E+00
  49 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  72 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
 128     2     6
  49 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  89 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  49 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  89 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  49 0.000E+00 0.000E+00 0.500E+00-0.106E+03 0.125E+01 0.000E+00
  89 0.000E+00 0.000E+00 0.500E+00 0.106E+03-0.125E+01 0.000E+00
  49 0.000E+00 0.000E+00 0.176E-02-0.250E+00 0.882E-02 0.000E+00
  89 0.000E+00 0.000E+00-0.175E-02-0.250E+00 0.882E-02 0.000E+00
  49 0.000E+00 0.000E+00-0.208E-04 0.882E-02 0.500E+00 0.000E+00
  89 0.000E+00 0.000E+00 0.208E-04 0.882E-02 0.500E+00 0.000E+00
  49 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  89 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
```

TABLE 4.4 (Continued)

129	2	6				
89	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
89	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
89	0.000E+00	0.000E+00	0.500E+00	-0.987E+02	0.250E+01	0.000E+00
94	0.000E+00	0.000E+00	0.500E+00	0.987E+02	-0.250E+01	0.000E+00
89	0.000E+00	0.000E+00	0.190E-02	-0.250E+00	0.190E-01	0.000E+00
94	0.000E+00	0.000E+00	-0.190E-02	-0.250E+00	0.190E-01	0.000E+00
89	0.000E+00	0.000E+00	-0.480E-04	0.190E-01	0.500E+00	0.000E+00
94	0.000E+00	0.000E+00	0.480E-04	0.190E-01	0.500E+00	0.000E+00
89	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
130	2	6				
12	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.500E+00	0.101E+03	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.500E+00	-0.101E+03	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	-0.185E-02	-0.250E+00	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.185E-02	-0.250E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00
94	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
145	2	6				
37	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.700E+02	0.000E+00
38	0.000E+00	0.000E+00	0.500E+00	0.000E+00	-0.700E+02	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
38	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	-0.268E-02	0.000E+00	-0.250E+00	0.000E+00
38	0.000E+00	0.000E+00	0.268E-02	0.000E+00	-0.250E+00	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
160	2	6				
13	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
14	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
14	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.000E+00	0.500E+00	0.125E+01	0.674E+02	0.000E+00
14	0.000E+00	0.000E+00	0.500E+00	-0.125E+01	-0.674E+02	0.000E+00
13	0.000E+00	0.000E+00	-0.516E-04	0.500E+00	-0.139E-01	0.000E+00
14	0.000E+00	0.000E+00	0.516E-04	0.500E+00	-0.139E-01	0.000E+00
13	0.000E+00	0.000E+00	-0.278E-02	-0.139E-01	-0.250E+00	0.000E+00
14	0.000E+00	0.000E+00	0.278E-02	-0.139E-01	-0.250E+00	0.000E+00
13	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
14	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
161	2	6				
36	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
36	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
36	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.668E+02	0.000E+00
37	0.000E+00	0.000E+00	0.500E+00	0.000E+00	-0.668E+02	0.000E+00
36	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
36	0.000E+00	0.000E+00	-0.281E-02	0.000E+00	-0.250E+00	0.000E+00
37	0.000E+00	0.000E+00	0.281E-02	0.000E+00	-0.250E+00	0.000E+00
36	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
162	2	6				
12	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.691E+02	0.000E+00
13	0.000E+00	0.000E+00	0.500E+00	0.000E+00	-0.691E+02	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
13	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	-0.271E-02	0.000E+00	-0.250E+00	0.000E+00
13	0.000E+00	0.000E+00	0.271E-02	0.000E+00	-0.250E+00	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00

TABLE 4.4 (Continued)

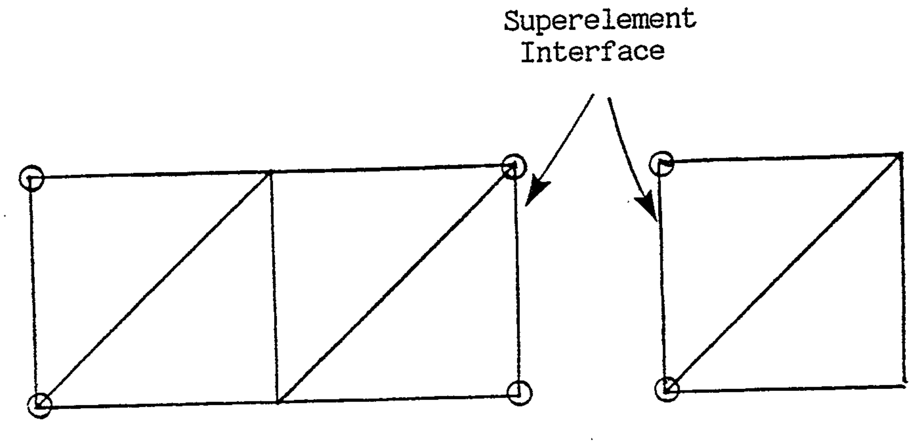
```
    2    91    10
    1     2     3     4     5     6     7     8     9    10    11    12    13    14    15    16
   17    18    19    20    21    22    23    24    25    26    27    28    29    30    31    32
   33    34    35    36    37    38    39    40    41    42    43    44    45    46    47    48
   49    50    51    52    53    54    55    56    57    69    70    91    92   101   104   105
  106   107   108   109   110   111   112   113   114   115   116   117   118   119   120   121
   67    72    89    94   144   145   160   162   163   164   165
   65    66    67    68    69    70    71    72    73    74    75    76    77    14    78    79
   80    81    82    83    84    85    86    87    88    89    90    91    92    93    94    95
   96    97    98    99   100    38   101   102   103   104   105   106   107   108   109   110
  111   112    51   113   114   115   116   117   118    58    59    60    61   119   120   121
  122   123   124   125   126   127   128   129   130   131   132   133   134   135   136   137
  278   279   280   281   287   298   285   286   282   283   284
 0.000E+00 0.000E+00 0.000E+00
  124     2     6
   36 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   67 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   36 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   67 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   36 0.000E+00 0.000E+00 0.500E+00-0.125E+03-0.125E+01 0.000E+00
   67 0.000E+00 0.000E+00 0.500E+00 0.125E+03 0.125E+01 0.000E+00
   36 0.000E+00 0.000E+00 0.150E-02-0.250E+00-0.750E-02 0.000E+00
   67 0.000E+00 0.000E+00-0.150E-02-0.250E+00-0.750E-02 0.000E+00
   36 0.000E+00 0.000E+00 0.150E-04-0.750E-02 0.500E+00 0.000E+00
   67 0.000E+00 0.000E+00-0.150E-04-0.750E-02 0.500E+00 0.000E+00
   36 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   67 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  125     2     6
   67 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   72 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   67 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   72 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   67 0.000E+00 0.000E+00 0.500E+00-0.109E+03 0.125E+01 0.000E+00
   72 0.000E+00 0.000E+00 0.500E+00 0.109E+03-0.125E+01 0.000E+00
   67 0.000E+00 0.000E+00 0.172E-02-0.250E+00 0.862E-02 0.000E+00
   72 0.000E+00 0.000E+00-0.172E-02-0.250E+00 0.862E-02 0.000E+00
   67 0.000E+00 0.000E+00-0.198E-04 0.862E-02 0.500E+00 0.000E+00
   72 0.000E+00 0.000E+00 0.198E-04 0.862E-02 0.500E+00 0.000E+00
   67 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   72 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  126     2     6
   49 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   72 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   49 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   72 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   49 0.000E+00 0.000E+00 0.500E+00 0.109E+03 0.000E+00 0.000E+00
   72 0.000E+00 0.000E+00 0.500E+00-0.109E+03 0.000E+00 0.000E+00
   49 0.000E+00 0.000E+00-0.172E-02-0.250E+00 0.000E+00 0.000E+00
   72 0.000E+00 0.000E+00 0.172E-02-0.250E+00 0.000E+00 0.000E+00
   49 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.500E+00 0.000E+00
   72 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.500E+00 0.000E+00
   49 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   72 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
  130     2     6
   49 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   89 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   49 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   89 0.000E+00 0.500E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   49 0.000E+00 0.000E+00 0.500E+00-0.106E+03-0.125E+01 0.000E+00
   89 0.000E+00 0.000E+00 0.500E+00 0.106E+03 0.125E+01 0.000E+00
   49 0.000E+00 0.000E+00 0.176E-02-0.250E+00-0.882E-02 0.000E+00
   89 0.000E+00 0.000E+00-0.176E-02-0.250E+00-0.882E-02 0.000E+00
   49 0.000E+00 0.000E+00 0.208E-04-0.882E-02 0.500E+00 0.000E+00
   89 0.000E+00 0.000E+00-0.208E-04-0.882E-02 0.500E+00 0.000E+00
   49 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
   89 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
```

TABLE 4.4 (Continued)

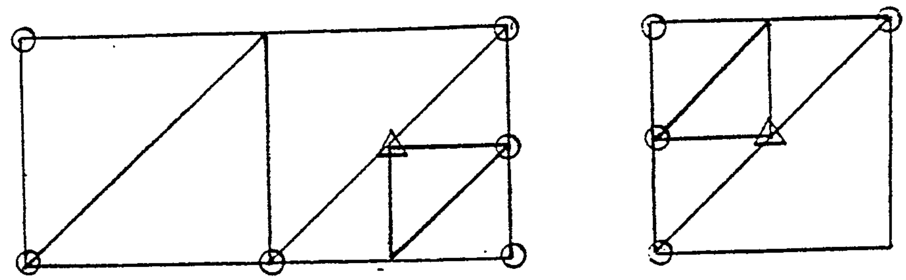
131	2	6				
89	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
89	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
89	0.000E+00	0.000E+00	0.500E+00	-0.987E+02	-0.250E+01	0.000E+00
94	0.000E+00	0.000E+00	0.500E+00	0.987E+02	0.250E+01	0.000E+00
89	0.000E+00	0.000E+00	0.190E-02	-0.250E+00	-0.190E-01	0.000E+00
94	0.000E+00	0.000E+00	-0.190E-02	-0.250E+00	-0.190E-01	0.000E+00
89	0.000E+00	0.000E+00	0.480E-04	-0.190E-01	0.500E+00	0.000E+00
94	0.000E+00	0.000E+00	-0.480E-04	-0.190E-01	0.500E+00	0.000E+00
89	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
132	2	6				
12	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.500E+00	0.101E+03	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.500E+00	-0.101E+03	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	-0.185E-02	-0.250E+00	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.185E-02	-0.250E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00
94	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
94	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
146	2	6				
13	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
14	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
14	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.000E-00	0.500E+00	0.125E+01	-0.694E+02	0.000E+00
14	0.000E+00	0.000E+00	0.500E+00	-0.125E+01	0.694E+02	0.000E+00
13	0.000E+00	0.000E+00	-0.487E-04	0.500E+00	0.135E-01	0.000E+00
14	0.000E+00	0.000E+00	0.487E-04	0.500E+00	0.135E-01	0.000E+00
13	0.000E+00	0.000E+00	0.270E-02	0.135E-01	-0.250E+00	0.000E+00
14	0.000E+00	0.000E+00	-0.270E-02	0.135E-01	-0.250E+00	0.000E+00
13	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
14	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
147	2	5				
37	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.500E+00	0.000E+00	-0.790E+02	0.000E+00
38	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.790E+02	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
38	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.237E-02	0.000E+00	-0.250E+00	0.000E+00
38	0.000E+00	0.000E+00	-0.237E-02	0.000E+00	-0.250E+00	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
38	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
166	2	5				
36	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
36	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
36	0.000E+00	0.000E+00	0.500E+00	0.000E+00	-0.668E+02	0.000E+00
37	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.668E+02	0.000E+00
36	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
36	0.000E+00	0.000E+00	0.281E-02	0.000E+00	-0.250E+00	0.000E+00
37	0.000E+00	0.000E+00	-0.281E-02	0.000E+00	-0.250E+00	0.000E+00
36	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
37	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
167	2	5				
12	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.500E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.500E+00	0.000E+00	-0.691E+02	0.000E+00
13	0.800E+00	0.000E+00	0.500E+00	0.000E+00	0.691E+02	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
13	0.000E+00	0.000E+00	0.000E+00	0.500E+00	0.000E+00	0.000E+00
12	0.000E+00	0.000E+00	0.271E-02	0.000E+00	-0.250E+00	0.000E+00
13	0.000E+00	0.000E+00	-0.271E-02	0.000E+00	-0.250E+00	0.000E+00
12	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
13	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00

TABLE 4.4 (Continued)

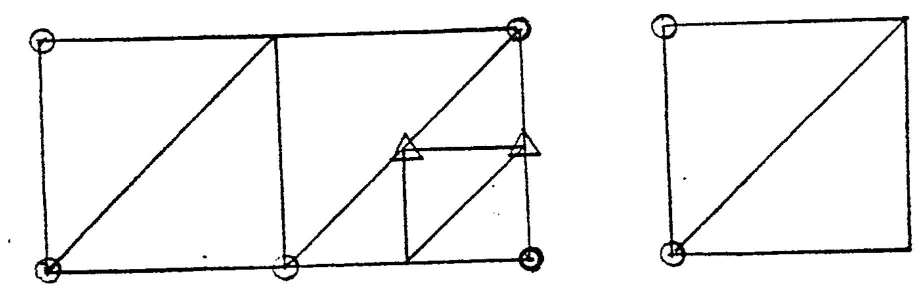
3	86	0													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	63	64	65	66	67	68	69	87
88	89	90	96	101	102	104	105	106	111	112	113	115	118	119	123
125	126	127	128	131	132										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	138	139
140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155
156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171
172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187
188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	64
203	204	205	206	207	208										
0.000E+00	0.000E+00	0.000E+00													
4	86	0													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	63	64	65	66	67	68	69	87
88	89	90	96	101	102	104	105	106	111	112	113	115	118	119	123
125	126	127	128	135	136										
65	66	67	68	69	70	71	72	73	74	75	76	77	14	209	210
211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226
227	228	229	230	231	232	233	163	234	235	236	237	238	239	240	241
242	243	244	175	245	246	247	248	249	250	251	252	253	254	255	256
257	258	259	191	192	193	260	261	262	263	198	264	265	266	267	128
268	269	270	271	272	273										
0.000E+00	0.000E+00	0.000E+00													



(a) Original Finite Element Model - 2 substructures



(b) Refinement of (a) using the option of all superelement interface nodes as master nodes. Requires adjacent superelements to be refined to same order along common interface.



(c) Refinement of (a) using the option of all superelement interface nodes as slave nodes.

FIGURE 4.1: Modelling Options for New Superelement Interface Nodes
(○ - Master Node, Δ - Slave Node)

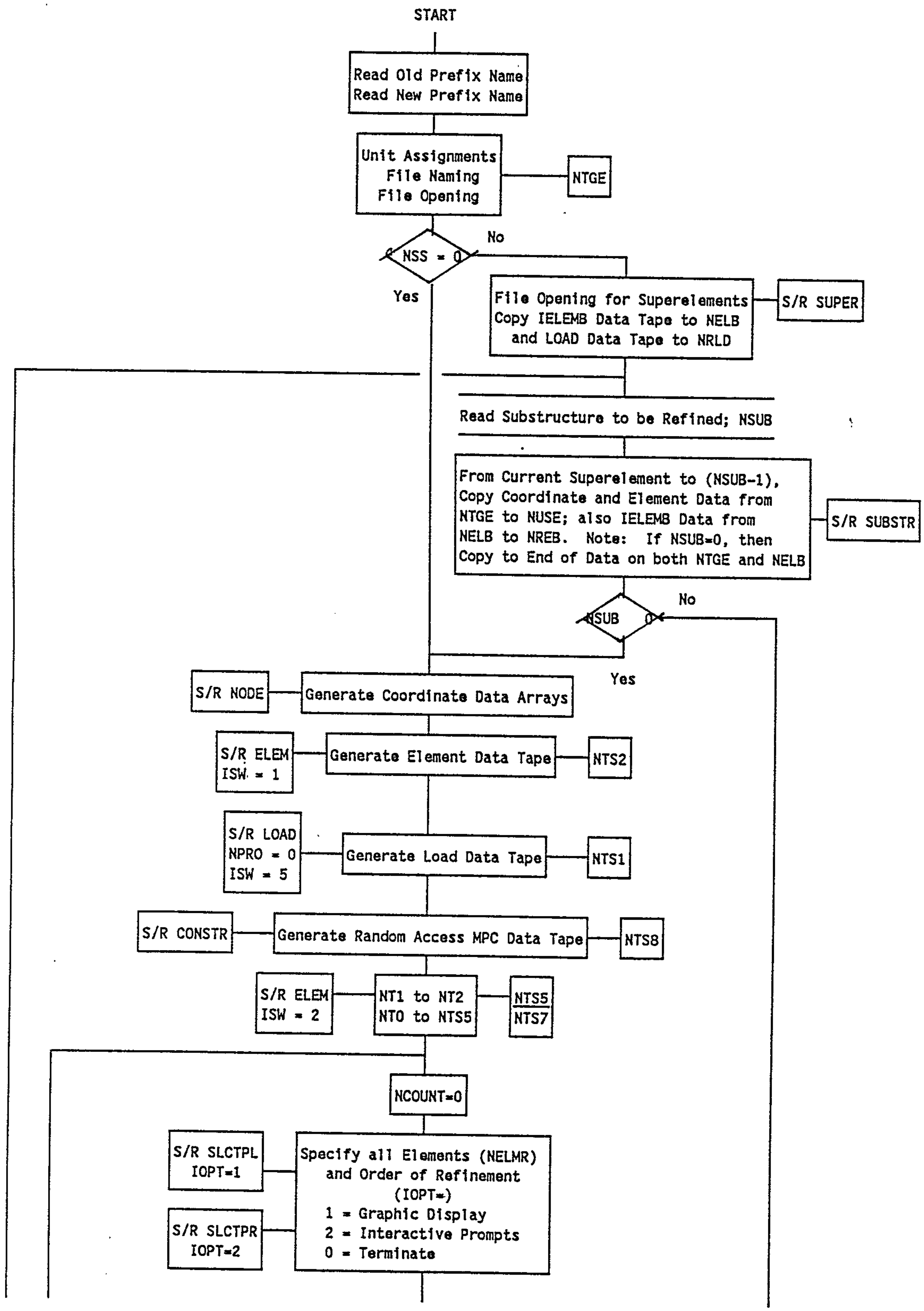


FIGURE 4.2: Flowchart for Program REFINE

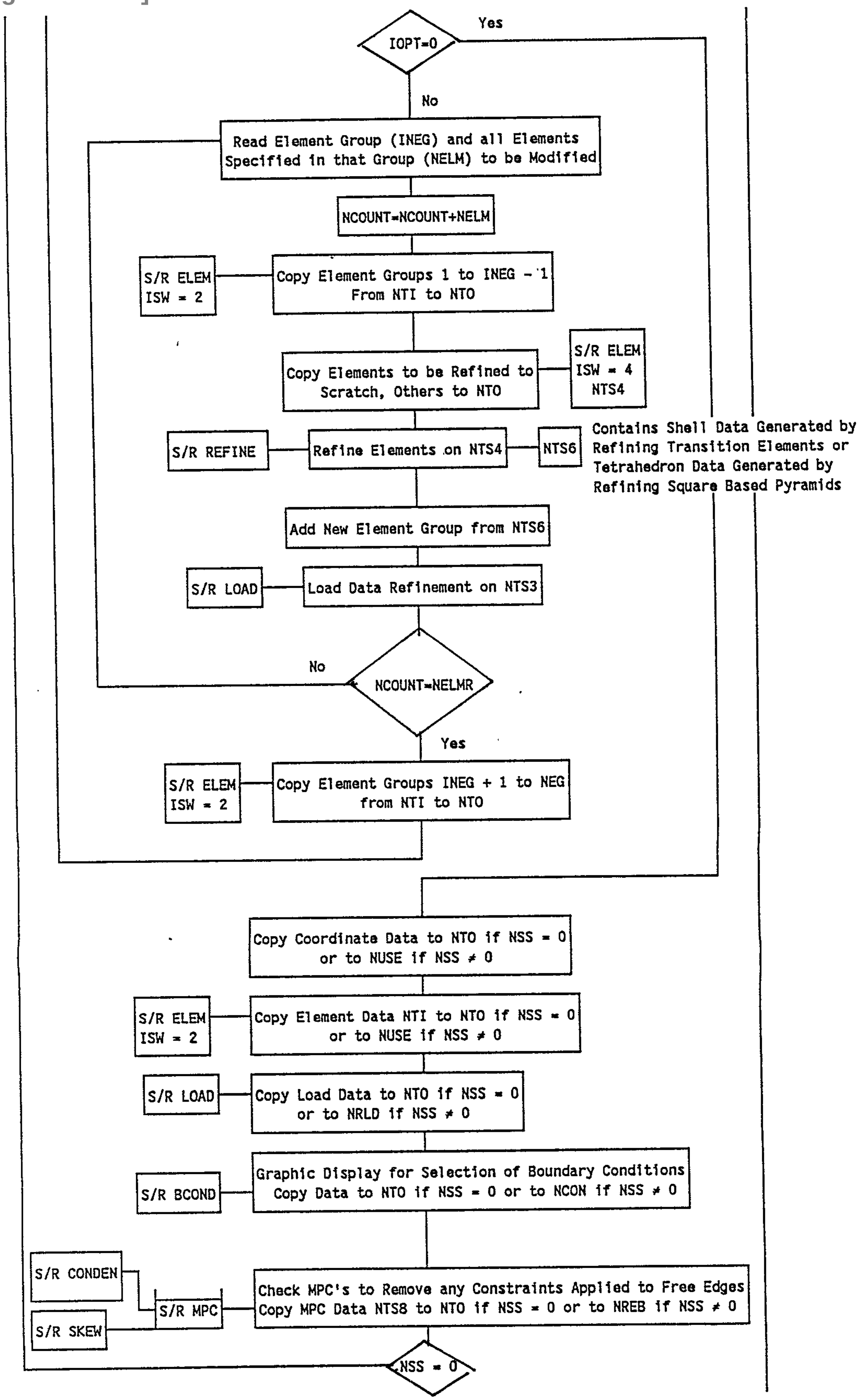


FIGURE 4.2: Flowchart for Program REFINE (Continued)

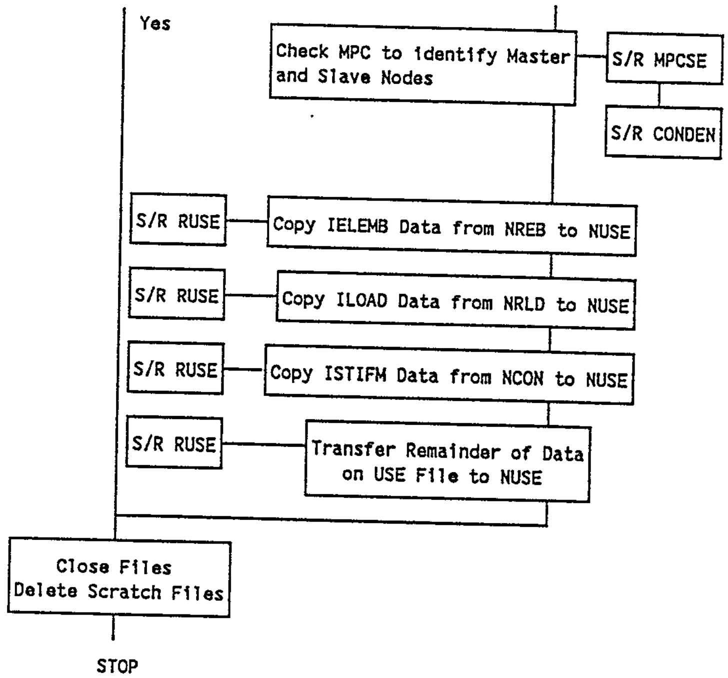


FIGURE 4.2: Flowchart for Program REFINE (Continued)

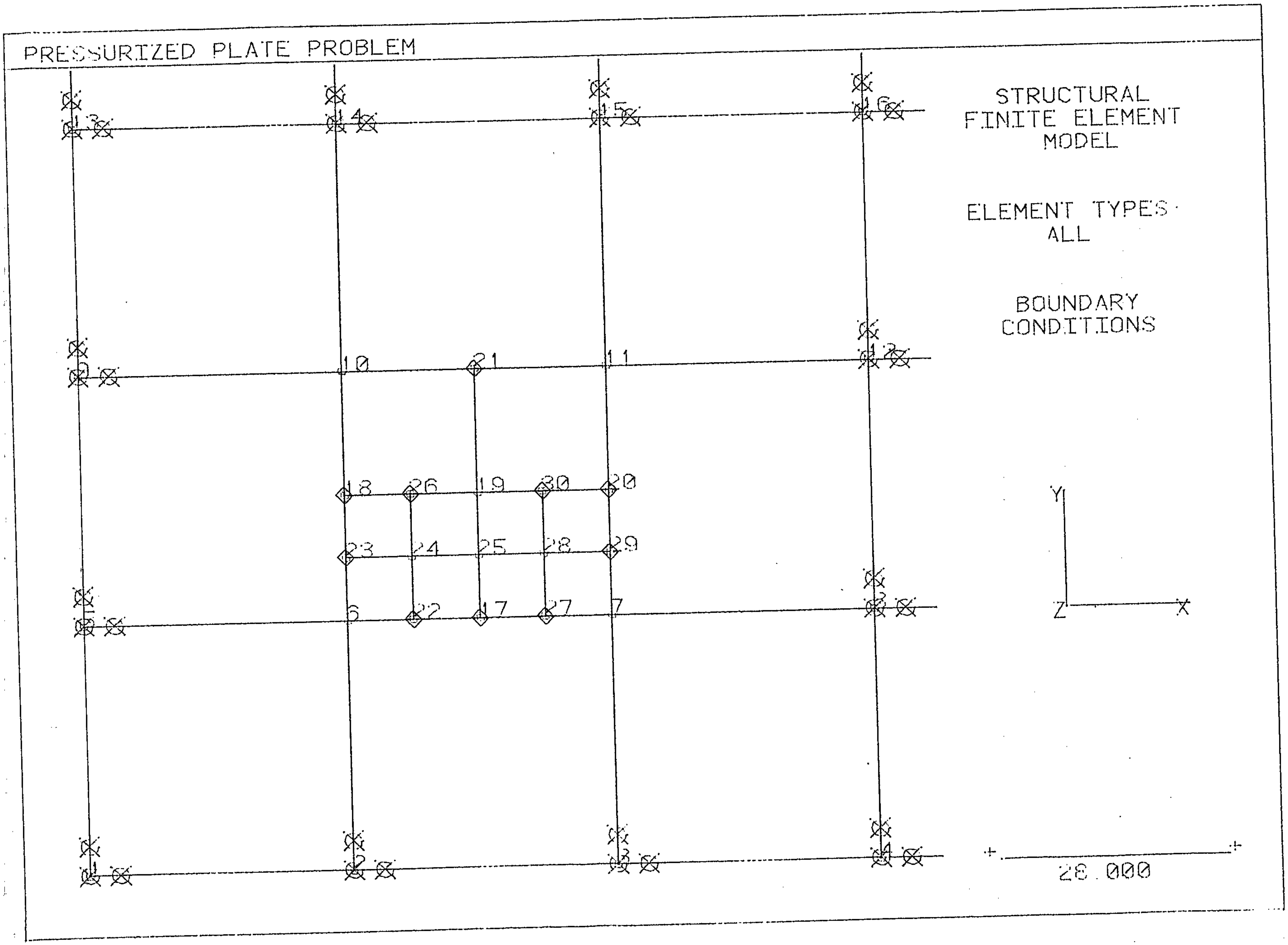


FIGURE 4.3: Recursive Refinement of Square Plate Discretized by 4-Noded Quadrilateral Elements (◇ = Dependent Node)

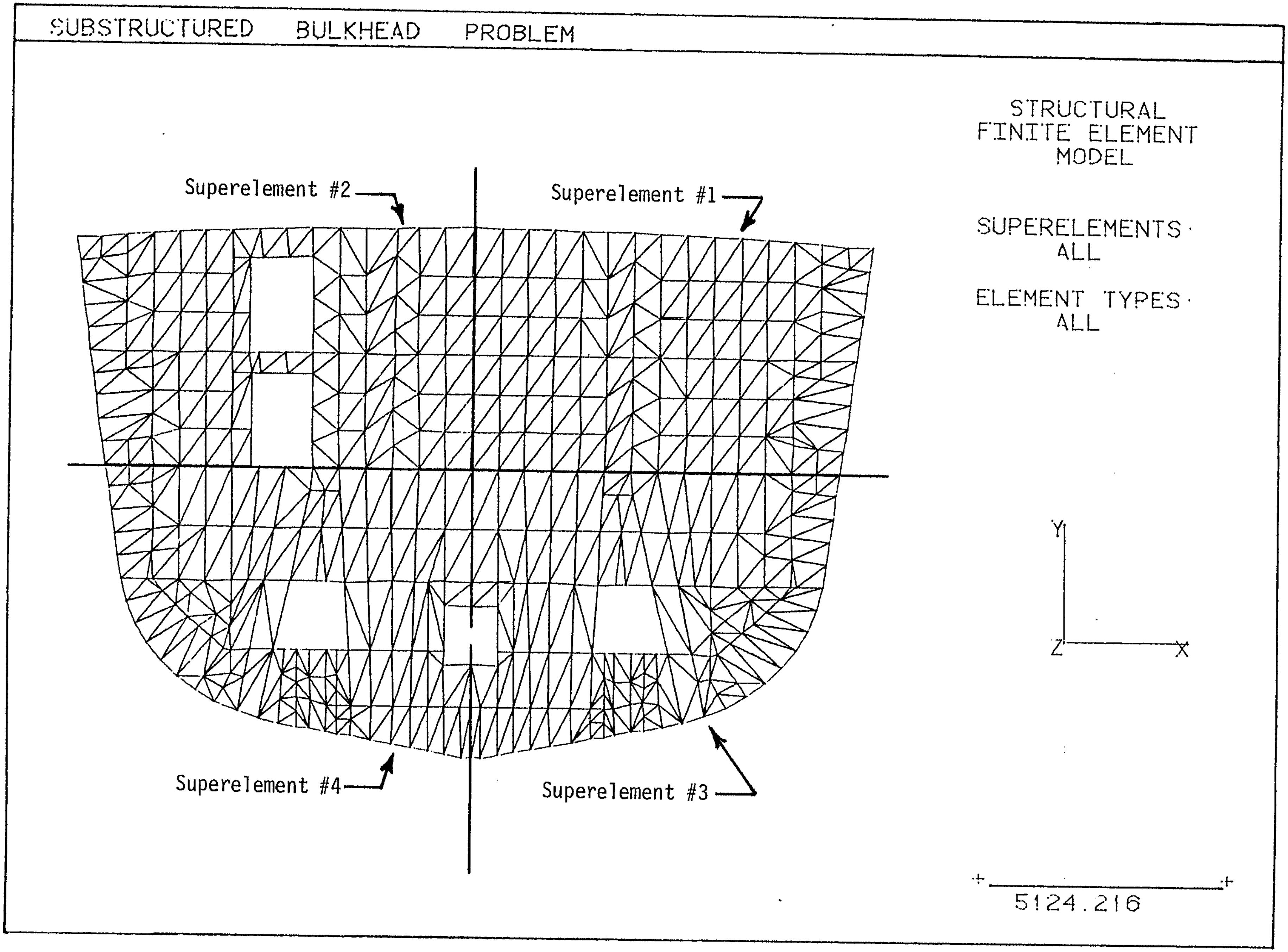


FIGURE 4.4: Substructured Bulkhead Model Before Refinement

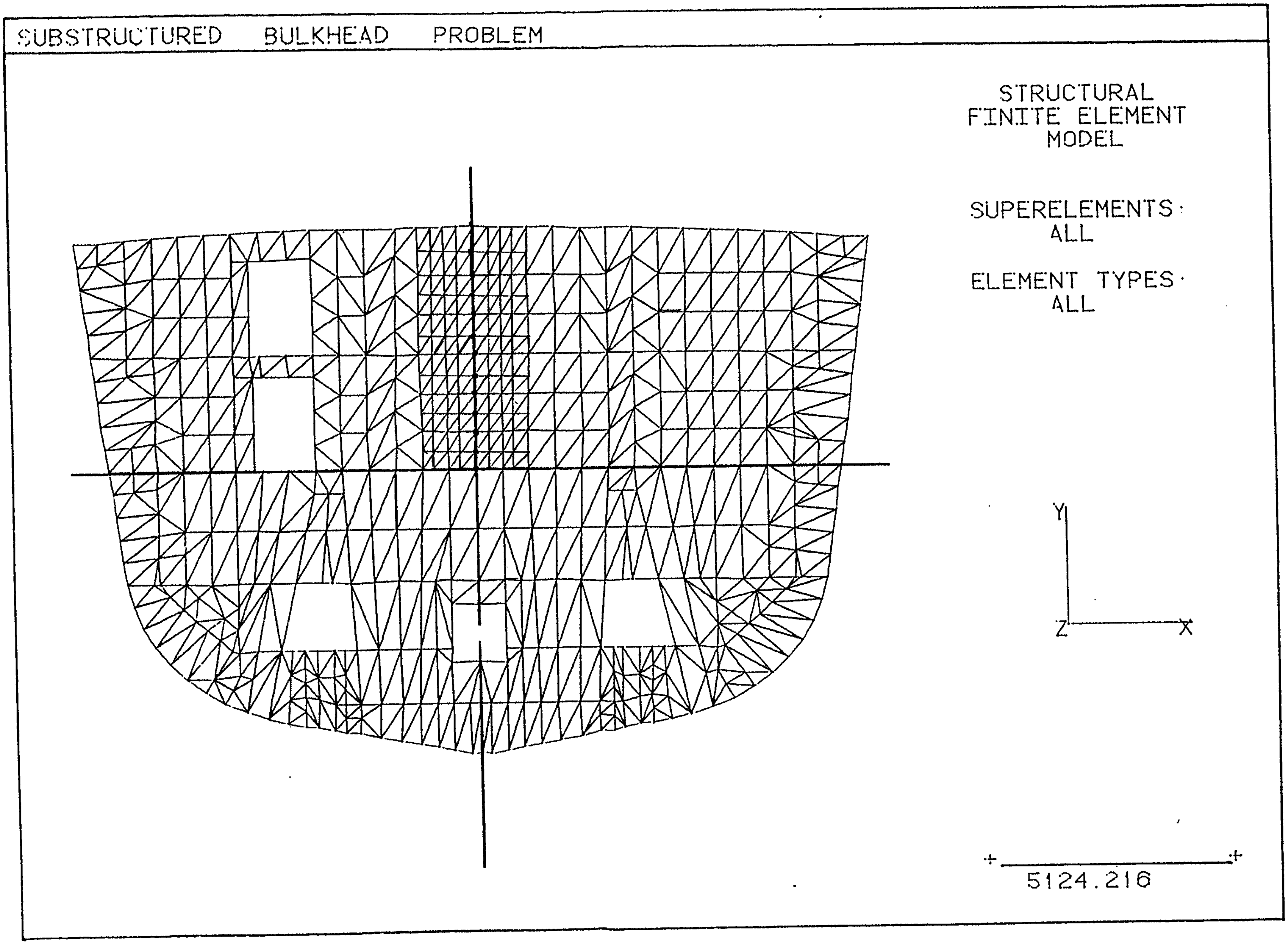


FIGURE 4.5: Substructured Bulkhead Model After Refinement

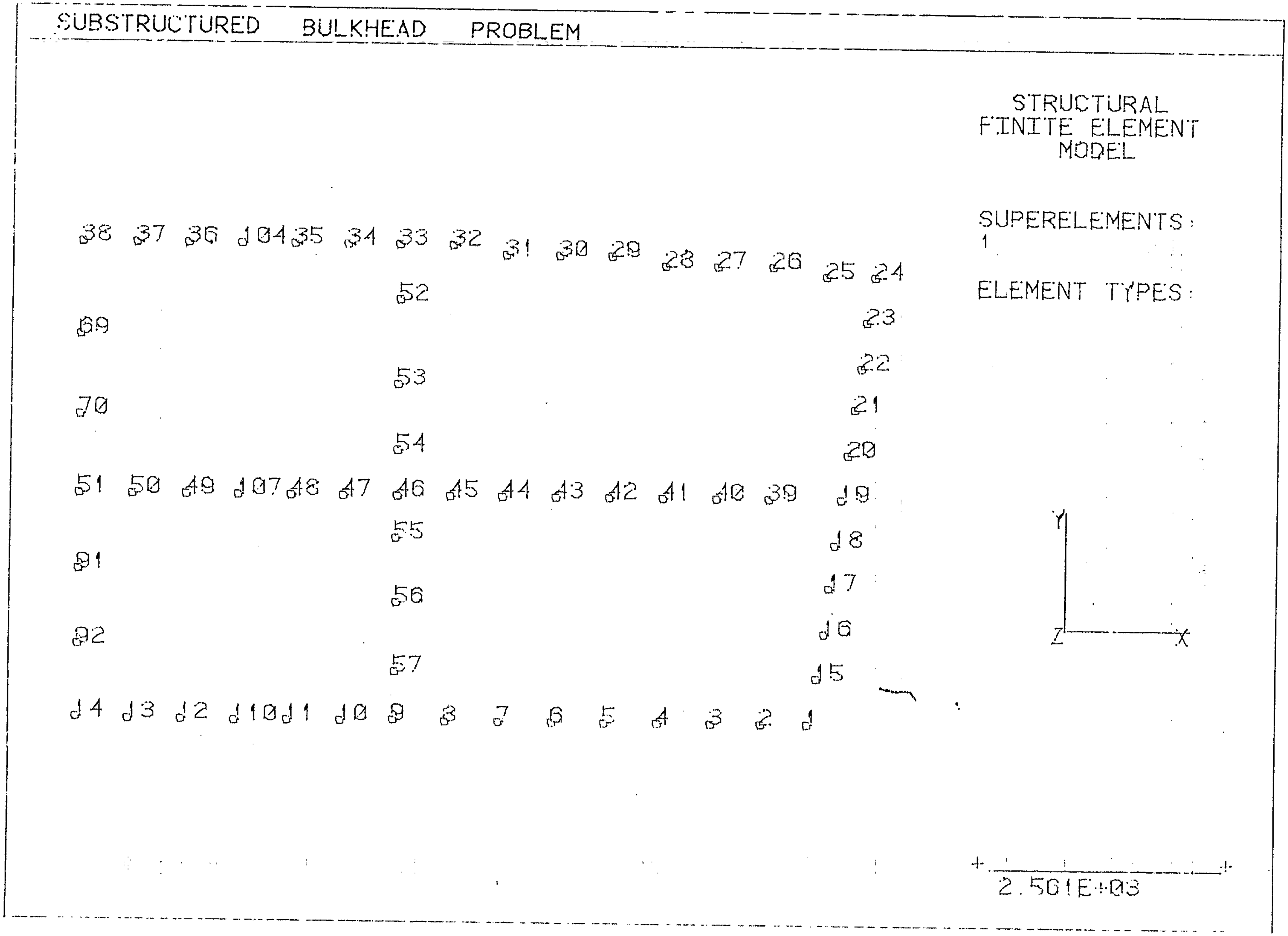


FIGURE 4.6: Master Nodes for First Superelement

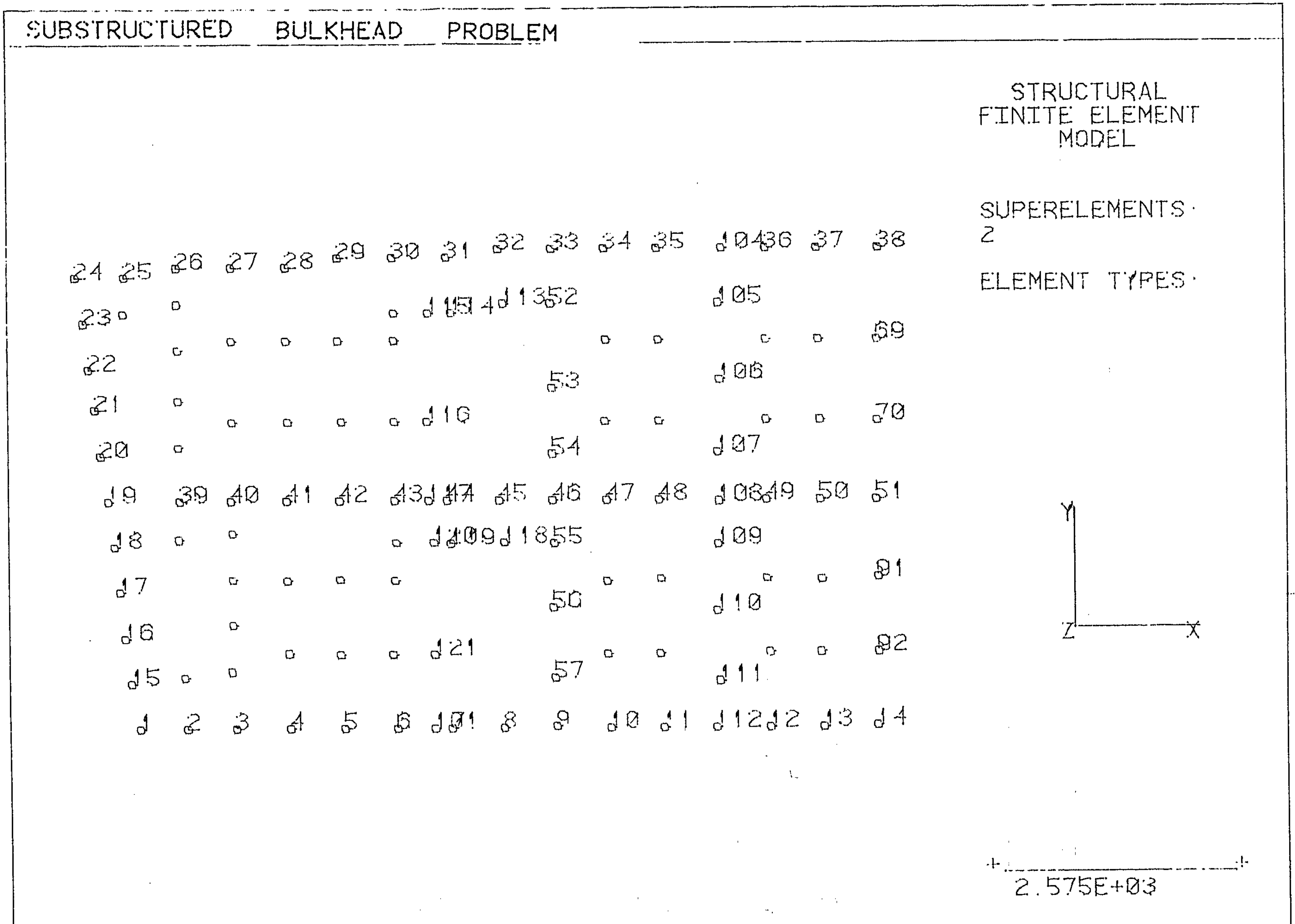


FIGURE 4.7: Master Nodes for Second Superelement

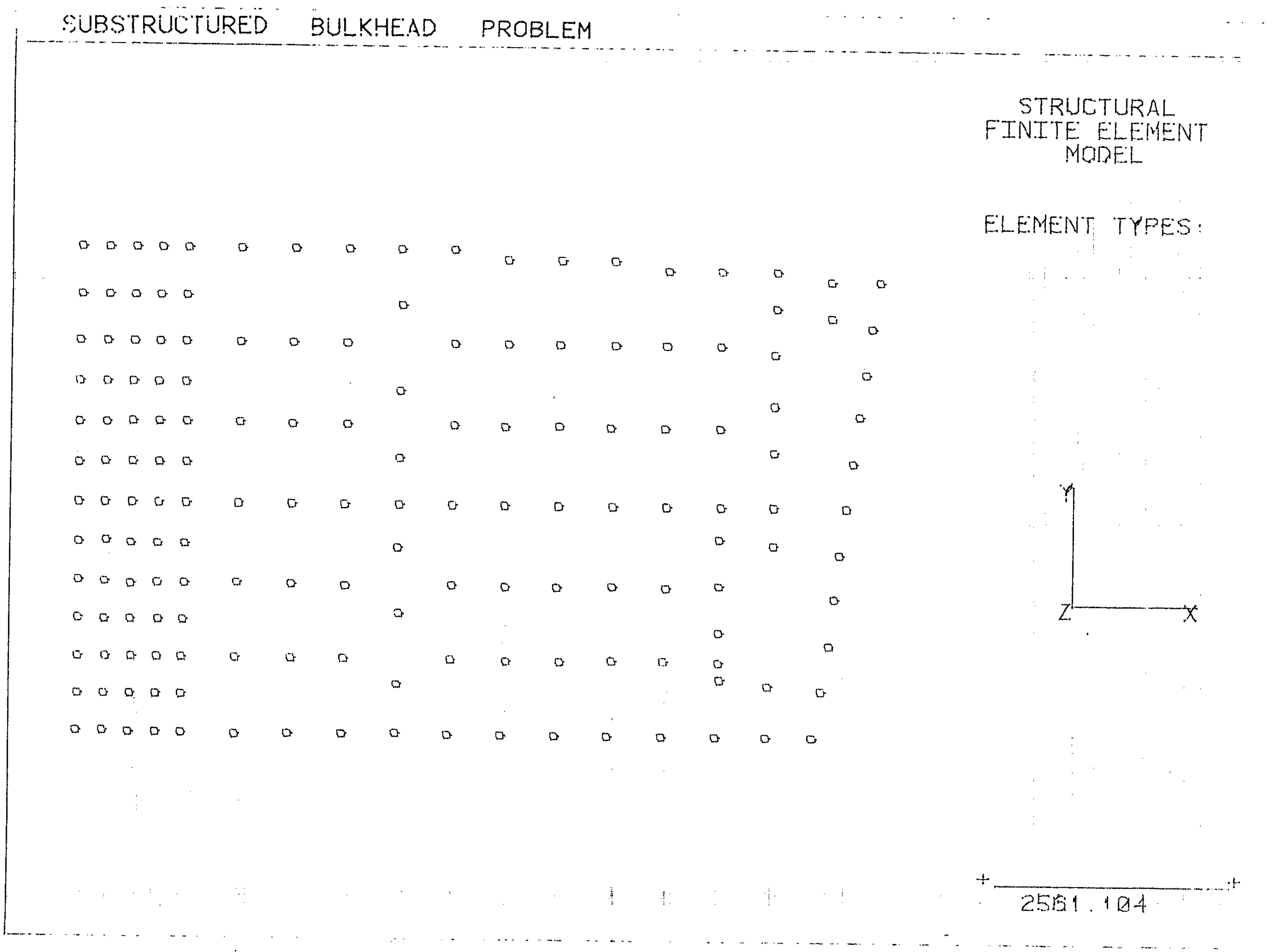


FIGURE 4.8: Nodes for First Superelement After Refinement

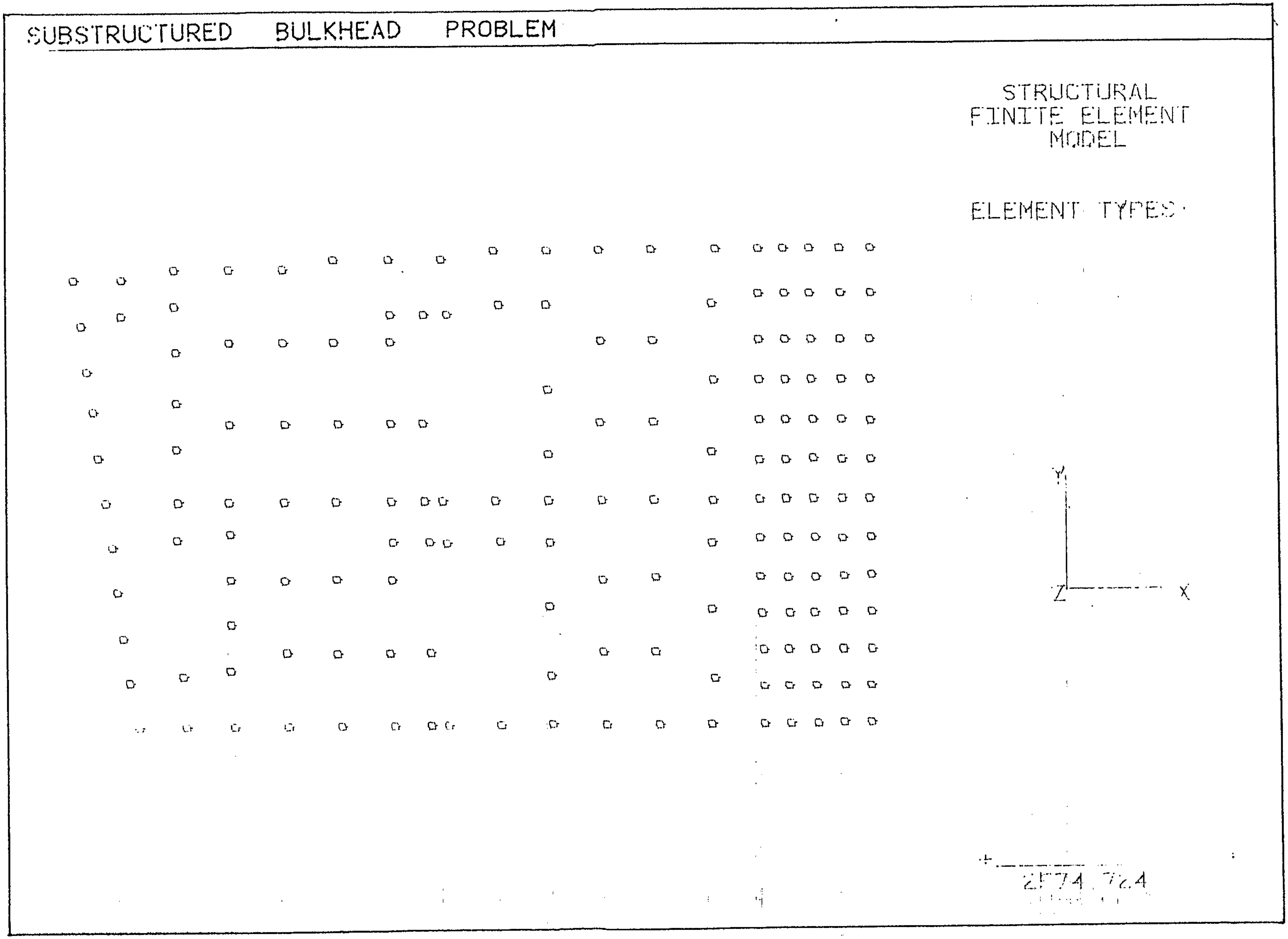


FIGURE 4.9: Nodes for Second Superelement After Refinement

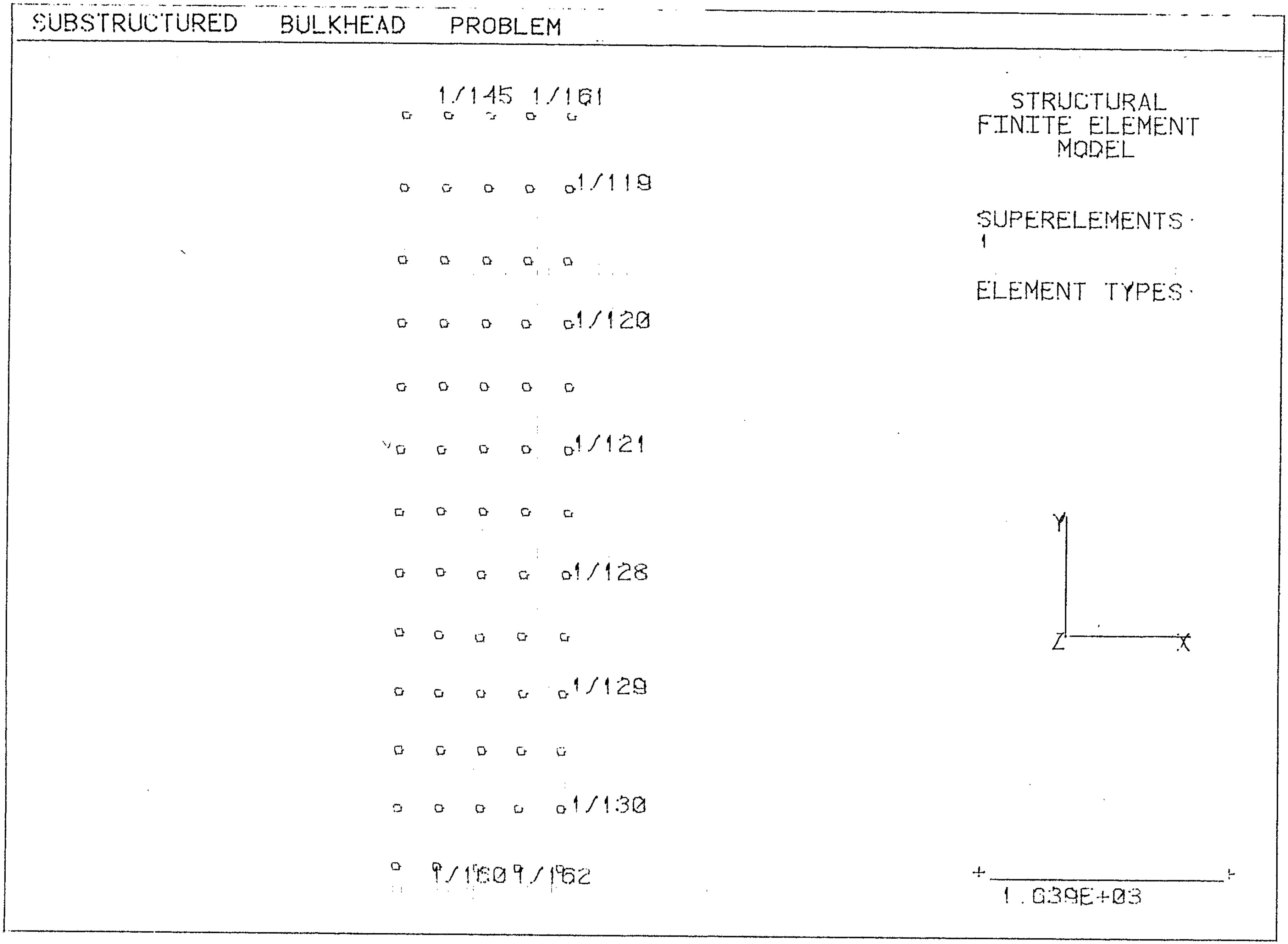


FIGURE 4.10: Slave Nodes for First Superelement Defined Interactively

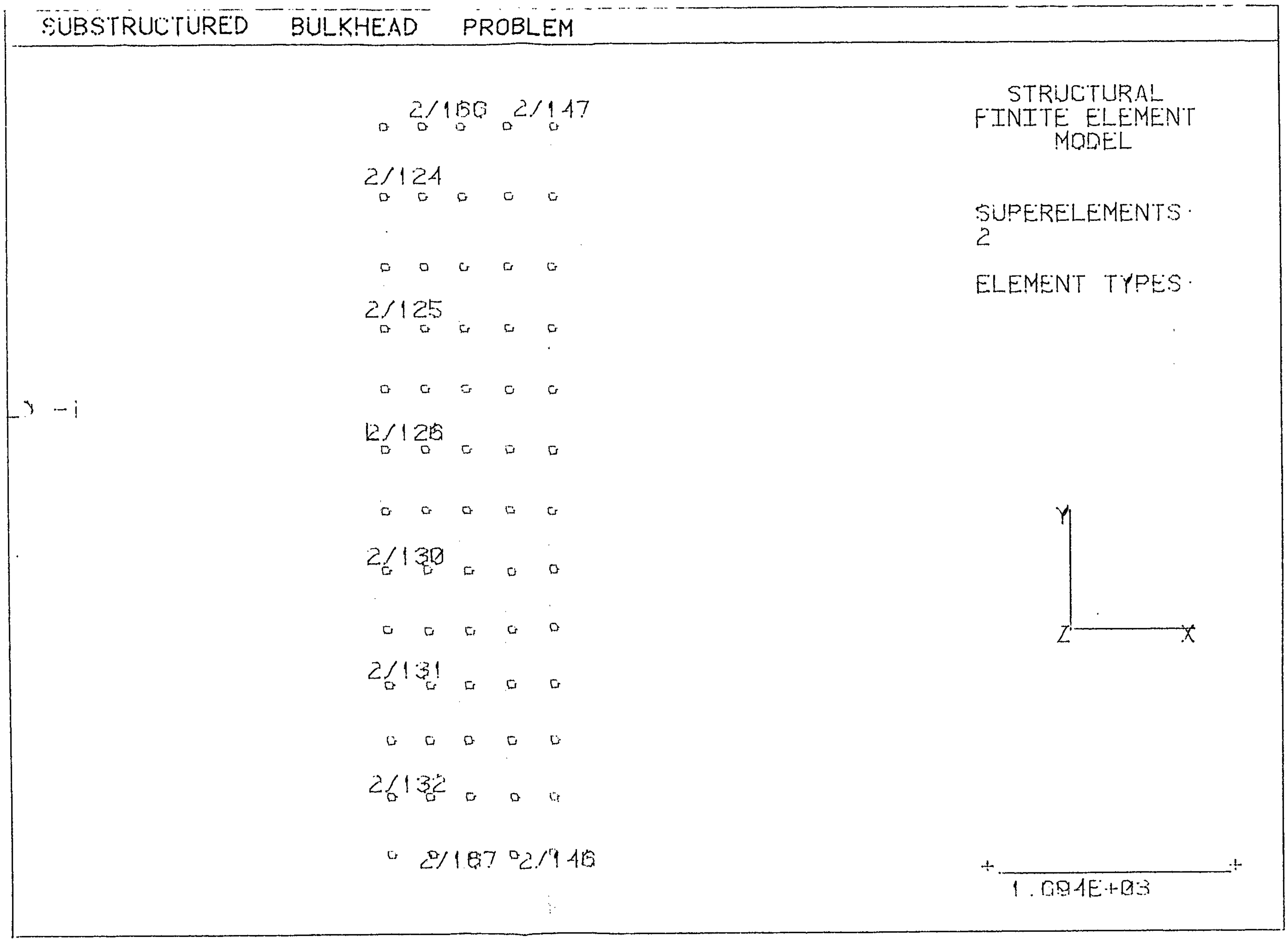


FIGURE 4.11: Slave Nodes for Second Superelement Defined Interactively

CHAPTER 5
RESTART CAPABILITY

5.1 Introduction

The REFINE program may eventually be automated to the point that it could operate as a component of an adaptive finite element model optimization capability which once initiated would, without any user intervention, alternately refine the mesh and resolve until desired levels of accuracy are attained with near optimal distribution of freedoms in the finite element model. But considerable further development of the algorithms within both the discretization error estimation and mesh refinement capabilities, is still required. The capability to refine a previously refined structure provides a so called "restart" feature and moves a step closer to the completely automated procedure.

Prior to the development of this capability, the user had to reconsider the original model and refine the model to increasing levels as desired. Now the user has only to "restart" and refine the previously refined model in areas of interest. Considerable savings in time are possible since the user no longer has to reconstruct the refinement process to attain the starting point from which to perform the desired refinements. The implementation of this capability is discussed in this chapter.

5.2 Implementation

The refinement of an element involves:

1) generation of new nodes;
2) replacement of the connectivity of a parent element with the connectivities of elements produced by the refinement; and
3) generation and manipulation of any resulting constraint equations.

When restarting the refinement of a finite element model, no particular problems are expected with the first two steps but difficulties may arise with the manipulation of constraint equations. The algorithm which determines whether constraint equations remaining on each new node requires that constraint equations be available in the uncondensed form. This implies that the constraint equations for the original model must be available in uncondensed form. The algorithm for evaluating whether constraint equations remain on a node is explained in detail in Chapter 4.

The uncondensed multipoint constraint data for a refined model cannot be readily extracted from the condensed constraint equations data written to the stiffness modification data section of the input file. The key to restarting the refinement of a finite element model is just simply to store the uncondensed constraint equations for any refined model in case the refinement process is to be restarted. The restart data associated with the refined model data is therefore stored on the USE file under the header REFINE. On restarts, the REFINE program searches for this header and transfers the data to a scratch file prior to any constraint data processing.

Additional data is required when restarting the refinement of a substructured model if the nodes which are boundary nodes are to be identified. Recall that a boundary node is one which is located on the perimeter of a superelement. Boundary nodes are flagged under the REFINE header by a change of sign on the dependent node. A new subroutine RESTART was developed to extract the information about the boundary nodes from the "restart" data and to fill the master node and boundary node arrays. Master node and boundary node data are used in subroutine MPCSE to determine whether a constraint equation on a node is to be retained and the node becomes a slave node or alternately the constraint equations on a node are eliminated and the node becomes a master node.

The format of the data under the REFINE header is the same for each superelement of a substructured model as for an unsubstructured model. The variable NDDOF equal to the number of degrees of freedom is entered as zero for those superelements which do not contain any slave nodes.

5.3 Sample Problem

The sequence of steps involved in the recursive refinement of a simple model composed of only two four-noded shell elements is now described. This example illustrates the difference between "uncondensed" and "condensed" equations and the requirement for the former in the restarted refinement of the model. The orginal model composed of two four-noded shell elements is shown in Figure 5.1(a).

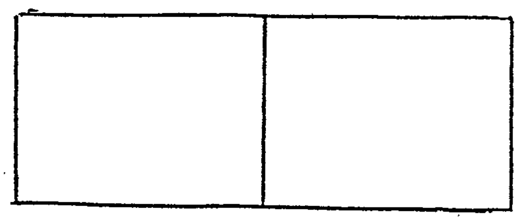
The first step in the refinement of the model is to refine one element, as shown in Figure 5.1(b), and generate a constraint equation for node c in subroutine REFINE with nodes a and b specified as dependent nodes.

One of the new elements is then refined further as shown in Figure 5.1(c). This results in the creation of constraint equations for nodes e,f, and g. Node e is dependent upon nodes a and c, node f is dependent upon nodes c and d, and node g is dependent upon nodes d and h. Because node c is already a dependent node, the constraint equations for nodes e and f are said to be "nested".

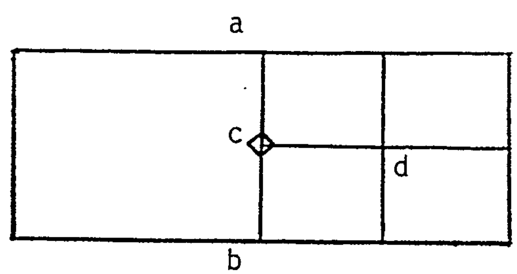
After all specified elements have been refined, the final step dealing with constraint equations involves determining which constraints are to be retained and writing them to the final output file in a format acceptable to VAST. In the example being considered, the constraints on nodes c,f, and g will remain since independent basis of each is contained in connectivity of another element. The constraint on node e also remains since it is dependent upon a dependent node and is not an interface node. The reader will recall that by the convention established in the introduction to the present chapter, an interface node is defined as a node located on the perimeter of a parent element that is common to two or more elements. The constraint equation for nodes e and f are condensed so that no dependent nodes are involved. The final result is node e dependent upon a and b and node f dependent upon nodes a, b and d.

Figure 5.1(d) shows the result of a restart and in which the second of the original elements is refined. Before the restart capability was implemented in the REFINE program, the constraint on e and f would both be incorrectly removed when the second element was refined. The reason for this is that the constraint equations on node c would be removed in subroutine REFINE since the independent basis for this equation would be determined to be only in the element being refined. The constraints in the remaining nodes would again be investigated by subroutine MPC. The constraint equations on node g would remain since its basis, as in the first model refinement, is contained the connectivity of an element. The contraint equations on nodes e and f are however removed since their independent condensed basis are not in the connectivity of any element in the model and there are no dependent nodes in the equations.

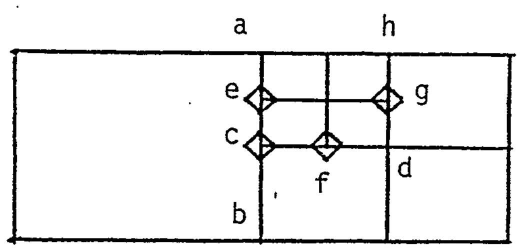
When REFINE is restarted using the uncondensed equations, the constraint equations are manipulated correctly. The constraints on nodes e and f are retained as they should since independent nodal basis is determined to be in an element connectivity.



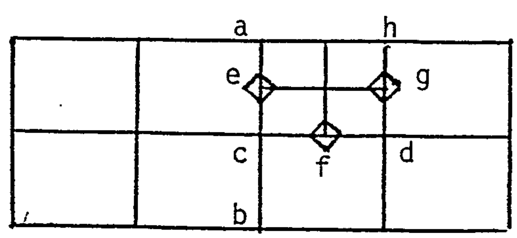
(a) Original mesh with two four-noded shell elements.



(b) Mesh after refining one of the original elements.



(c) Mesh after refining one of the elements generated by refinement.



(d) Mesh after refining second of the original elements.

FIGURE 5.1: Recursive Mesh Refinements and Implication for Dependent Node Determinations
(◇– Dependent Node)

CHAPTER 6
OTHER EXTENSIONS OF SUPERELEMENT REFINEMENT CAPABILITIES

6.1 Introduction

Earlier versions of the program REFINE were not fully operational on superelements. A number of restrictions had to be adopted to facilitate producing a preliminary capability to refine superelements. They were identified in Chapter 4. Items d and e were addressed in Chapters 4 and 5, respectively. Items a to c will be discussed in this chapter.

a. Only level one superelements can be refined.

b. A substructure may only define one superelement.

c. Substructures must be refined in ascending order.

6.2 Extension to Multilevel Superelements

The restriction that only level one superelements can be refined still applies. The extension of the program to multilevel superelements would generalize the range of program application but this development appears to be premature at the moment. It is preferred that the superelement capability be thoroughly tested and evaluated before the multilevel superelement capability be introduced.

6.3 Removal of Limitation that One Substructure Defines One Superelement

The removal of the restriction that one substructure may define only one superelement represents the first step in generalizing the refinement algorithm for superelements. Although necessary program modifications are sketched in the following section, they have not as yet been implemented. The time and fiscal constraints of the current contract did not permit this work to be completed.

Of several possible approaches to perform this task that have been investigated, only the two approaches which appear most feasible are discussed in this section.

One requirement arising from the removal of the limitation that one substructure define one superelement is the need for an accounting procedure to determine the number of new substructures created to relate superelements with them. The only information required to determine the number of substructures in a refined model is a list of superelement numbers to be refined. Prompting the user at the start of the program's execution to enter the superelements to be refined provides data which if placed in an array can be used to direct the refinement of the model.

This data is required by subroutine RELEMB which reads the superelement data and now must relate the superelement to the correct substructure numbers. The VAST data input requirement that superelement must be ordered for increasing substructure number limits the options available for programming the superelement refinement. Firstly, the original superelement number scheme can remain unchanged and the substructure numbering changed to effectively create one substructure for each superelement. This has the advantage of allowing the load refinement process to remain unchanged and the refinement to proceed along the input file. Secondly, the superelement numbering may change to exploit any repeated substructures which remain after refinement.

The two options for removal of the limitation that one substructure defines one superelement are best presented by an example. The example to be considered involves a model comprising of 6 superelements generated from 2 substructures for which there is a requirement to refine superelements 2 and 4 (see Table 6.1). The new substructure numbering scheme resulting from changes in the substructure numbers is shown in Table 6.2. This procedure would result in 6 substructures when 4 would be sufficient since substructures 1, 3, and 5 are the same. The renumbering of the superelements only results in the minimum of 4 substructures as shown in Table 6.3. However, this procedure would require reordering the load data as well as the superelement data. The geometry data for all the substructures to be refined would have to be written to a file so that the refinement process could proceed sequentially along the file.

6.4 Refinement of Superelements in Arbitrary Orders

The restriction that the substructures must be refined in ascending order also still applies. The present algorithm proceeds along the input file with no rewinds which is necessary to keep the I/O at a minimum. The execution time of program REFINE appears to be greatly affected by I/O.

TABLE 6.1

Sample Model with Six Superelements Generated from Two Substructures

Superelement Number	Substructure Number
1	1
2*	1
3	1
4*	1
5	1
6	2

* Superelement to be refined.

TABLE 6.2

Superelement Refinement with all Superelements Associated with Independent Substructures

Superelement Number		Substructure Number	
Old	New	Old	New
1	1	1	1
2*	2	1	2
3	3	1	3
4*	4	1	4
5	5	1	5
6	6	2	6

* Superelement to be refined.

TABLE 6.3

Superelement Refinement with Substructure Reordering to Minimize Substructures in Model

Superelement Number		Substructure Number	
Old	New	Old	New
1	1	1	1
3	2	1	1
5	3	1	1
6	4	1	2
2*	5	1	3
4*	6	2	4

* Superelement to be refined.

CHAPTER 7
ADDITIONAL DEVELOPMENTS IN PROGRAM REFINE

7.1 Introduction

A number of additional changes to program REFINE, not explicitly called for in the contract statement of work but determined to be desirable during development work and in application of the program, were also implemented. These changes are described briefly in the following sections.

7.2 Option for Data Input from Separate Files

The REFINE program was modified to optionally accept the following data on separate files:

1) Geometry Data from file PREFX.GOM
2) Superelement Data from file PREFX.SED
3) Stiffness Modification Data from file PREFX.SMD
4) Load Data from file PREFX.LOD

The REFINE program still places all the output data in the PREFN.USE file and not in separate files.

7.3 Arrays Dimensioning in the Main Module

The changes required to achieve the option for data input from separate files in the REFINE program resulted in some more general organizational changes in the program. For instance, array data transfers within the program were modified so that all arrays are now dimensioned in the main program and passed through the argument lists of the various subroutines. This change facilitates any future increases in array dimensions to accommodate large scale problems. The restrictions adopted for array sizes are as follows:

NGNMAX = 999 (number of geometric nodes)
NMNMAX = 999 (number of master nodes)
NTIMAX = 999 (number of time steps)
NERMAX = 250* (number of elements to be refined)
NEGMAX = 250* (number of element groups)
NDDMAX = 999 (number of dependent nodes)
NIDMAX = 8* (number of independent nodes)
NDFMAX = 6* (number of dependent degrees of freedom)
NSKCMAX = 100 (number of skewed systems)

All of the above array sizes except those marked with an asterisk correspond with VAST program limitations.

7.4 Superelement Load Data Format Change

The REFINE program was modified to accommodate the new format in VAST for load data on superelements which involves the card identifying the number of the superelement to be loaded being preceeded by an asterisk (*). The implied changes for the REFINE program were concentrated in the subroutine RUSE, which reads from the specified input file and optionally writes it to the specified output file according to the value of the ISW parameter. This subroutine previously terminated the data transfer process and returned control to the main program when either a "I", "R" or "*" was found in the first column. The asterisk previously only identified the beginning of the geometry for a new substructure. The ISW parameter was modified so that "0" indicates that the data is to be read only, "1" indicates that the data is to be read and written, and "3" indicates that the asterisk is to be ignored as a return parameter.

7.5 Identification of Nodes Requiring Multipoint Constraint Equations

The procedure previously employed by REFINE to determine whether a newly generated nodal point requires a constraint or alternately, whether the constraint on a node can be removed was described in detail in Appendix E of Reference 1. The only check to be performed by subroutine MPC had been to determine if any independent nodes of a constraint equation were themselves dependent nodes. If they were, then the constraint equation was assumed to remain. This could result in the retention of unnecessary constraints as will be demonstrated in an example problem below.

An unrefined model for a square plate composed of 18 VAST triangular plate bending elements (IEC=4) is considered. The central element is first refined to order 2 and then the resulting four elements are refined again to order 2. The final mesh is shown in Figure 7.1. The refinement of the first element results in new nodes 17, 18 and 19 which are all border nodes and have constraints generated for them. The refinement of the first of the new elements (connectivity of 6, 17, 18) results in the new border nodes 20, 21 and 22 with constraints generated for them. Similarly, the refinement of the remaining new elements produces the new border nodes 23 to 28. It is clear that although the constraint equations on nodes 22, 23 and 24 would be retained as a result of their independent nodes being themselves dependent, the constraint equations are in fact not necessary.

As explained in detail in Chapter 4, this algorithmic deficiency has now been corrected. The REFINE program was modified to identify and remove the constraint on these nodes by using the concept of an "interface" node. An interface node has been defined to be a node whose constraint equation is common to two or more parent elements. The additional qualification on the check for constraint equation retention is that if dependent nodes are found in the constraint equation and the node is not an interface node, the constraint should remain.

Returning to our example problem, it can be readily demonstrated that the checks for constraint equation retention is now reliable. As before, the refinement of the centre element produces new nodes 17, 18 and 19 which as border nodes have constraint equation generated for them. Again the refinement of the first of the new elements produces new border nodes 20, 21 and 22 with appropriate constraint equations generated. The refinement of the second new element results in new border nodes 23 and 24 having constraints generated. Node 22, however, is specified as an "interface" node since its independent basis is contained in more than one element. The refinement of the third and fourth new elements results in new border nodes 25, 26, 27 and 28 having constraints generated and 23 and 24 being flagged as "interface" nodes. (It is important to note that the logic for recursive refinement of finite element models is reliable only if no restarts are involved. Sufficient information is not stored for these operations to be carried out reliably if model refinement is restarted.)

The element refinement specifications are complete and now subroutine MPC is used to determine which constraints are required. Constrained nodes 17, 18 and 19 remain since the comparison against the connectivity list of the independent basis determine it to be found. Constrained nodes 20, 21, 25, 26, 27 and 28 remain since the independent basis of these nodes contain dependent nodes and these nodes are not interface nodes. The constraints on nodes 22, 23 and 24 were removed since the nodes are identified as interface nodes and have dependent nodes in their constraint equations.

7.6 Graphical Boundary Condition Selection

When program REFINE is used to refine a finite element model, new nodes are inserted between the old ones and any regularity of node numbering is destroyed. Without extensive experience in the operation of the program, the user will not be able to readily establish node numbers on selected portions of the structure as required for instance when defining boundary conditions. Graphical aids of some sort are therefore desirable, the minimum being an option to plot the refined model or a portion of it and to label the node numbers. A capability which is somewhat more sophisticated and automated has been developed within REFINE in a previous contract [1]. It utilizes the graphics cursor capabilities of the VASTG graphics program PLOTV1.

The graphical node point selection capability is optionally activated when defining boundary conditions after specifications for the refinement of a substructured or unsubstructured finite element model have been completed. Within the current contract, the subroutine BCOND in REFINE program has been extended to retain boundary conditions assigned to the original model and in the case of a substructured model to ensure that a node having boundary conditions applied becomes a master node (if it is not already a master node).

The operation of the program to define problem boundary conditions is described in detail in Appendix D of reference [1]. Basically, the REFINE program transfers control to the VASTG graphics program PLOTV1 to activate graphical node selection. This selection is performed by placing a "window" around the nodes of interest. However, if the boundary conditions are to be specified along a curved edge then this becomes a tedious procedure. A user may therefore define the boundary conditions using the VASTBC [2] program after the model has been refined. VASTBC provides the user with the option to select nodal points individually using the graphics cursor or multiply using the graphics cursor to define a window in much the same way that the capability in REFINE based on the program PLOTV1 does.

7.7 Treatment of Skewed Coordinate Systems

When boundary conditions are associated with spatial directions not aligned with global coordinate directions, a skewed coordinate system can be utilized to align model degrees of freedom with directions appropriate for the definition of constraints. Such a capability has been implemented within VAST [3] and thus it is desirable that the effect of defining skewed coordinate systems be completely taken into effect when refining a model. The REFINE program previously had the capability to only transfer the skewed coordinate system data from the original model to the refined model. In this contract, REFINE was further enhanced to account for the effect of the skewed coordinate system in the constraint equation definitions.

7.8 Improved Element Selection Capability

The REFINE program has been modified to provide the user the option to select elements graphically or through interactive prompting at each stage of the element refinement specifications. The program previously allowed the user only once to define the option for the selection of elements for refinement either by graphical means or through interactive prompting. After the option was defined, all subsequent element selections would have to utilize the same selection option.

The capability to select elements graphically is described in Appendix D of Reference 1. A file PREFX.ELM with the elements sorted in ascending order with respect to first the superelement number, next the element group number, and finally the individual element number, is produced. The organization of data in the PREFX.ELM file in ascending order proved extremely beneficial. In REFINE, it was possible to improve the program efficiency (to decrease CPU requirements) and to reduce the amount of I/O. This improvement is only realized in the cases where the window contains several element groups. The I/O reductions (and associated time savings) were accomplished by skipping two sections of the REFINE main program; firstly, the one that copies remaining element group data to the output file and secondly, the section that reverses the input and output file unit numbers before proceeding to refine another element group. The refinement proceeds sequentially through elements in the input file as directed by the PREFX.ELM file until all elements are refined. The two steps which were skipped were generally necessary when selecting elements via prompting because each time the user was prompted for an element group to be refined, the reply could be an arbitrary group number and could therefore involve data at an arbitrary point on the USE file.

The procedure now employed to specify the elements through interactive prompting results in the generation of a PREFX.ELM file which is the same format as the file created when the elements are selected graphically. The refinement proceeds sequentially through the elements identified on the PREFX.ELM file and therefore the unnecessary data handling is eliminated.

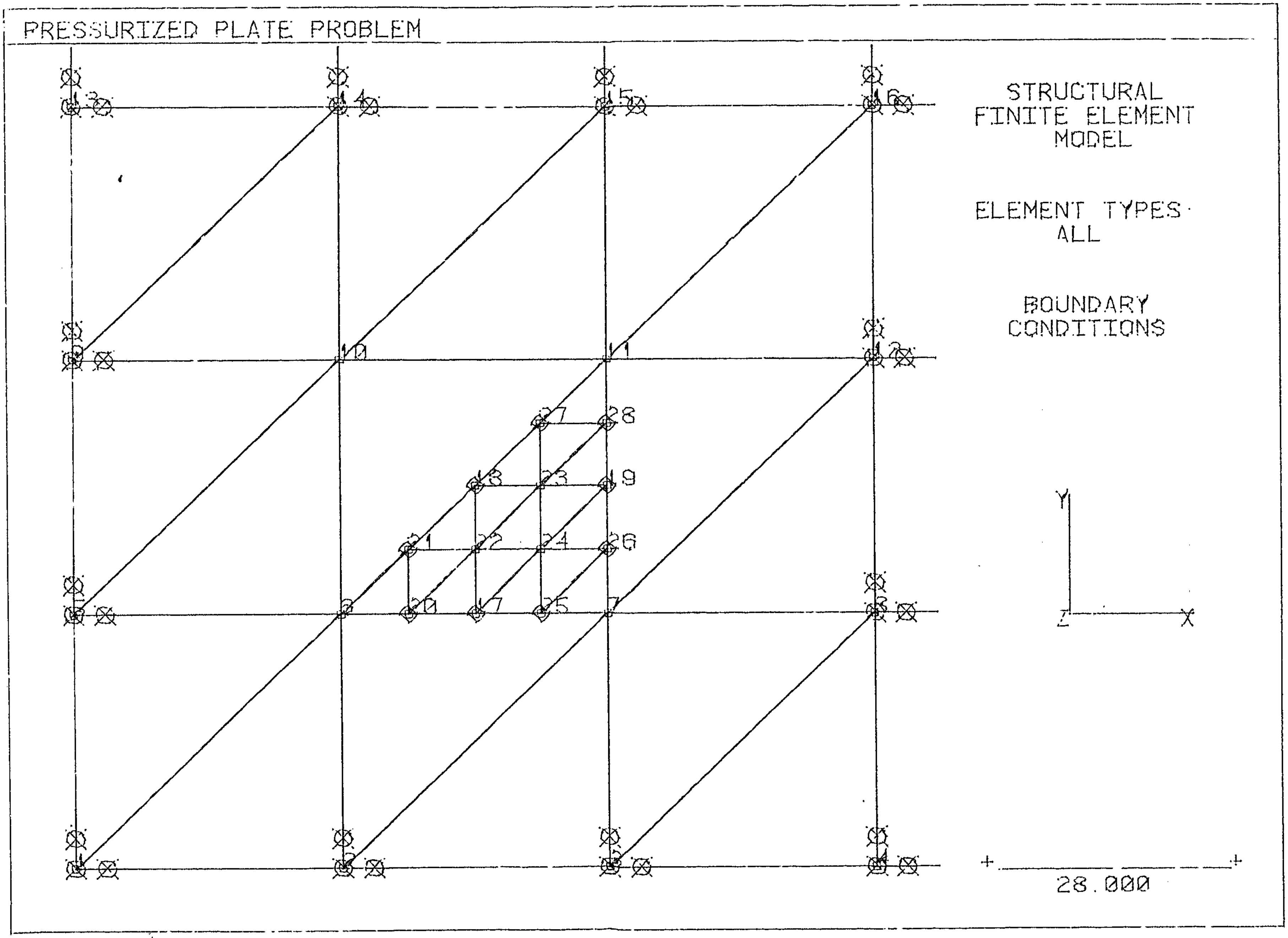


FIGURE 7.1: Recursively Refined Square Plate Model (◇= Dependent Nodes)

REFERENCES

1. M.W. CHERNUKA, M.F. PALMETER, R.A. RICCIUTI and G.L. NIGH. Finite Element Modelling and Optimization, Phase II. Final Report, Martec Limited, Halifax, Nova Scotia (May 1986).

2. M.E. NORWOOD, M.W. CHERNUKA, D.F. FANNING, M.F. PALMETER, E.J. CROCKER and C.G. DESROCHERS. Simplified Models for Ship Structural Analysis. Martec Technical Report TR-93-01. Martec Limited, Halifax, Nova Scotia (December 1992).

3. Vibration and Strength Analysis Program (VAST): User's Manual - Version #04. Martec Limited (January 1986).

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Martec Limited
1888 Brunswick St
Suite 400
Halifax, Nova Scotia B3J 3J8

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Modifications to the Finite Element Modeller Program REFINE

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

Palmeter, M. F.; Chernuka, M.W.

5. DATE OF PUBLICATION (month and year of publication of document)

August 1990

6a. NO OF PAGES (total containing information Include Annexes, Appendices, etc).

6b. NO. OF REFS (total cited in document)

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

DREA Contractor Report

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address).

Defence Research Establishment Atlantic
Grove Street
Dartmouth, Nova Scotia B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

W7707-6-7749

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document).

R-90/91-11

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor).

CR/92/462

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
( ) Distribution limited to defence departments and defence contractors; further distribution only as approved
( ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
( ) Distribution limited to government departments and agencies; further distribution only as approved
( ) Distribution limited to defence departments; further distribution only as approved
( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected).

UNCLASSIFIED
SECURITY CLASSIFICATION OF FORM

DCD03 2/06/87-M

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

Program REFINE was initially developed as part of an adaptive finite element modelling package for VAST where from estimates of discretization errors based on element residuals, the finite element model would be selectively refined (by element subdivision) until acceptable levels of accuracy (according to computed error estimates) was attained [1]. However, increasingly the REFINE program finds application as a general modelling tool which complements the data generation capabilities of PVAST and HVAST. Under these circumstances, the PVAST and HVAST programs need only concentrate on achieving accurate geometric models. Since subsequent refinement of the model to attain acceptable finite element accuracy can be achieved with REFINE. This is particularly useful when the initial model is defined in a pointwise fashion with a graphics tablet. In this context as well, selective model refinement would generally be employed but the experience of the modeller would dictate where the model refinements would be.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

ships
structures
finite element
grid refinement

#151394

NO. OF COPIES NOMBRE DE COPIES	COPY NO. COPIE N°	INFORMATION SCIENTIST'S INITIALS INITIALES DE L'AGENT D'INFORMATION SCIENTIFIQUE
1	1	CML
AQUISITION ROUTE FOURNI PAR ▶	DREA	
DATE ▶	04 MAY 1995	
DSIS ACCESSION NO. NUMÉRO DSIS ▶		

DND 1158 (6-87)

National Defence Défense nationale

PLEASE RETURN THIS DOCUMENT TO THE FOLLOWING ADDRESS:

DIRECTOR
SCIENTIFIC INFORMATION SERVICES
NATIONAL DEFENCE
HEADQUARTERS
OTTAWA, ONT. - CANADA K1A 0K2

PRIÈRE DE RETOURNER CE DOCUMENT À L'ADRESSE SUIVANTE:

DIRECTEUR
SERVICES D'INFORMATION SCIENTIFIQUES
QUARTIER GÉNÉRAL
DE LA DÉFENSE NATIONALE
OTTAWA, ONT. - CANADA K1A 0K2