Fox

AEDC-TR-91-8

PEGSUS 4.0 User's Manual

N. E. Suhs and R. W. Tramel
Calspan Corporation/AEDC Operations

November 1991

Final Report for Period March 1, 1990 through June 30, 1991

ARNOLD ENGINEERING DEVELOPMENT CENTER
ARNOLD AIR FORCE BASE, TENNESSEE
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE

NOTICES

When U. S. Government drawings, specifications, or other data are used for any purpose other than a definitely related Government procurement operation, the Government thereby incurs no responsibility nor any obligation whatsoever, and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise, or in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified users may obtain copies of this report from the Defense Technical Information Center.

References to named commercial products in this report are not to be considered in any sense as an endorsement of the product by the United States Air Force or the Government.

This report has been reviewed by the Office of Public Affairs (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

APPROVAL STATEMENT

This report has been reviewed and approved.

Murray O King

MURRAY O. KING
Aeronautical Systems Division
Directorate of Aerospace Flight Dynamics Test
Deputy for Operations

Approved for publication:

FOR THE COMMANDER

Russell B. Sorrells

RUSSELL B. SORRELLS
DOF Analysis Manager
Directorate of Flight Dynamics Test
Deputy for Operations

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1991	Final - March 1, 1990 - June 30, 1991

4. TITLE AND SUBTITLE

PEGSUS 4.0 User's Manual

5. FUNDING NUMBERS

PN CP52

6. AUTHOR(S)

Suhs, N. E. and Tramel, R. W.
Calspan Corporation/AEDC Operations

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Arnold Engineering Development Center/DOF
Air Force Systems Command
Arnold Air Force Base, TN 37389-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

AEDC-TR-91-8

9 SPONSORING/MONITORING AGENCY NAMES(S) AND ADDRESS(ES)

Air Force Systems Command/SKE
Eglin Air Force Base, FL 32542

10 SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

Available in Defense Technical Information Center (DTIC).

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

Many aerodynamic configurations are too complex to be represented by a single computational mesh. A multiple-mesh domain decomposition method called chimera has been developed to allow complex aerodynamic configurations to be modeled by a system of individually generated meshes, each representing a component of the overall configuration. The chimera approach requires the use of two computer codes, PEGSUS and a flow solver. PEGSUS calculates the interactions among the meshes and produces a file consisting of interpolation information that is used by the flow solver to obtain a global flow-field solution for the entire mesh system. This report documents the design and use of the latest version (4.0) of PEGSUS, and describes techniques for proper modeling of complex aerodynamic configurations.

14 SUBJECT TERMS		15. NUMBER OF PAGES
computational fluid dynamics domain decomposition techniques PEGSUS 4.0	chimera grid embedding techniques	234
		16. PRICE CODE

17 SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAME AS REPORT

PREFACE

The work reported herein was conducted by the Arnold Engineering Development Center (AEDC), Air Force Systems Command (AFSC), under AEDC Project Number CP52PW (Program Element 52790F) at the request of the AFSC/SKE, Eglin AFB, FL 32542. It was produced by Calspan Corporation/AEDC Operations, an Air Force contractor operating the aerospace flight dynamics testing program at AEDC, Arnold Air Force Base, TN 37389. The SKE Project Manager was Capt. D. Smith, and the AEDC/DOFA Project Manager was Mr. M. O. King. D. R. Carlson was the Calspan Project Manager. The work was conducted during the period March 1, 1990 through June 30, 1991, and the manuscript was submitted for publication September 26, 1991.

The authors acknowledge the contributions of Dr. J. C. Erickson, Jr., for his editorial assistance during the preparation of this report. Also, the authors acknowledge the assistance provided during the design and testing of the the code from W. E. Dietz, T. L. Donegan, M. J. Rist, Dr. J. L. Jacocks, and J. K. Jordan.

CONTENTS

ILLUSTRATIONS

Figure Page

Figure Page

APPENDIXES

1.0 INTRODUCTION

Computational fluid dynamics (CFD) calculations to support ground testing are an important aspect of the AEDC mission. Aerodynamic problems related to wind tunnel testing present two important challenges to CFD: the need to produce timely solutions in response to testing requirements, and the solution of problems related to complex three-dimensional configurations. The main obstacle to overcoming these challenges is the difficulty of generating computational meshes. Although the time required to generate meshes is decreasing with the development of interactive grid generation software (Ref. 1), it is still desirable to keep the region that requires a mesh to a simple topology that is easy to modify. In addition, a single mesh that is fine enough to resolve the desired aerodynamic features of a complex flow field may be too large for available computer memory.

To address these problems, Benek, et al. (Refs. 2 – 5) have developed a method of domain decomposition called "chimera" which allows a system of relatively simple grids, each describing a component of a complex aerodynamic configuration, to be combined into a composite grid to yield solutions for complex flow fields. The chimera scheme has greatly increased the range of aerodynamic problems that can be addressed by CFD, and has been applied to store separation (Refs. 6 – 8), cavity flow (Ref. 9), transonic tunnel wall interference calculations (Refs. 10 – 11), space shuttle ascent (Ref. 12), dynamic body movement (Ref. 13), and flow in an artificial heart (Ref. 14).

The general concept behind chimera is illustrated in Fig. 1, which depicts two independently generated meshes representing a flapped airfoil. The flap mesh is embedded within the airfoil mesh. Clearly, the flap mesh outer boundary can receive flow-field information interpolated from appropriate mesh points (often referred to as interpolation stencils) of the airfoil mesh. However, a reverse process must occur as well; the airfoil mesh must receive flow-field information from the flap mesh. An artificial boundary must be defined within the airfoil mesh, since certain points within the airfoil mesh are interior to the flap and thus do not lie within the airfoil/flap flow field. The artificial boundary points of the airfoil mesh that are fully contained within the computational region of the flap mesh can be updated by interpolation from the appropriate mesh points of the flap mesh. Generally, any mesh can receive information from other meshes through outer boundary and artificial boundary points.

The interpolation process is further illustrated in Fig. 2, which depicts a portion of the overlap region between the airfoil and flap meshes. Airfoil mesh points that are contained within a certain region surrounding the flap are excluded from the computational domain of the airfoil mesh. (In chimera terminology, they are "blanked" points, or hole points.) The exclusion of points is accomplished by defining a hole creation boundary within the flap mesh that will define the region within which all airfoil points are to be blanked. The points

in the airfoil mesh surrounding the blanked points are hole boundary points, and they receive flow-field information interpolated from mesh points within the flap mesh. Correspondingly, points on the outer boundary of the flap mesh receive flow-field information interpolated from mesh points within the airfoil mesh.

Application of the chimera scheme requires two steps: (1) a description of how each mesh is to communicate flow-field information to other meshes, and (2) execution of a flow solver that uses the communication information generated in step 1. Step 1 is performed by the code named PEGSUS. The processes accomplished by PEGSUS include establishing which boundary points in a mesh must be updated by interpolated flow variables from other meshes and by calculating the required interpolation coefficients for the mesh points that send information to an interpolated boundary point in another mesh. The relationship of PEGSUS to the chimera domain decomposition scheme is depicted schematically in Fig. 3. The individual meshes and user-defined mesh connection data are input into PEGSUS. PEGSUS produces (1) a composite mesh, i.e., a single file consisting of the concatenation of all meshes in the multiple-mesh configuration, and (2) an interpolation file, which is a table associating all interpolated boundary points in the composite mesh with mesh points that supply the interpolated flow-field values. In step 2, the composite mesh and interpolation file are input into the flow solver. The flow solver can be any finite-difference or finite-volume code that has been modified for the chimera interpolation data structure. Several compressible flow codes, namely XMER3D (Ref. 5), F3D (Ref. 12), XLIM3D (Ref. 15), and EAGLE (Ref. 16), and an incompressible flow code (Ref. 14) can currently use the chimera scheme.

The chimera method has evolved in terms of its capabilities and applicability to complex three-dimensional flow configurations. Much of the evolution has been manifested in several versions of the PEGSUS program. PEGSUS 1.x required the definition of a hierarchical mesh structure which restricted which meshes could embed solid surfaces in other meshes. PEGSUS 2.x allowed much more generality in relationships among meshes by eliminating any hierarchical relationships. However, PEGSUS 2.x was limited in the generality in which mesh-to-mesh communications could be defined, was restricted to simple hole boundary shapes, and required different mesh topologies to be treated as special cases. PEGSUS 3.x (Refs. 17 - 18) added several enhanced features that increased the flexibility and applicability of the code to multiple mesh problems. Most importantly, PEGSUS 3.x provided greater flexibility in the definition of hole boundaries and mesh interactions. Complex mesh interactions and hole boundary definitions were handled through input, usually without requiring program modification.

PEGSUS 4.0 is the latest version of the PEGSUS series of mesh interpolation codes. It is a fully three-dimensional code. PEGSUS 4.0 adds further flexibility to the user inputs and control of the overall process. Major features are as follows:

— Meshes, boundaries, and/or regions can be added and/or subtracted using the restart capability

— Single or double level hole boundaries can be specified

— Hole boundary points on a reflection plane can be ignored

— The restrictions for obtaining a valid interpolation stencil can be relaxed

— Holes can be defined using a number of methods

— Regions can be directly blanked for interior boundaries

— The code has been written to be portable (The code has been successfully run on Apollo and Silicon Graphics workstations and on a Cray supercomputer).

A glossary of PEGSUS terminology is provided in Appendix A. Appendixes B through F contain information about PEGSUS program variables, program structure, subroutines, direct access files, and output file structure, respectively. Appendixes G and H contain complex aerodynamic configuration examples and sample Cray job control language required for the examples, respectively.

The purpose of PEGSUS 4.0 is identical to that of earlier versions, namely to produce a composite mesh and interpolation file for the flow solver in question. To produce the required inputs for the flow solver, PEGSUS 4.0 has four basic steps to execute. These steps are: (1) process the meshes and user inputs, (2) identify the hole and interpolation boundary points, (3) determine the interpolation stencil and interpolation coefficients for each interpolation boundary point, and (4) supply diagnostic information on the execution and output the results for input to the flow solver.

The details of PEGSUS 4.0 are given in the subsequent sections. Section 2.0 describes the overall approach adopted by PEGSUS 4.0. The methods used to identify holes and interpolation boundaries are described in Sec. 2.1, whereas the intricacies of the interpolation process and the different controls that can be placed on this process by user inputs are described in Sec. 2.2. Section 2.3 concludes the description of the overall approach with a discussion of the additional features available for the control of PEGSUS results. Section 3.0 describes the implementation of PEGSUS 4.0. The program parameters are described in Sec. 3.1. Section 3.2 contains a description of the user inputs referred to earlier. In Sec. 3.3, the format for the input meshes is given. The files output by PEGSUS 4.0 are described in Sec. 3.4. Finally, six examples demonstrating many of the capabilities of PEGSUS 4.0 are given in Sec. 4.0.

Since the descriptions in Secs. 2.0 and 3.0 are given in relatively general and abstract terms, careful study of the examples in Sec. 4.0 and Appendix G will aid the reader in acquiring a working knowledge of the concepts and their implementation.

2.0 GENERAL APPROACH

This section discusses the procedures used to determine which points in a mesh will require interpolation and how data for these points are interpolated. Also presented are the special features of PEGSUS 4.0 that allow additional control of the results. A glossary of terms is given in Appendix A to assist the reader in grasping the concepts and their practical implementation.

2.1 INTERPOLATION BOUNDARY POINT SPECIFICATION

There are two ways, hole boundaries and outer boundaries, in which information is transferred from one mesh to another. The user must identify these interpolation boundaries so that PEGSUS can find the interpolation stencil and calculate interpolation coefficients for each boundary point. The identification of interpolation boundaries is made through the user inputs $BOUNDARY, $SURFACE, $BOX, $REGION, and $VOLUME (see Sec. 3.2 for format details). These user inputs are supplied in a NAMELIST-like format that is read by an interpreter within PEGSUS and thus is not restricted to any particular FORTRAN compiler (since NAMELIST is not standard FORTRAN 77). The term NAMELIST that is used throughout this report refers to this modified NAMELIST-like format. Using these inputs, three methods for specifying hole boundaries and one for specifying outer boundaries are defined.

2.1.1 Blanking

Before considering the different methods for identifying interpolation boundary points, the definition of blanking must be understood. The identification of a point as a hole point, an interpolation boundary point, or a field point must be passed on to the flow solver so that the point will be handled correctly. The status of each point is contained within an integer array called IBLANK. The IBLANK array, which is dimensioned identically to the number of points in each mesh, has a value for each point within each computational mesh. The values contained within the IBLANK array are either 0 for a point that is a hole or interpolation boundary point, or 1 for a point that is a field point. Within the flow solver, points with IBLANK = 0 are either considered to be completely out of the computational domain or are to be updated by interpolation, while points with IBLANK = 1 are updated by the solution algorithm and associated boundary conditions. The IBLANK array is supplied to the flow

solver through the interpolation file output of PEGSUS (see Appendix F). References 2 and 4 describe how the IBLANK array is incorporated into the flow solution algorithm.

2.1.2 Hole Boundary Specification

Three methods for creating holes are presented. Two of these methods use an indirect means (not specifying the points in the hole directly), while the third method directly specifies the points that make up the hole. The first method uses the $SURFACE and $BOUNDARY user inputs to indirectly specify hole locations. The user input for this method is defined in the computational domain but utilizes surface normals in the physical domain to determine the hole points. The second method, also for indirect hole specification, uses the $BOX and $BOUNDARY user inputs. These input sets describe the hole in the physical domain and employ the physical domain to find the hole points. The third method, which is a direct means for defining holes, uses the $VOLUME and $REGION user inputs. These input sets define the hole in the computational domain and are applied in the computational domain to find hole points. The two indirect methods are described in Sec. 2.1.2.1 while the direct method is described in Sec. 2.1.2.2. Once the holes are found, a hole boundary is placed around each hole. A description of the method used to determine hole boundaries is given in Sec. 2.1.2.3. Finally, in Sec. 2.1.2.4 special problems which occur in the hole creation process are presented along with their solutions.

2.1.2.1 Indirect Hole Specification

Typically, a hole is defined through indirect means, since the actual points within the hole are not known beforehand, and the actual hole boundary is irregular in shape (not a continuous J, K, or L plane, where the grid points in each mesh are indexed by separable indices J, K, and L). The first method uses the $SURFACE user input (identical to the method in PEGSUS 3.x) and requires the use of a surface or group of surfaces from one mesh to create a hole in another mesh. An example of this method is given in Sec. 4.1.1. A hole creation boundary consists of a surface or a group of surfaces. The purpose of a hole creation boundary is to identify points that are within this boundary. A mesh point is considered to be inside a hole creation boundary if it is inside all surfaces that define the boundary. The method used to determine whether a point is inside or outside a surface is illustrated in Fig. 4. A mesh point (P) is considered to be inside a surface if the dot product between $\vec{R}$ (the vector from the closest point on the surface to the mesh point), and $\vec{N}$ (the normal vector on the surface at the closest point, directed outward from the hole region) is negative or zero. If the dot product is positive, the mesh point is considered to be outside the surface.

The hole location process for a triangular hole creation boundary consisting of the three surfaces S1, S2, and S3 is illustrated in Fig. 5. PEGSUS first puts the indices of all points

of the mesh in which the hole is to be created into a list. Then, all points that fall outside surface S1 are eliminated from the index list (as indicated by the shaded region in Fig. 5a). Then all points contained in the shortened list that are outside the surface S2 (Fig. 5b) are eliminated, which shortens the index list further. When the process is repeated for surface S3 (Fig. 5c), the points remaining in the list are the hole points of the mesh in which the hole is created.

This method of finding hole points has one important restriction, which is illustrated in Fig. 6. In Fig. 6a, a concave hole creation boundary is defined. The candidate point P will be found to lie outside Surface 1, and therefore outside the entire hole boundary. As a result, the point will not be identified as a hole point, which clearly is an error. Generally, hole creation boundaries (as viewed from the outside) must be convex to obtain correct results. However, PEGSUS 4.0 allows holes to be defined by multiple hole creation boundaries. Any hole creation boundary that contains concavities can be divided into one or more entirely convex regions, each of which can create holes in one or more meshes. If the hole creation boundary is redefined as two separate convex boundaries, as in Fig. 6b, the candidate point will be inside one of the hole creation boundaries, and will therefore be correctly identified as a hole point. The candidate points within the two boundaries will all be labeled as hole points and will therefore merge into a single hole.

Now that the method for defining a hole using surfaces has been described, the relationship of a surface or group of surfaces to a hole creation boundary and how this hole creation boundary relates to meshes must be described. The relationship between a hole creation boundary and a mesh is particularly important since the hole creation boundary must have a connection to the mesh that contains the surfaces defining the hole as well as a connection to the the mesh or meshes that will have a hole made in them. The connections between hole creation boundaries and meshes are depicted in Fig. 7. Boundaries are defined as collections of level surfaces (i.e., surfaces that share a common J, K, or L value) within meshes. Shown in Fig. 7 are H-Boundary a and H-Boundary b, which are hole creation boundaries defined by Mesh n and Mesh n+2, respectively. In NAMELIST nomenclature, H-Boundary a "ISPARTOF" (IS PART OF) Mesh n and H-Boundary b "ISPARTOF" Mesh n+2. Hole creation boundaries also have relationships to other meshes, i.e., they make holes in other meshes (see Fig. 2). This relationship is denoted as "MHOLEIN" (Makes a HOLE IN) in the NAMELIST input. According to Fig. 7, H-Boundary a makes holes in both Mesh m and Mesh n-1 while H-Boundary b makes holes in Mesh n-1, Mesh n, and Mesh n+1. The maximum number of meshes that a boundary can make a hole in is set by the program parameter MHDIM (see Sec. 3.1). The examples in Sec. 4.0 and Appendix G will clarify these relationships.

Figure 8 illustrates the relationship between surfaces and hole creation boundaries. Hole creation boundaries are comprised of collections of surfaces, and therefore, surfaces have "ISPARTOF" relationships to boundaries in much the same way as boundaries do to meshes. Hole creation boundaries are normally, but not always, composed of level surfaces from the meshes to which they are connected. This situation is graphically depicted in Fig. 8 with Surface l being the exception. As was shown in Fig. 7, H-Boundary a is part of (ISPARTOF) Mesh n and, therefore, the two surfaces composing H-Boundary a, namely Surface l and Surface q-2, would normally be connected to Mesh n. In this case, though, one of the connections must be overridden by the input "MESH NAME" (see Sec. 3.2). Surface l has a "MESH NAME" specified for Mesh n-2 in the NAMELIST input (Sec. 3.2) and is therefore composed of a level surface in Mesh n-2. That is, a level surface from another mesh is required to successfully complete the hole creation boundary. An example demonstrating hole creation using surfaces from two separate meshes is given in Sec. 4.1.4. The remaining surfaces are connected to meshes through the ISPARTOF specification for each boundary (See Fig. 7).

So far, the hole creation boundaries that have been demonstrated completely enclose the hole region. The fact that most hole creation boundaries are completely closed has been used to accelerate the process of finding holes (hole point determination was one of the most computationally intensive processes in PEGSUS 3.0). The acceleration is obtained by creating a parallelepiped based on the minimum and maximum X, Y, and Z values of the surfaces specified to create the hole. The minimum/maximum parallelepiped is used to quickly determine if a point can possibly be in the hole. If a point is outside this parallelepiped then it is immediately removed from the list of candidate hole points. This results in a greatly reduced list of candidate hole points. Those points on the inside of the minimum/maximum parallelepiped must still be checked to see if they are actually in the hole or not. As discussed earlier, the final determination of whether a point is in a hole is done with surfaces. The minimum/maximum ranges of the parallelepiped are given in the PEGSUS output for each hole boundary description. However, under certain circumstances when the hole creation boundary does not clearly distinguish between points inside and outside the hole, such as when the user has not closed the hole creation boundary, the minimum/maximum parallelepiped may not produce the hole that the user wanted. To allow for these special cases, the user input "CLOSED" can be specified to be equal to .FALSE. (default is .TRUE.). Figure 9a illustrates the effects of such a specification with the shaded area being the area within the hole. In some instances, for example when the boundary is not closed but intersects an outer plane of the domain, the specification of "CLOSED" equal to .TRUE. or .FALSE. gives the same results (although specifying .FALSE. increases the computational time). This case is demonstrated in Fig. 9b. The "CLOSED" user input is used only as part of the $BOUNDARY user input when a hole creation boundary is defined by a collection of surfaces.

The second method for defining holes, also an indirect method, uses the $BOX user input. The $BOX user input is connected to a boundary and can be used in conjunction with surfaces or by itself. An example that demonstrates the $BOX user input is given in Sec. 4.1.2. A box is described by a minimum-to-maximum range of X, Y, and Z coordinates. This box is used to create holes in those meshes to which the boundary is connected by the MHOLEIN input. For those meshes in question, any mesh point located within the box will be considered to be a hole point. The box-hole creation boundary connection is depicted in Fig. 10. H-Boundary a has three box definitions, Box m, Box n-2, and Box n-1. H-Boundary b has no box connections. H-Boundary c has two box connections, Box n and Box n + 1. Similarly, H-Boundary d has two box connections, Box n + 2 and Box p. Therefore, similar to the surface-hole creation boundary connection, the box-hole creation boundary connection makes holes in meshes specified with the "MHOLEIN" input.

2.1.2.2 Direct Hole Specification

The third method for defining holes, a direct method, is accomplished through the $REGION and $VOLUME user inputs. An example demonstrating this method is given in Sec. 4.1.3. The $REGION user input differs from the $BOUNDARY user input in that it is connected directly to only one mesh, the mesh in which the hole will be located. The ISPARTOF user input specifies this mesh. Typical connections between regions and meshes are shown in Fig. 11. For this example, Region a is connected to and makes a hole in Mesh n-2, and likewise, Region b and Region c are connected to and make holes in Mesh n and Mesh p, respectively.

Specification of the ranges of the J, K, and L indices that will become a hole is accomplished through the $VOLUME user input. In Fig. 12 the volume-region connection is demonstrated. Similar to the surface-boundary connection and box-boundary connection, any number of volumes can be used to create a region (limited by the parameter NVDIM, see Sec. 3.1). For this case, three volumes are connected to Region a, Volume m, Volume n-2, and Volume n-1. Region b and Region c are connected to one volume each, Volume n and Volume n + 1, respectively. Finally, Region d has two volume connections, Volume n + 2 and Volume p.

2.1.2.3 Determining the Hole Boundary (Fringe)

Once all holes in all meshes have been created, mesh points surrounding holes (i.e., hole boundary points, also called fringe points in chimera terminology) must be identified. Hole boundary points are by definition the point or points separating hole points from field points. Field points are points within the computational domain for which data are updated by the flow solver rather than by interpolation from another mesh. To find hole boundary points, each point in a mesh is checked. If the point is a hole point, nothing is done. If the point

is a field point and is next to a hole point, it is identified as a hole boundary point. This procedure results in a hole boundary with only one interpolated boundary point between a hole point and a field point . This type of hole boundary is termed a single fringe boundary and is demonstrated in Fig. 13a. Alternatively, two interpolation points can be specified between a hole point and a field point. This type of hole boundary is termed a double fringe boundary and is demonstrated in Fig. 13b.

When using a single fringe in conjunction with a flow solver which uses 4th-order smoothing, the 4th-order terms require a 5-point difference stencil. The difference stencil must be either converted to 2nd order (3-point difference stencil) or otherwise reduced at the hole boundaries. This reduction to 2nd order is not necessary for a double fringe boundary. Experience has shown that using a double fringe gives better results, particularly in regions of high gradients. However, double fringes require more mesh points and make it more difficult to obtain the required overlap between hole boundaries and outer boundaries. Overlap is defined as those field points lying between interpolation boundary points contained in adjacent meshes.

2.1.2.4 Common Problems in Creating Hole Boundaries

Because of the nature of the algorithms employed in PEGSUS 4.0, undesirable mesh features such as internal discontinuities or extreme skewness may result in erroneous results unless proper care is exercised. Almost all errors resulting from undesirable mesh features will be manifested as spurious hole points. In these cases, a review of the diagnostic maps, described in Sec. 3.4 and Sec. 4.0, will reveal hole points occurring in regions that should not be blanked out. For instance, Fig. 14a depicts a hole creation boundary that has been defined as a single level surface within a mesh. In this case, the level surface contains a discontinuity in slope. PEGSUS 4.0 will attempt to identify points belonging to other meshes that fall inside the hole creation boundary. The hole location procedure will, for the candidate point (P) shown, identify the point on the vertex of the level surface as the closest point to the candidate point, calculate the dot product of the normal at the vertex and the vector defined by the vertex and candidate points, and decide that the candidate is inside the hole creation boundary, because the dot product found in the previous step will be negative. This determination is clearly incorrect; overall, the effect of the discontinuity in the normal vectors will be a "cloud" of hole points surrounding the vertex.

Fortunately, the flexible surface and boundary definitions incorporated into PEGSUS 4.0 allow such problems to be solved by modifying the input. If the hole creation boundary is divided at the discontinuity in slope into two separate surfaces (Fig. 14b), then the candidate point will be found to be inside surface 1 (solid line), outside surface 2 (dashed line), and

therefore, outside the hole creation boundary. It is good practice, in general, to construct hole creation boundaries such that each surface comprising the boundary is smooth.

Another common problem occurs in discontinuous regions of meshes. Figure 15a depicts a hole creation boundary defined as a level surface in an O-mesh in which the edges of the boundary are coincident in the region of the cut. That is, the first and last points circumferentially coincide on the cut. This leads to a discontinuity in slope of the same nature as that shown in Fig. 14. Here again the closest points to a candidate point (P) are the coincident points on the cut. PEGSUS will simply choose one of the points as the closest to the candidate (i.e., the first one it finds). A problem arises if the point chosen is associated with a normal vector pointing away from the candidate point; in this case the candidate will be determined to be inside the hole creation boundary. Typically this problem manifests itself as a funnel-shaped "cloud" of hole points extending from the cut region of the hole creation boundary into the interior of the mesh containing the candidate point. The solution of this problem is similar to that employed in the previous example. If the hole creation boundary is divided into two surfaces (solid line and dashed line), as shown in Fig. 15b, then the candidate point will be found to be outside the hole creation boundary.

Another example is illustrated in Fig. 16a and displays the concavity problem discussed in Sec. 2.1.2.1 and shown in Fig. 6. Two meshes are included in this configuration. The first mesh models a body with an attached fin. The second mesh models a strut attached to the body. The mesh topology of the body/fin mesh is such that the fin and body surfaces lie on level surfaces that are normal to different computational coordinates. It is necessary to blank out points in the strut mesh that lie within the body and fin. Initially, a hole creation boundary is created from the two level surfaces comprising the body and fin. Application of the hole location algorithm will have an undesired effect; all points within the body will be inside Surface 2 and outside Surface 1, whereas all points within the fin will be outside Surface 2 and inside Surface 1. Since a point must be inside all surfaces to be considered inside a boundary comprised of those surfaces, no points will be blanked out.

The solution to this problem is depicted in Fig. 16b, where the original hole creation boundary has been divided into two separate boundaries, each comprised of a single level surface. Now the points inside the fin will be inside Boundary 1, and points inside the body will be inside Boundary 2. Note that the fact that the fin points are outside Boundary 2 has no overall effect, since a point is blanked if it is inside any boundary.

2.1.3 Outer Boundary Specification

Outer boundary points are directly specified by giving the J, K, and L index ranges of the boundary with the user input $SURFACE. The surface or group of surfaces is connected

to a boundary that points to the mesh that is to have these outer boundary points. The user input ISPARTOF points to the mesh that contains the outer boundary points. The outer boundary input is distinguishable from hole boundary input in that no MHOLEIN input is specified. Shown in Fig. 17 are outer boundary-to-mesh connections, namely O-Boundary a and O-Boundary b with the "ISPARTOF" relationship to Mesh n and Mesh p, respectively. Outer boundaries are always composed of level surfaces from the meshes to which the boundaries are connected (see Fig. 18). To specify a double interpolation outer boundary (used when a double fringe is specified), two level surfaces are defined in $SURFACE. Outer boundary specifications are demonstrated in the examples detailed in Sec. 4.1.

2.2 INTERPOLATION

Once all hole boundary and outer boundary points have been identified, they are combined into a list of points from each mesh that require interpolation of flow-field data. At this point, PEGSUS searches other meshes in the composite mesh for interpolation stencils that may be used to interpolate flow-field information to the boundary points requiring data.

2.2.1 LINK Lists

Each mesh in the composite mesh receives information from other meshes in one of two ways: (1) through its outer boundaries, or (2) through hole boundaries. To determine which meshes are to communicate with each other, the user input LINK must be specified within the $MESH user input (see Sec. 3.2). The LINKs are priority lists that indicate the order in which other meshes will be searched for suitable interpolation stencils. A typical LINK for a mesh is shown in Fig. 19. In this case Mesh n will search Mesh n + 1, Mesh p, Mesh n − 1, and Mesh n + 2, in that order, for interpolation stencils that can provide information to Mesh n's interpolation boundary points. Each mesh in the composite mesh will be related to the other meshes by similar linkages. The maximum number of LINKs allowed per mesh is set by the program parameter LNDIM (see Sec. 3.1).

The LINK definitions for each mesh comprise a priority list containing the names of other meshes that are to be searched for interpolation stencils that can provide information for boundary points. The meshes in the priority list are searched in the order in which they appear in the input. If an interpolation stencil is found, then the interpolation information for the point is stored in a set of arrays. If the boundary points cannot be interpolated by a stencil in the first mesh in the priority list, then the second mesh is searched for interpolation stencils. The process continues until either interpolation stencils for all boundary points have been found, or the mesh priority list is exhausted. Any boundary points that remain after the mesh priority list is exhausted are termed "orphans." Orphaned boundary points are treated as hole points (IBLANK = 0). The user must decide whether to accept the orphan points and

update them in the flow solver (e.g., average them from nearby points) or to change the meshes to eliminate the orphan points.

To reduce the computational time required for an execution of PEGSUS 4.0, the LINK list for each mesh is preprocessed before being used as the interpolation priority list. First, a parallelepiped formed by the minimum and maximum X, Y, and Z values of each mesh is created. Using the minimum/maximum parallelepiped for each mesh, each LINK input of a mesh is checked to determine if it intersects with the mesh. If the LINKed mesh does not intersect with the mesh, the LINKed mesh is dropped from the mesh's LINK list. For example, if 'MESH B' is a LINK of 'MESH A' and the parallelepiped of 'MESH A' does not intersect with the parallelepiped of 'MESH B', then 'MESH B' is dropped from the LINK list of 'MESH A'.

Since hole location and boundary point interpolation occur as two separate functions, a mesh that causes a hole in another mesh does not necessarily have to be the mesh that provides interpolated information to the resultant boundary points. This allows meshes to be used for no other function than to generate holes in other meshes; this feature is used extensively in certain instances, such as the representation of a fuselage and attached wing (see Sec. 2.3.3).

2.2.2 Stencil Searching

Identifying interpolation stencils, i.e., mesh points that can be used to supply interpolated information to boundary points, requires a search procedure. An exhaustive search of a mesh will always find a permissible interpolation stencil if one exists; however, exhaustive searches are computationally intensive. PEGSUS uses a heuristic search mechanism that generally can find an interpolation stencil in relatively few steps.

The steps required for obtaining valid interpolation points (or a stencil) for a candidate boundary point are described below. Briefly, the steps are: (1) find a point in the LINKed mesh near the candidate boundary point, (2) find a stencil with satisfactory interpolation coefficients, and (3) check that the stencil is valid. A candidate boundary point is given three attempts at steps 2 and 3. The first attempt uses the stencil found for the previous candidate boundary point as the near point. The second and third attempts use subsets of the LINKed mesh to find a near point. The near point is determined by finding the smallest distance between the candidate point and a point in the subset mesh. The first subset mesh is every fourth point in each of the J, K, and L directions. The second subset mesh is also every fourth point but shifted by two points in each of the J, K, and L directions. Once a near point has been found, the "stencil-walking" procedure, described in the next paragraph, is performed on the candidate stencil.

The interpolation algorithm used in PEGSUS is a trilinear interpolation scheme, and is fully documented in Ref. 4. The trilinear interpolation algorithm yields interpolation coefficients for a point either inside or outside the trial stencil (the latter case is actually an extrapolation). The interpolation coefficients will all be between 0.0 and 1.0 if the point is inside the stencil; a point outside the stencil will cause at least one coefficient to be outside the range of 0.0 to 1.0. The interpolation stencil in effect defines a local coordinate system, while the interpolation coefficients define the location of the point in the local system. In PEGSUS, the interpolation coefficients are used to determine the direction (in computational coordinates) in which the search should proceed to eventually find a stencil that surrounds the point. For instance, a given boundary point and a stencil with lowest index corner point at J, K, and L may yield interpolation coefficients of 5.2, −2.8, and 1.3. Pegsus will then move to the stencil corresponding to the point J+2, K−2, and L+1. (Note that the J index is only changed by 2 since the procedure limits the "walking" to 2 indices during the first 10 iterations and 1 index for the final 10 iterations.) The new stencil will usually be closer to the boundary point. The procedure is repeated until all interpolation coefficients are between 0.0 and 1.0, which is considered a satisfactory interpolation. Although this searching mechanism can, in principle, fail in certain cases (e.g., highly curved or distorted meshes), it is highly reliable in practice. This search mechanism is referred to in chimera terminology as "stencil walking." The method of "stencil walking" in conjunction with finding a near point in the subset mesh was found to be much more computationally efficient than the "stencil-jumping" procedure used in PEGSUS 3.x (Ref. 17).

Once a satisfactory interpolation has been found, the stencil from which this interpolation is made must be checked to determine if the stencil is valid. The difference between valid and invalid interpolations is illustrated in Fig. 20. In the illustration, the outer boundary of a mesh (dotted lines) is to receive interpolated information from another mesh (solid lines). (We will refer to the first mesh as a "recipient" mesh, and the second mesh as the "donor" of interpolated information). An interpolation is valid if an interpolation stencil can be found in the donor mesh for which none of the corner points making up the stencil are a hole or boundary point, or a boundary point is coincident with a corner point of an interpolation stencil in the donor mesh, and the corner point is not a hole or boundary point. The hole boundary point in Fig. 20 comprises a corner of two stencils that contain Points A and B; as a result, Points A and B cannot receive interpolated information from the donor mesh. Point C is coincident with a corner point of the donor mesh. Since the corner point is not itself a boundary or hole point, Point C can receive information from the donor mesh. None of the corner points of the stencil containing Point D is a boundary or hole point; therefore, Point D will be permitted to receive interpolated information from the donor mesh. In Fig. 20, Points A and B will be orphans unless the recipient mesh has a link to another mesh that can provide a valid interpolation stencil. PEGSUS prints a list of orphan points if any are found; orphans are also identified in graphical form (See Sec. 3.4) for convenience.

2.2.3 Quality

The preceding rules for determining a valid stencil can be relaxed by use of the QUALITY user input. QUALITY, which can be set globally or input for each mesh, is a value between 0.0 and 1.0, with 0.0 being bad and 1.0 being good (the previous discussion in Sec. 2.2.2 corresponds to a QUALITY = 1.0 for a valid stencil). QUALITY is specified by three values. The first value is the initial value, the second value is the ending value (which must be less than the initial value), and the third value is the increment (which must be a negative value). Using these three values, all LINKed meshes are searched for valid interpolations based on the initial QUALITY value. If any interpolation stencils are not found, the minimum acceptable value for QUALITY is changed by the increment value and the search for valid interpolations is repeated for the link list. The process is continued until all interpolation stencils are found or QUALITY has reached the ending value.

The method in which QUALITY is implemented is illustrated in Fig. 21. For demonstration purposes, a two-dimensional stencil is used with interpolation coefficients in the xi and eta directions. As part of the check for a valid interpolation, each stencil point is checked to determine if it is an interpolation point. When a stencil point is found to be an interpolation point, the coefficients of the interpolation are checked to determine if they are valid based on QUALITY. Figure 21a demonstrates this check for QUALITY = 0.8. In this case, the lower left corner is an interpolation boundary point. The shaded region is the area where interpolation coefficients are valid. If a second point is also an interpolation boundary point, then the region for valid interpolation coefficients is reduced as shown in Fig. 21b. Finally, in Fig. 21c the region is further narrowed for three interpolation boundary points within the stencil. Of course, other combinations are possible when three dimensions are considered.

2.3 ADDITIONAL FEATURES

Several additional features are implemented in PEGSUS 4.0 to allow greater user control of PEGSUS results. The features discussed in the following sections are interior blanking, inclusion, phantom meshes, and restarting by addition and subtraction.

2.3.1 Interior Blanking

When complex configurations are modeled, there sometimes arises a need to use the blanking capability of the chimera scheme to allow points within the interior of a mesh to be updated by the application of boundary conditions on a surface. Typically a solid wall geometry internal to a mesh is handled in this way. For example, the fins on a store or the corner of a wall can be handled with a single mesh and minimal mesh stretching (see Fig.

22). This interior region, which is referred to as an interior blanked region, will have no interpolation boundary points (fringe points) placed around the blanked area. To specify these computational regions for interior blanking, the $REGION input is used with TYPE = 'INTR' (See Sec. 3.2). The ranges of J, K, and L indices of the blanked region are set with the $VOLUME user input (See Sec. 3.2). The user must specify the correct boundary conditions on the surfaces of the region within the flow solver itself.

Within PEGSUS, interior blanked points are handled differently from the points blanked for holes, particularly when an interpolation stencil is being checked to determine if it is valid. Valid and invalid interpolation stencils near interior blanked regions (see Sec. 2.2.2 for discussion of valid and invalid interpolation) are shown in Fig. 23 with two interpolation points that are considered valid (open circles) and two points that are considered invalid (shaded squares). As can be seen, an interpolation boundary point must be completely enclosed by interior blanked points to be considered invalid. When the validity of an interpolation stencil in or near interior blanked regions is checked, QUALITY (See Sec. 2.2.3) is never invoked. Interior blanked points are also handled differently in the IBLANK array for plotting (see Sec. 3.4).

2.3.2 Inclusion

To help reduce the number of interpolations and possible orphan points, user inputs can be specified to enclose the physical domain globally; in addition the physical and/or the computational domains may be enclosed for each mesh. The enclosure of the domains will exclude boundary points outside the specified ranges. The specification of the domain is accomplished through the inputs XINCLUDE, YINCLUDE, ZINCLUDE, JINCLUDE, KINCLUDE, and LINCLUDE. These inputs specify either the physical or computational coordinates within which all boundary points must be included. Boundary points outside the prescribed coordinates are dropped from the interpolation point lists. This capability allows points on a reflection plane to be excluded from the interpolation boundary point list and, therefore, reflection plane points will not show up as orphan points.

2.3.3 Phantom Meshes

Phantom meshes are used to improve and simplify the creation of holes, as mentioned in Sec. 2.2.1. An example of phantom mesh use occurs in representing a wing/body junction. A complex wing/body junction can result in incorrect hole points if the junction is not properly represented. Figure 24a depicts the junction of a wing and a body. The boundary of the wing mesh extending from the wing root is coincident with the surface of the fuselage. As a result, the fuselage surface does not create a hole in the wing mesh. The wing does, however, create

a hole in the fuselage mesh. A hole creation boundary defined in the wing mesh cannot be completely closed; an aperture in the hole creation boundary will exist at the intersection of the wing surface and the hole creation boundary. As a result, some points close to the fuselage may be considered by the hole location algorithm to be outside the hole creation boundary; these points will not be blanked as required (see Fig. 24a).

A solution to this problem is illustrated in Fig. 24b. The structure of PEGSUS 4.0 allows a mesh to serve only as a hole creation boundary. Such meshes are called phantom meshes and do not interact in any other way with the other meshes; i.e., they do not provide interpolated information to boundaries of other meshes nor do they receive information interpolated from other meshes. In Fig. 24b, the hole creation boundary shown in Fig. 24a has been used to define the outer boundary of a phantom mesh. However, the phantom mesh extends into the interior of the fuselage mesh. The outer boundary of the phantom mesh is defined as a hole creation boundary and, therefore, creates a hole in the fuselage mesh. In effect, the phantom mesh acts to create the proper hole in the fuselage mesh for the wing mesh. Since the hole creation boundary of the phantom mesh is completely closed, hole points are correctly identified in the fuselage mesh.

The surfaces of the fuselage and wing do not create holes in the phantom mesh, since the phantom mesh is not used in flow computations. PEGSUS 4.0 does not include phantom meshes in the composite mesh file. However, all phantom meshes are stored for restart executions within the PEGSUS restart file. Phantom meshes are identified by PEGSUS when no LINK input is specified for a mesh.

2.3.4 Restarting — Addition and Subtraction

Applications of the chimera scheme to more complex configurations have increased the number of meshes required as well as the computational time to perform the functions of PEGSUS. PEGSUS 3.x, being a batch operation, required all corrections to be recalculated when a small change or correction was made to the user inputs or meshes. PEGSUS 4.0 includes a restart capability to allow subtraction and/or addition of meshes, boundaries, and/or regions. A composite mesh and associated interpolation data can therefore be incrementally developed for a given configuration.

The philosophy used in PEGSUS 4.0 for restart conditions is to recalculate as little as possible. This requires setting up an internal data structure that allows the reuse of previous boundary interpolation data. Previous boundary interpolation stencils, and coefficients associated with subtracted meshes are recalculated. Interpolation boundary points, stencils, and coefficients that are no longer required are removed from the appropriate lists. Thus,

as much as possible of the previously interpolated boundary results is used when calculating new and/or changed boundaries.

The determination of the interpolation boundary points is performed differently when adding and subtracting a component, particularly for hole boundaries. The addition or subtraction of outer boundaries is easily determined since they are a direct specification of the boundary points. The difficulty comes with redefining hole boundaries because they are created by several different methods and can be generated by one or more meshes or boundaries.

The process used for adding a hole is depicted in Fig. 25 for a case with a single fringe. The previous hole and hole boundary shown in Fig. 25a are brought into PEGSUS from the previous interpolation file. The new hole and hole boundary that have been specified for this restart execution are shown in Fig. 25b. The new hole and hole boundary are created in a temporary array within PEGSUS. Next, the new hole and hole boundary are superimposed on the previous hole and hole boundary. The result is illustrated in Fig. 25c. Note that some of the previous hole boundary points are unchanged, while others have become hole points and must be removed from the interpolation data file. Also, there are new boundary points requiring interpolation.

Subtraction of a hole and hole boundary is done differently, as depicted in Fig. 26. The hole defined in Fig. 25c is shown in Fig. 26a. The hole and hole boundary defined in Fig. 25a are to be subtracted from the hole and hole boundary in Fig. 26a. The new resulting hole and hole boundary are found by creating a list of points that includes all points of the previous hole and hole boundary, searching this list of points to determine which points are still in the hole from hole definitions not subtracted (See Sec. 2.1.2), and creating a fringe around the resulting hole. The resulting hole and hole boundary are shown in Fig. 26b, which is the same hole and hole boundary shown in Fig. 25b. Just like addition, there are previous boundary points that must be removed from the interpolation data file and new boundary points that require interpolation.

The process of subtraction and addition of a hole and hole boundary can be performed within the same PEGSUS run. This process is basically a combination of the two earlier processes with subtraction being performed first and the superimposition of the added hole coming second. Examples of addition and subtraction are given in Sec. 4.2 and Appendix G.

3.0 PROGRAM IMPLEMENTATION

The input and output files for PEGSUS 4.0 are shown in Fig. 27. The following is a description of the parameters, NAMELIST user inputs, input mesh format, and outputs for PEGSUS 4.0.

3.1 PARAMETERS

The following constants are defined in PARAMETER statements at the beginning of all program modules. The values of the constants must be equal to or exceed the expected values for a given composite mesh.

Name	Description
MDIM	maximum number of meshes
LNDIM	maximum number of LINKs
ICHAR	maximum number of characters in a name (must be 40 characters or less). Note that this parameter also controls the length of the mesh name read from the mesh input file (See Sec. 3.3)
NBDIM	maximum number of boundaries
MHDIM	maximum number of MHOLEIN
NSDIM	maximum number of surfaces
MSLEN	maximum number of points defining a single surface
NXDIM	maximum number of boxes
NRDIM	maximum number of regions
NVDIM	maximum number of volumes
INCORE	= 0 external storage used (i.e., SSD or disk) = 1 internal storage used (i.e., CRAY 2)
MLEN	maximum number of points in a single mesh

MLEMAX	if INCORE = 0, MLEMAX = MLEN if INCORE = 1, total number of points in all meshes
MILEN	maximum number of interpolation or stencil points in a single mesh
NBYTEI	number of bytes for an integer value
NBYTER	number of bytes for a real value

(NBYTEI and NBYTER are used to set up the record lengths of the direct access files when INCORE = 0.)

The parameters MSSMAX, MIMAX1, and MIMAX2 are calculated based on earlier defined values and should not be changed.

As may be noted in the parameters, PEGSUS 4.0 has the ability to keep all run time memory within core or to utilize the disk (i.e., for computers with limited memory). To keep all run time memory within core the INCORE parameter must be set equal to one, MLEMAX must be set to the total of all mesh points, and MLEN equal to the maximum sized mesh. To utilize disk memory, set INCORE equal to zero, MLEN equal to the maximum sized mesh, and MLEMAX equal to MLEN. Using INCORE = 1 is preferable (though typically not obtainable) since the I/O of the program is reduced.

3.2 NAMELIST GROUPS AND RECORD DEFINITIONS

The NAMELIST input is divided into seven groups: $GLOBAL, $MESH, $BOUNDARY, $SURFACE, $BOX, $REGION, and $VOLUME. Note that all user input record names are capitalized and all alphanumeric names (i.e., NAME, MHOLEIN, LINK) are case sensitive. The group names and associated records are defined as follows:

$GLOBAL

This input sets global values for all meshes. Each record can be overridden for any individual mesh by using the $MESH input. However, the mesh override is limited to that individual mesh. For subsequent meshes, the global values are used unless specifically overridden. The $GLOBAL inputs are recognized and implemented during an initial execution of PEGSUS 4.0 but not for a restart execution (i.e., $GLOBAL cannot be changed for a restart execution).

EPS — Tolerance of interpolation coefficient. Value of interpolation coefficients less than 0 and greater than 1.0 that is allowed for a converged stencil (default = 0.001).

FRINGE — Set equal to 1 for single fringe (default) or equal to 2 for double fringe (See Sec. 2.1.2.3).

QUALITY — Value defining quality of interpolation stencil ranging from 0.0 (bad) to 1.0 (good), (default = 1.0) (See Sec. 2.2.3). Three values are entered in the following order: the initial value, the ending value, and the increment (which must be negative).

XINCLUDE, YINCLUDE, ZINCLUDE — Range of X, Y, and Z values, respectively, to be included as boundary points (See Sec. 2.3.2). Composed of two values (a minimum and a maximum value) in each of the X, Y, and Z directions, respectively.

$MESH

ADDHOLE — List of hole creation boundaries defined in the previous PEGSUS execution that are to make a hole in this added mesh (must be any alphanumeric characters with no special symbols). This input is used only for restart when MODE = 'ADD'. Each boundary name is a string not exceeding ICHAR characters. No default values are defined.

ADDLINK — List of meshes defined in the previous PEGSUS execution that must be linked to this added mesh (must be any alphanumeric characters with no special symbols). The named mesh (specified with NAME) is put at the top of the LINK list of each mesh specified by ADDLINK. This input is used only for restart when MODE = 'ADD'. Each mesh name is a string not exceeding ICHAR characters. No default values are defined.

ALFA, BETA, GAMA — Euler angles (degrees) determining rotation of input meshes. ALFA, BETA, and GAMA are rotations about the Z, Y, and X coordinates, respectively. Default values for ALFA, BETA, and GAMA are 0.0. The coordinates of a mesh are transformed in the following sequence of operations: scaling, translation, and rotation.

EPS — Tolerance of interpolation coefficient. Value of interpolation coefficients less than 0 and greater than 1.0 that is allowed for a converged stencil. Default is $GLOBAL value.

FRINGE — Set equal to 1 for single fringe or equal to 2 for double fringe. Default is $GLOBAL value (See Sec. 2.1.2.3).

JINCLUDE, KINCLUDE, LINCLUDE — Range of J, K, and L indices, respectively, to be included for boundary points (See Sec. 2.3.2). Composed of two values (a starting and an ending value) in each of the J, K, and L directions, respectively.

LINK — Priority list of meshes that are to be searched to provide interpolations for boundary points (must be any alphanumeric characters with no special symbols). Each LINK is a string not exceeding ICHAR characters. LINKs are specified as follows: LINK = 'mesh name 1', 'mesh name 2', etc., (to link all meshes specify LINK = ,)

MODE — Defines a mesh that is added (='ADD'), or subtracted (='SUB'). MODE is specified only for restart.

NAME — Name of mesh to which NAMELIST record refers (must be any alphanumeric characters with no special symbols). A NAME is a string not exceeding ICHAR characters. No default values are defined.

QUALITY — Value defining quality of interpolation stencil ranging from 0.0 (bad) to 1.0 (good), (default is $GLOBAL value) (See Sec. 2.2.3). Three values are entered in the following order: the initial value, the ending value, and the increment (which must be negative).

SCALE — Mesh scaling factor. The original mesh coordinates are multiplied by the scaling factor. The default value is 1.0. The coordinates of a mesh are transformed in the following sequence of operations: scaling, translation, and rotation.

XINCLUDE, YINCLUDE, ZINCLUDE — Range of X, Y, and Z values, respectively, to be included for boundary points (See Sec. 2.3.2). Composed of two values (a minimum and a maximum value) in each of the X, Y, and Z directions, respectively.

XR,YR,ZR — Point (in translated and scaled coordinates) about which input mesh is to be rotated. Default values are 0.0. The coordinates of a mesh are transformed in the following sequence of operations: scaling, translation, and rotation.

X0,Y0,Z0 — Mesh translation factors. These factors are added to the original mesh coordinates. Default values are 0.0. The coordinates of a mesh are transformed in the following sequence of operations: scaling, translation, and rotation.

$BOUNDARY

CLOSED — Either .TRUE. (default) or .FALSE. Set equal to .FALSE. when the surfaces do not describe a completely closed boundary (See Sec. 2.1.2.1).

ISPARTOF — Mesh which contains boundary (must be any alphanumeric characters with no special symbols). ISPARTOF is a string not exceeding ICHAR characters. No default is defined.

MHOLEIN — Meshes in which the named boundary causes holes. MHOLEIN's are specified as follows: MHOLEIN = 'mesh name 1', 'mesh name 2', etc. Each mesh name in MHOLEIN is a string not exceeding ICHAR characters. No defaults are defined.

MODE — Defines a boundary that is added (= 'ADD'), or subtracted (= 'SUB'). MODE is specified only for restart. If boundary is connected to added mesh, MODE is not required. If boundary is connected to subtracted mesh, $BOUNDARY need not be specified.

NAME — Name of boundary (must be any alphanumeric characters with no special symbols). A boundary name is a string not exceeding ICHAR characters. No default is defined.

$SURFACE

ISPARTOF — Boundary to which surface belongs (must be any alphanumeric characters with no special symbols). ISPARTOF is a string not exceeding ICHAR characters. No default is defined.

JRANGE, KRANGE, LRANGE — Ranges of indices that define surface. Composed of two values (a starting and an ending value) in each of the J, K, and L directions, respectively. Maximum allowable ranges are:

JRANGE: 1..JMAX
KRANGE: 1..KMAX
LRANGE: 1..LMAX

No defaults are defined.

MESH NAME — Name of mesh that contains the surface (ISPARTOF) (must be any alphanumeric characters with no special symbols). MESH NAME is used only for hole creation boundaries. Name is a string not exceeding ICHAR characters. Default is the mesh name given in ISPARTOF for $BOUNDARY.

NVOUT — Direction of normal outward from hole. NVOUT is required only for hole creation boundaries. Allowable values are "+J", "−J", "+K", "−K", "+L", and "−L".

$BOX

ISPARTOF — Boundary to which box belongs (must be any alphanumeric characters with no special symbols). ISPARTOF is a string not exceeding ICHAR characters. No default is defined.

XRANGE, YRANGE, ZRANGE — Ranges of X, Y, and Z coordinates that define box. Composed of two values (a minimum and a maximum value) in each of the X, Y, and Z directions, respectively.

$REGION

ISPARTOF — Mesh which contains region (must be any alphanumeric characters with no special symbols). ISPARTOF is a string not exceeding ICHAR characters. No default is defined.

MODE — Defines a region that is added (='ADD'), or subtracted (='SUB'). MODE is specified only for restart. If region is connected to added mesh, MODE is not required. If region is connected to subtracted mesh, $REGION need not be specified.

NAME — Name of region (must be any alphanumeric characters with no special symbols). A region name is a string not exceeding ICHAR characters. No default is defined.

TYPE — Defines type of region being specified. If TYPE='HOLE', then hole with a fringe is created. If TYPE='INTR', then an interior region will be blanked out but with no fringe. Default is 'HOLE'.

$VOLUME

ISPARTOF — Region to which volume belongs (must be any alphanumeric characters with no special symbols). ISPARTOF is a string not exceeding ICHAR characters. No default is defined.

JRANGE, KRANGE, LRANGE — Ranges of indices that define volume. Composed of two values (a starting and an ending value) in each of the J, K, and L directions, respectively. Maximum allowable ranges are:

JRANGE: 1..JMAX
KRANGE: 1..KMAX
LRANGE: 1..LMAX

No defaults are defined.

3.3 INPUT MESH FORMAT

The format of the mesh input file is identical to that of PEGSUS 3.x. The input format uses three records. These records are repeated for each mesh.

Record	Variables
1	Mesh Name
2	JMAX, KMAX, LMAX
3	(((X(J,K,L),J = 1,JMAX),K = 1,KMAX),L = 1,LMAX), (((Y(J,K,L),J = 1,JMAX),K = 1,KMAX),L = 1,LMAX), (((Z(J,K,L),J = 1,JMAX),K = 1,KMAX),L = 1,LMAX)

where JMAX, KMAX, and LMAX are the maximum mesh indices in the J, K, and L directions, respectively. Mesh Name must be ICHAR in length.

3.4 OUTPUT

There are seven output files generated by PEGSUS 4.0 during each execution (see Fig. 27). The files are: (1) composite mesh, (2) interpolation data, (3) execution summary, (4) diagnostic maps, (5) PEGSUS restart, (6) IBLANK array for plotting, and (7) orphan point list.

The composite mesh file, interpolation data file, and the orphan point list file are used by the flow solver. The structure of these files is given in Appendix F. The composite mesh file and the interpolation data file are required for a restart of PEGSUS.

The execution summary, which is written to logical unit 6, contains (1) a summary of the meshes read, (2) a summary of user inputs, (3) error messages, (4) an orphan point summary, and (5) flow solver parameters. Items 1, 2, 4, and 5 will be illustrated in Sec. 4.0. Error checking is performed by PEGSUS on the inputs specified by the user and on array overruns. First, key words (user inputs, i.e., NAME, FRINGE, ISPARTOF, etc.) are checked to determine if they are recognized by the program. If a key word is not recognized, an error message is printed and the value is not read by PEGSUS. The second level of checking determines if connections (i.e., ISPARTOF and MHOLEIN) specified by the user point to existing meshes, boundaries, or regions. Again, if an error is found, a message defining the problem is printed. The third level of checking determines if an array is overrun. The error message for these errors indicates what parameter needs to be changed to correct the problem. PEGSUS tries to find all user input error before terminating the processing.

The diagnostic maps generated by PEGSUS are diagrams of computational space showing holes, boundary points, and interpolation stencils. These maps and their usefulness will be illustrated in Sec. 4.0.

The PEGSUS restart file contains all user inputs and calculated values from the current PEGSUS run. This file is required to restart PEGSUS. The structure of this file is given in Appendix F.

The IBLANK array for plotting is a file used in conjunction with the mesh files for use with PLOT3D (Ref. 19) and FAST (Ref. 20). The IBLANK array for plotting is different from the IBLANK array sent to the flow solver in that the boundary points are not set equal to zero but equal to minus the mesh number of the donor mesh (see Sec. 2.3.1). Field points are still equal to 1 and hole points equal to 0. For interior blanked points the surfaces (minimum and maximum index planes) of the region are set equal to one with the points within the region set equal to zero. The difference in setting the IBLANK array for plotting is to allow values to be plotted on the surfaces where the boundary conditions are set. The structure of this file is given in Appendix F.

4.0 EXAMPLES

Six examples of the application of PEGSUS 4.0 to representative multiple-mesh problems are presented in Secs. 4.1 and 4.2. These examples are not intended to be illustrations of the application of PEGSUS 4.0 to actual engineering problems. Instead, they are intended to demonstrate most of the basic principles of PEGSUS 4.0 that are common to a majority of PEGSUS 4.0 scenarios. A set of examples of the application of PEGSUS 4.0 to an actual engineering problem (an empty cavity, a store/sting combination in a cavity, and a store in a cavity) is presented in Appendix G.

In Sec. 4.1 four examples are given, the first three of which illustrate in turn each of the different methods available for hole creation. The fourth example consists of a hole creation boundary comprised of surfaces from two different meshes. Section 4.2 consists of two examples designed to illustrate the use of the addition and subtraction features of PEGSUS 4.0. Common to all examples is the use of a double fringe. It is recommended that prospective PEGSUS 4.0 users carefully study the annotated input and diagnostic maps included for all six examples to gain a solid understanding of the process.

4.1 HOLE CREATION EXAMPLES

The following four examples illustrate various means of hole creation through the use of simple cylindrical and Cartesian mesh configurations.

4.1.1 Indirect Hole Creation Using $BOUNDARY and $SURFACE

A simple two-mesh problem is depicted in Fig. 28. A right circular cylinder of finite length (surrounded by a cylindrical mesh) is embedded in a Cartesian mesh. The outer boundary of the cylindrical mesh receives data interpolated from the surrounding Cartesian mesh. Since the cylinder is embedded in the Cartesian mesh, a region of the Cartesian mesh must be excluded from the computation. The exclusion is achieved by a hole creation boundary defined by a set of surfaces within the cylindrical mesh. To illustrate the flexibility of PEGSUS 4.0, a hole creation boundary is defined that consists of two half cylinders of unequal radii. If the hole creation boundary were to be defined as a collection of four surfaces (the two half-cylindrical surfaces and the two rectangular surfaces between the larger and smaller radii), the hole boundary would contain concavities. For reasons described in Sec. 2.1.2.1, defining a hole boundary with concavities will result in an incorrect hole in the Cartesian mesh. The correct way to describe the desired hole boundary is as two separate hole boundaries, neither of which contains any concavities. In this case, one hole boundary is defined as the larger half-cylindrical and two rectangular surfaces extending to the surface of the cylinder; the other hole boundary consists of the smaller half-cylindrical and two corresponding rectangular surfaces. The two hole boundaries will merge into a single hole created in the Cartesian mesh.

The annotated input for the problem of Fig. 28 is given in Fig. 29 (see Sec. 3.2 for a description of user input). Two mesh, three boundary, and eight surface definitions are required. Maximum index values for both the cylindrical and Cartesian meshes are 41, 41, and 5 in the J, K, and L directions, respectively. The hole boundary defined by the half-cylinder of smaller radius is specified by four surfaces, while the hole boundary defined by the half-cylinder of larger radius is specified by three surfaces. The outer boundary of the cylindrical mesh is defined by a single surface. No boundaries or surfaces of the Cartesian mesh are defined in the input, since no predefined surface in the Cartesian mesh receives information from the cylindrical mesh.

When PEGSUS 4.0 is run, an execution summary file is produced as well as diagnostic maps which are intended to aid the user in determining whether the mesh communications are being performed properly. Figure 30 is a reproduction of a map produced from the input in Fig. 29. The maps are graphical depictions of the computational meshes which show: (1) which points are blanked out hole points, (2) which points are boundary points and which mesh updates the boundary points, (3) which points are stencil reference points (i.e., the point in the interpolation stencil with lowest J, K, and L indices) and which meshes the interpolation stencils update, and (4) which points are "orphans," i.e., interpolation boundary points which, for some reason, cannot be interpolated from other meshes. At the beginning of the diagnostic maps is a legend giving a key to the letters and symbols used. The map in Fig. 30 clearly shows the location of the hole in the Cartesian mesh as well as indicating

that the hole boundary points are being updated by the cylindrical mesh. Also, the location of stencil reference points in the Cartesian mesh that are interpolating information to the outer boundary of the cylindrical mesh are shown. With a little practice, a user can obtain much useful information from the maps, such as the amount of mesh overlap, and whether interpolation stencils have been found for boundary points. For example, it can be seen from the map depicted in Fig. 30 that the amount of mesh overlap (i.e., the distance from the hole boundary points to outer boundary points of the cylindrical mesh) is marginal near the bottom of the hole boundary. Shown in Fig. 31 is a portion of the execution summary for this problem. The execution summary is only included for this example.

4.1.2 Indirect Hole Creation Using $BOUNDARY and $BOX

PEGSUS 4.0 provides a great deal of flexibility for the creation of holes. Figure 32 depicts the same two-mesh system used in Sec. 4.1.1. In this instance, the hole creation boundaries have been replaced by a box. The box consists of ranges of X, Y, and Z values (the minimum and maximum X and Y ranges are shown in Fig. 32) which define a region within the cylindrical mesh. All points in the Cartesian mesh contained within this box are considered hole points. A hole boundary (fringe) is placed around the hole to specify which points require interpolation. The remaining points are considered field points. The annotated input for this problem is shown in Fig. 33. Figure 34 shows a portion of the diagnostic maps that are output by PEGSUS 4.0 for this problem. The hole in the Cartesian mesh created by the box is clearly visible.

4.1.3 Direct Hole Creation Using $REGION and $VOLUME

In many situations, one would like to directly exclude a specific region (or regions) in a particular mesh. PEGSUS 4.0 allows this process through the use of the region/volume combination. Figure 35 illustrates just such a situation with the familiar Cartesian/cylindrical mesh system. Here the points to be blanked consist of a single region, with the volume being the J, K, and L values of the points in the Cartesian mesh which are to be blanked. The NAMELIST input for this problem is shown in Fig. 36 and a sample diagnostic map for the Cartesian mesh is shown in Fig. 37. Examination of the map shows that a hole boundary has been placed around the hole created by the region. This boundary would be omitted if the input for the region had included the TYPE = 'INTR' statement. In the absence of this input, the region type was defaulted to 'HOLE'.

4.1.4 Hole Creation Using Surfaces from Two Separate Meshes

When computing the flow field about complex geometric configurations, it is often desirable to break the mesh surrounding an object with a simple topology, such as a cylindrical mesh, into separate meshes. The need to do this frequently arises from the fact that a single

mesh would be too large for the computer memory available, although this is certainly not the only cause. Breaking a single mesh into two or more separate meshes can lead to difficulties when the split mesh configuration is embedded within other meshes. In earlier versions of PEGSUS, the surfaces defining a hole creation boundary were required to be part of the same mesh to which the hole creation boundary was connected. This would lead to problems for a split mesh system as the separate hole creation boundaries from each mesh would not create the hole that was needed. A phantom mesh would then be required to obtain the correct hole. This shortcoming has been alleviated in PEGSUS 4.0 by allowing the surfaces comprising a hole creation boundary to come from meshes other than the one to which the hole creation boundary is connected. To illustrate this feature, we again turn to the Cartesian mesh and cylindrical mesh combination. In this case the cylindrical mesh surrounding the cylinder has been split into two pieces as is shown in Fig. 38. The PEGSUS 4.0 input is shown in Fig. 39. Examination of the PEGSUS input reveals that a single hole creation boundary has been defined, and that this boundary is part of 'CYLINDER GRID 1'. This boundary is composed of two surfaces: one surface from 'CYLINDER GRID 1' and one surface contained in 'CYLINDER GRID 2'. The maps produced by PEGSUS 4.0 for the L = 1 plane are shown in Fig. 40.

4.2 ADDITION AND SUBTRACTION EXAMPLES

A major enhancement of PEGSUS 4.0 over earlier versions of the PEGSUS code is the ability to add and subtract meshes, boundaries, and regions to multiple-mesh configurations which have already been processed by PEGSUS 4.0 in a previous run. With earlier versions of PEGSUS, the only means by which objects could be added or subtracted from a multiple-mesh configuration was to rerun PEGSUS with the corresponding lines of input added or removed. The entire configuration then was reprocessed by the code. This was very costly and time consuming when only minor changes were required. With PEGSUS 4.0 the situation is different. Information from an earlier execution of the code is stored in such a way that it may be reused as input for later runs. In this manner, when one wishes to make changes to a multiple-mesh configuration, any information generated by an earlier run of the PEGSUS 4.0 code is recomputed only if the newly added or subtracted features require it to be recomputed. This capability leads to substantial savings in the time required to construct a complex mesh configuration. The following two examples illustrate the use of the addition and subtraction features of PEGSUS 4.0.

4.2.1 Addition

To demonstrate the addition feature of PEGSUS 4.0, the example in Sec. 4.1.1 is reconsidered (Figs. 28 – 31). It is assumed that the composite mesh file, the interpolation file, and the PEGSUS restart file are available for input to PEGSUS in the restart mode

(see Fig. 27). In this example, another Cartesian mesh will be added to the existing two-mesh system. The new three-mesh configuration is illustrated in Fig. 41, where the J and K coordinates are the same as in Fig. 28. Since the new Cartesian mesh overlaps both the original Cartesian mesh and the cylindrical mesh, PEGSUS 4.0 must have some way to incorporate the new mesh interactions. Examination of Fig. 41 shows that an outer boundary must be added to the original Cartesian mesh to allow information to be transferred from the new Cartesian mesh to the original Cartesian mesh. It must be recalled that in PEGSUS terminology an outer boundary specifically refers to those points in a mesh other than hole boundary points which receive data interpolated from other meshes. In addition, the new Cartesian mesh must be added to the MHOLEIN list of the hole boundaries created by the cylindrical mesh and also must be added to the link lists of the other two meshes. The PEGSUS 4.0 input for this problem is shown in Fig. 42. The outer boundary added to the original Cartesian mesh contains the MODE = 'ADD' statement so that PEGSUS will add the boundary. The new Cartesian mesh is added to the MHOLEIN list for the hole boundaries created by the cylindrical mesh through inclusion of the line ADDHOLE = 'CYLINDER HOLE BOUNDARY1', 'CYLINDER HOLE BOUNDARY2'. The new Cartesian mesh is also added to the link lists of the original Cartesian mesh and cylindrical mesh through inclusion of the line ADDLINK = 'CARTESIAN GRID', 'CYLINDER GRID'. The maps output by PEGSUS 4.0 for the L = 1 plane for all three meshes are shown in Fig. 43.

4.2.2 Subtraction

For the last example, the cylindrical grid will be subtracted from the previous example of Sec. 4.2.1 (Figs. 41 – 43). The PEGSUS input for this problem is shown in Fig. 44, while the maps for the Cartesian grids for the L = 1 plane are shown in Fig. 45. An examination of the maps shows that the cylindrical grid has been removed as well as the boundaries contained within the mesh. Note that the hole has been removed from both Cartesian meshes. For generality, note that the processes of addition and subtraction may be performed simultaneously.

5.0 CONCLUDING REMARKS

PEGSUS 4.0 is a code used in the chimera multiple-mesh flow-field calculation scheme. It has been developed, implemented, and applied successfully to aerodynamic configurations. PEGSUS 4.0 is an improvement over earlier versions in that it contains a restart capability that allows for addition and/or subtraction of meshes, boundaries and/or regions, the user has more control of the process through additional inputs, the code was written to be portable, the code is 4 to 10 times faster than PEGSUS 3.x, and requires 30 percent less memory than PEGSUS 3.x.

PEGSUS 4.0 may be obtained by contacting AEDC/DOCS, Arnold Engineering Development Center, Arnold AFB, TN 37389.

REFERENCES

1. Dietz, W. E. and Evans, S. B. "Interactive EAGLE: An Interactive Surface Mesh and Three-Dimensional Grid Generation System, Version 1.0, User's Guide." AEDC-TR-90-25 (AD-A230086), December 1990.

2. Benek, J. A., Steger, J. L., and Dougherty, F. C. "A Flexible Grid Embedding Technique with Applications to the Euler Equations." AIAA Paper No. 83-1944, July 1983.

3. Benek, J. A., Buning, P. G., and Steger, J. L. "A 3-D Chimera Grid Embedding Technique." AIAA Paper No. 85-1523, July 1985.

4. Benek, J. A., Dougherty, F. C., and Buning, P. G. "Chimera: A Grid-Embedding Technique." AEDC-TR-85-64 (AD-A167466), December 1985.

5. Benek, J. A., Donegan, T. L., and Suhs, N. E. "Extended Chimera Grid Embedding Scheme with Applications to Viscous Flows." AIAA Paper No. 87-1126, June 1987.

6. Dougherty, F. C., Benek, J. A., and Steger, J. L. "On Applications of Chimera Grid Scheme to Store Separation." NASA-TM-88193, October 1985.

7. Fox, J. H., et al. "Computations of the Euler Flow Field Produced about a Transonic Aircraft with Stores." AIAA Paper No. 89-2219, July-August 1989.

8. Dougherty, F. C. and Kuan, J. "Transonic Store Separation Using a Three-Dimensional Chimera Grid Scheme." AIAA Paper 89-0637, January 1989.

9. Suhs, N. E. "Computations of Three-Dimensional Cavity Flow at Subsonic and Supersonic Mach Numbers." AIAA Paper No. 87-1208, June 1987.

10. Donegan, T. L., Benek, J. A., and Erickson, J. C. "Calculations of Transonic Wall Interference." AIAA Paper No. 87-1432, June 1987.

11. Sickles, W. L. and Erickson, J. C. "Wall Interference Corrections for Three-Dimensional Transonic Flows." AIAA Paper No. 90-1408, June 1990.

12. NASA Ames Space Shuttle Flow Simulation Group, Buning, P., et al. "Numerical Simulation of the Integrated Space Shuttle Vehicle in Ascent." 4th International Conference on Super Computing, Santa Clara, CA, April 1989.

13. Meakin, R. L. and Suhs, N. E. "Unsteady Aerodynamics Simulation of Multiple Bodies in Relative Motion." AIAA Paper No. 89-1966, June 1989.

14. Kiris, C., et al. "Numerical Simulation of the Incompressible Internal Flow Through a Tilting Disk Valve." AIAA-90-0682, January 1990.

15. Reddy, K. C. and Benek, J. A. "A Locally Implicit Scheme for 3-D Compressible Viscous Flows." AIAA-90-1525, June 1990.

16. Lijewski, L. E. "Store Interference Aerodynamics." *Proceedings of the 8th JOCG Aircraft/Store Compatibility Symposium*, October 1990.

17. Dietz, W. E. and Suhs, N. E. "PEGSUS 3.0 User's Manual." AEDC-TR-89-7 (AD-A211974), August 1989.

18. Dietz, W. E., Jacocks, J. L., and Fox, J. H. "Application of Domain Decomposition to the Analysis of Complex Aerodynamic Configurations." *Proceedings of the Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, eds. T. F. Chan, R. Glowinski, J. Periaux, and O. B. Widlund, SIAM, Philadelphia, 1990.

19. Walatka, P. P., et al. "PLOT3D User's Manual." NASA TM-101067, 1990.

20. Bancroft, G. V., et al. "FAST: A Multi-Processed Environment for Visualization of Computational Fluid Dynamics." AIAA-91-0793, January 1991.

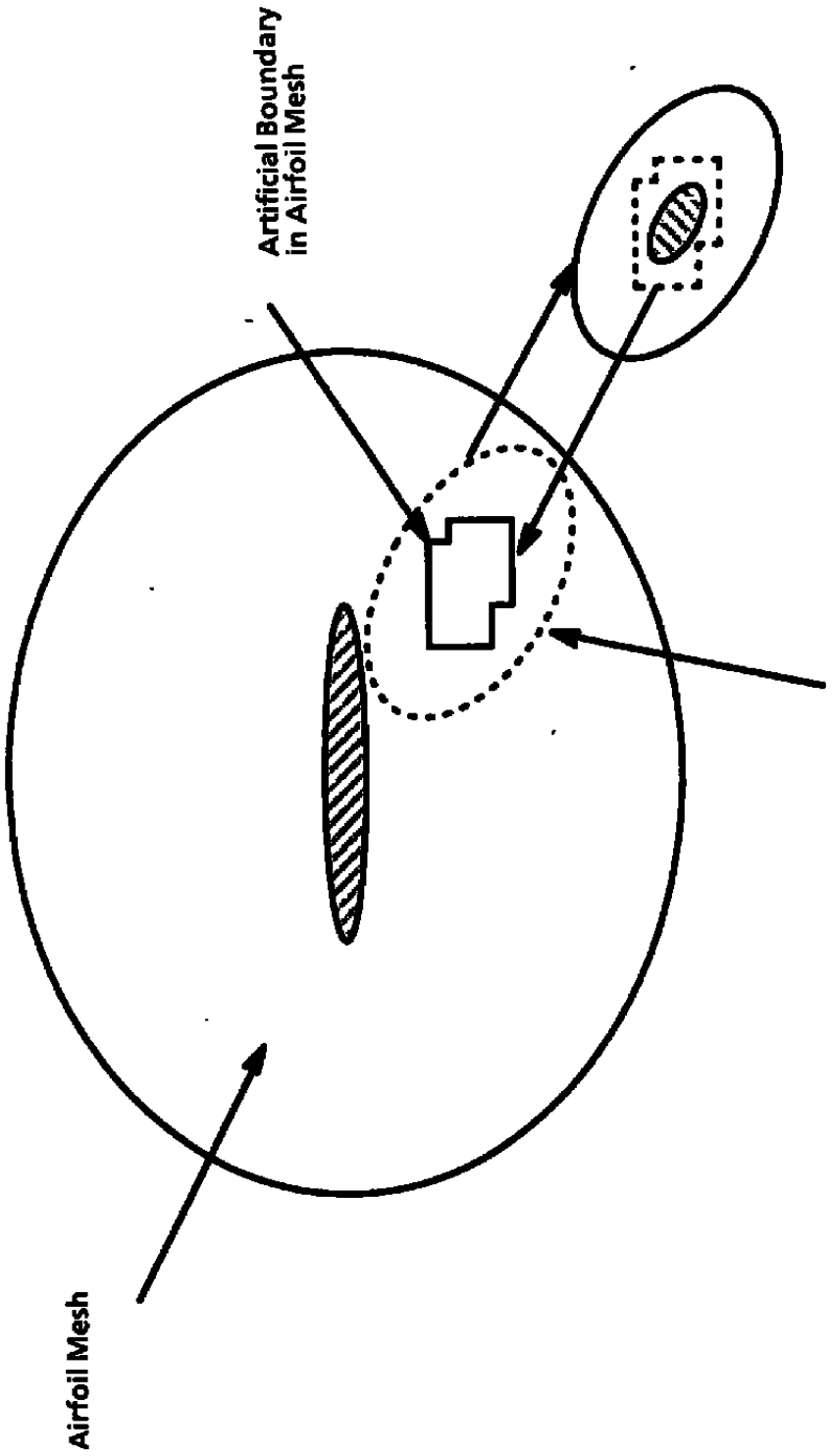


Figure 1. Mesh-to-mesh communication.

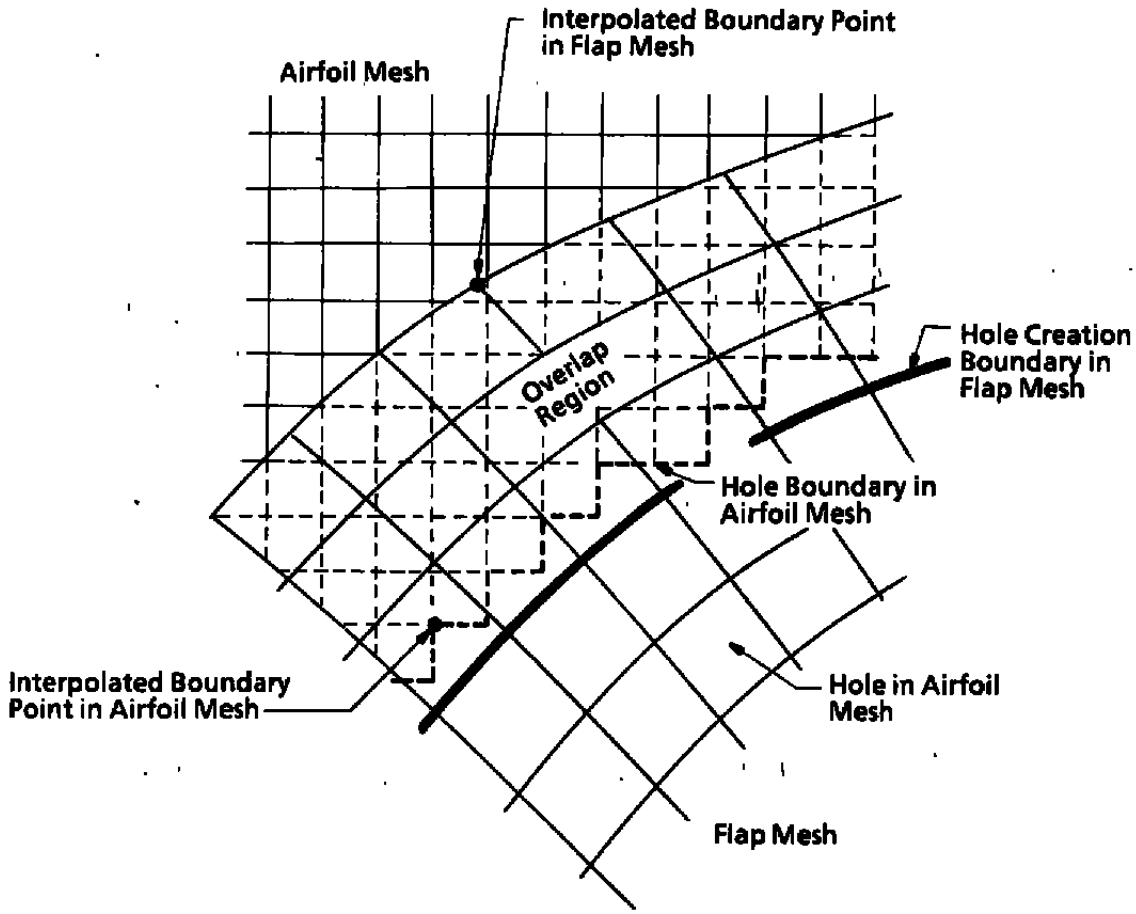


Figure 2. Overlap region between meshes.

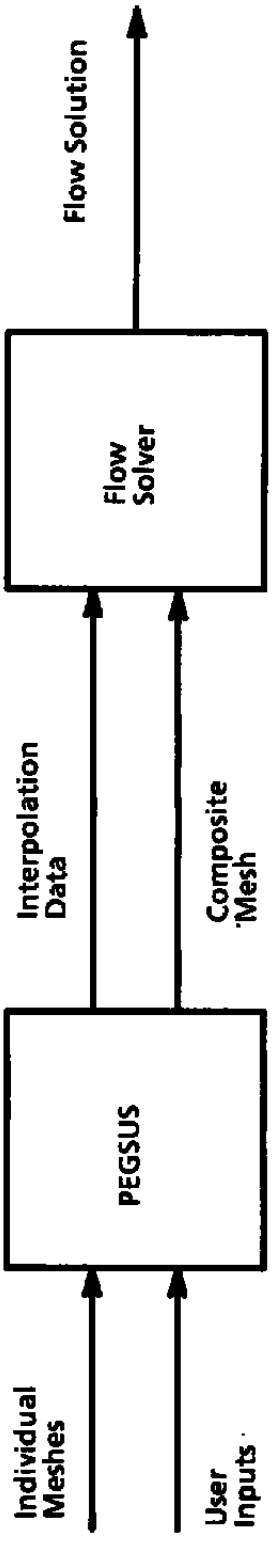


Figure 3. Chimera scheme.

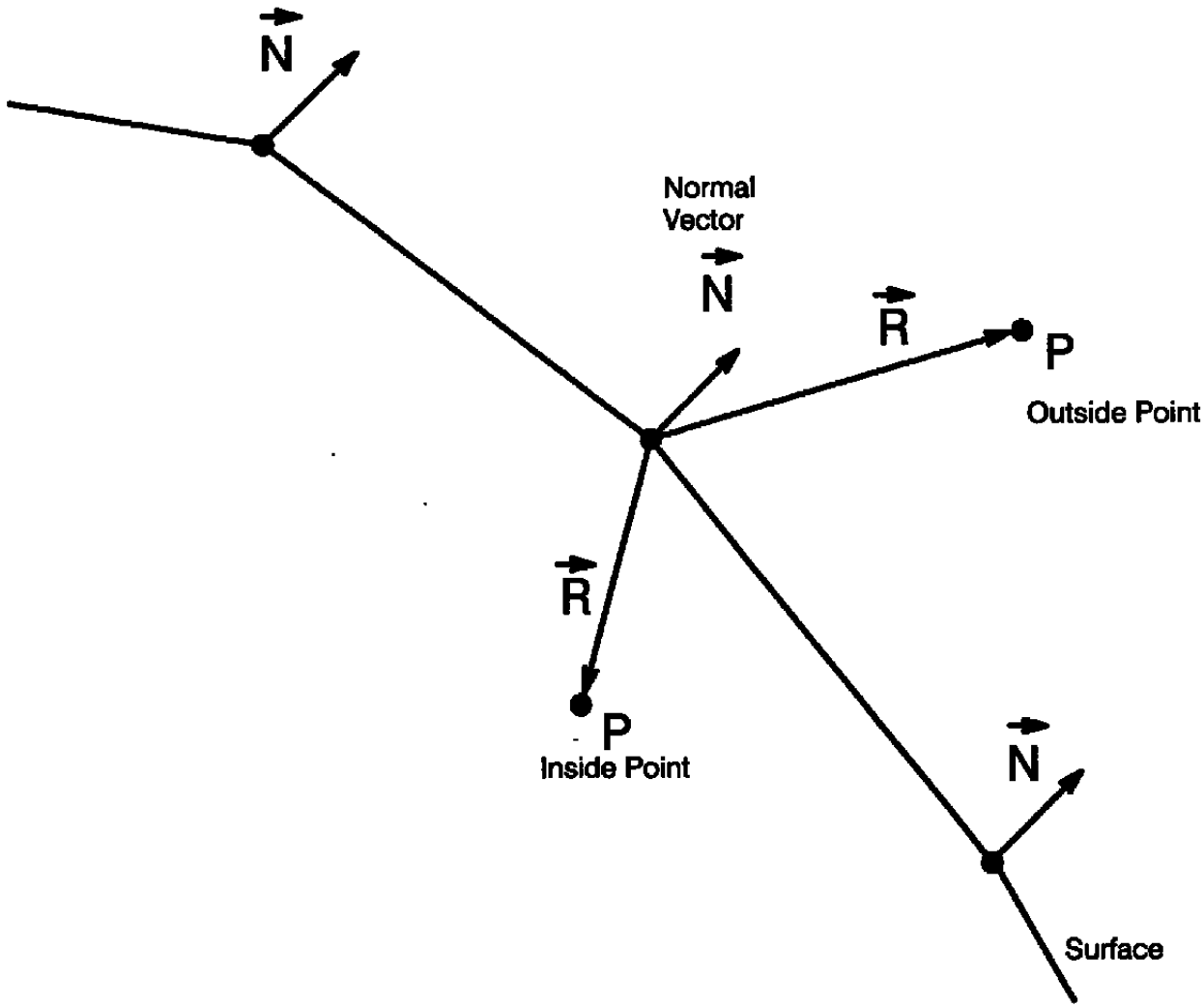
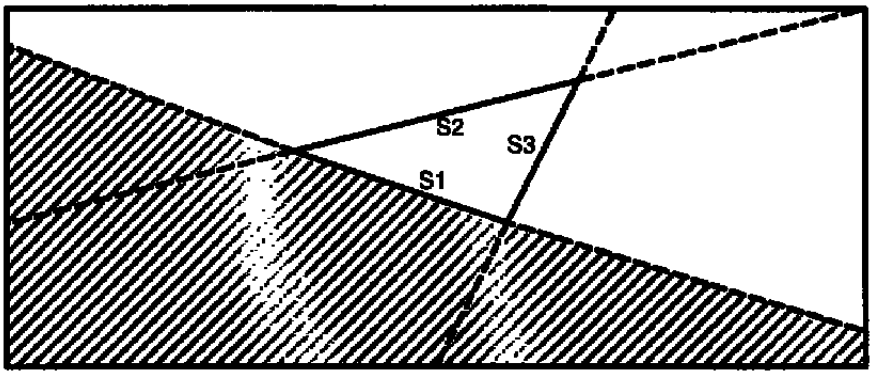
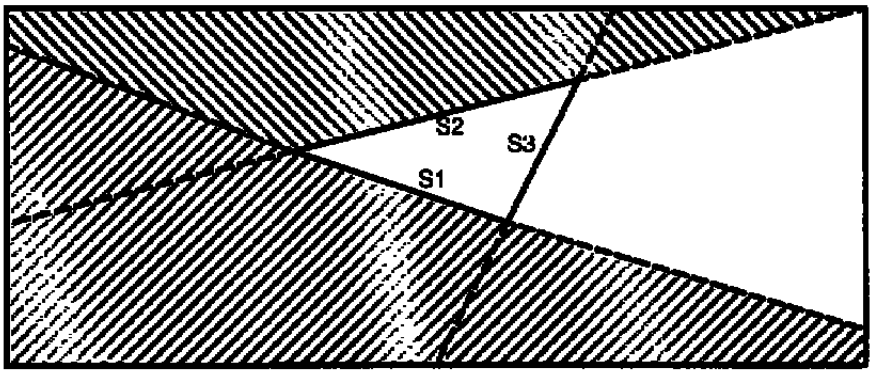


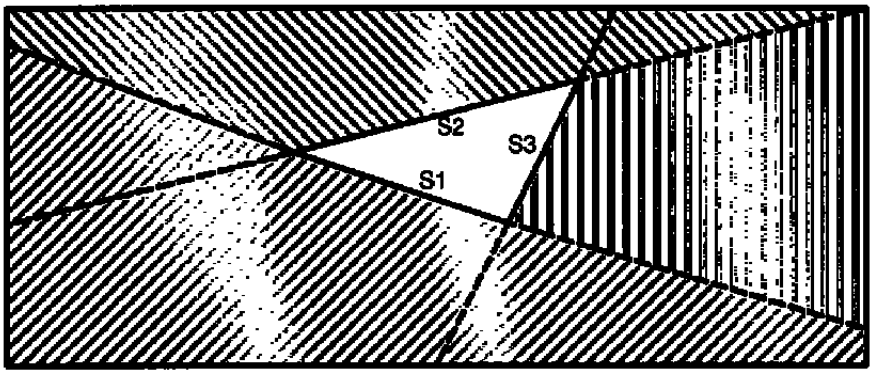
Figure 4. "Inside" and "outside" a surface.



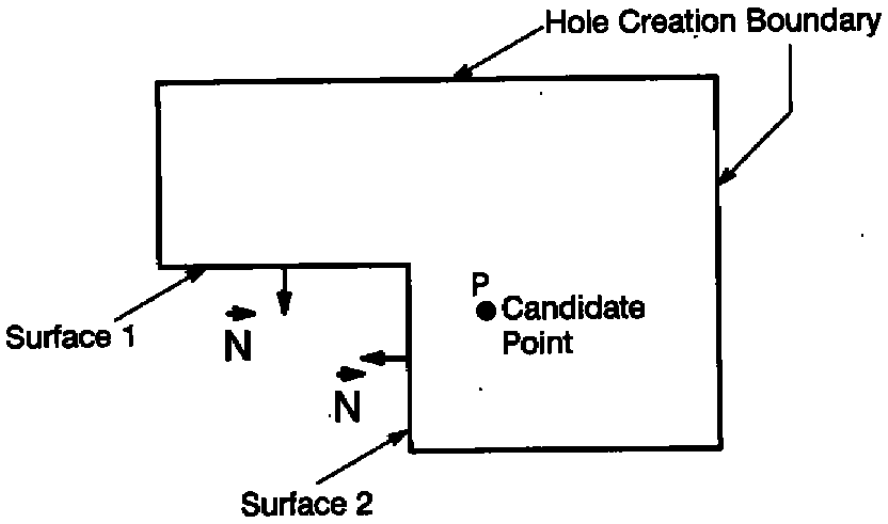
a. Area eliminated by S1



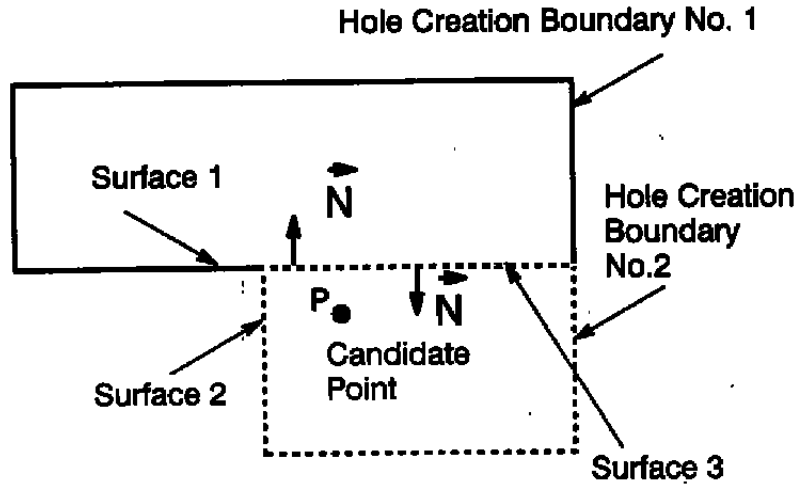
b. Area eliminated by S1 and S2



c. Area eliminated by S1, S2, and S3
Figure 5. Hole location.



a. Hole creation boundary with concavity.



b. Removal of concavity

Figure 6. Restrictions on hole creation boundaries.

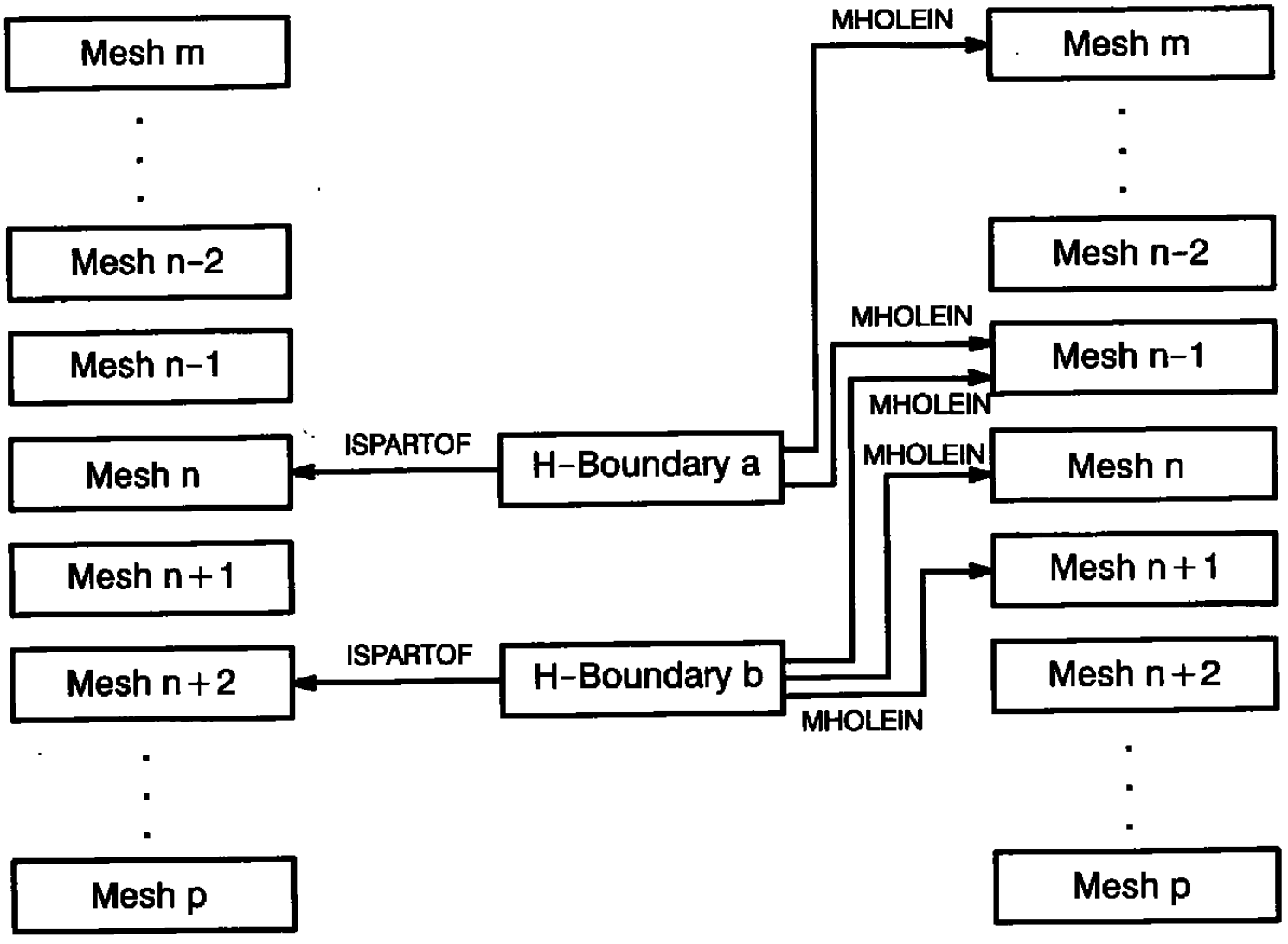


Figure 7. Hole creation boundary-mesh connections.

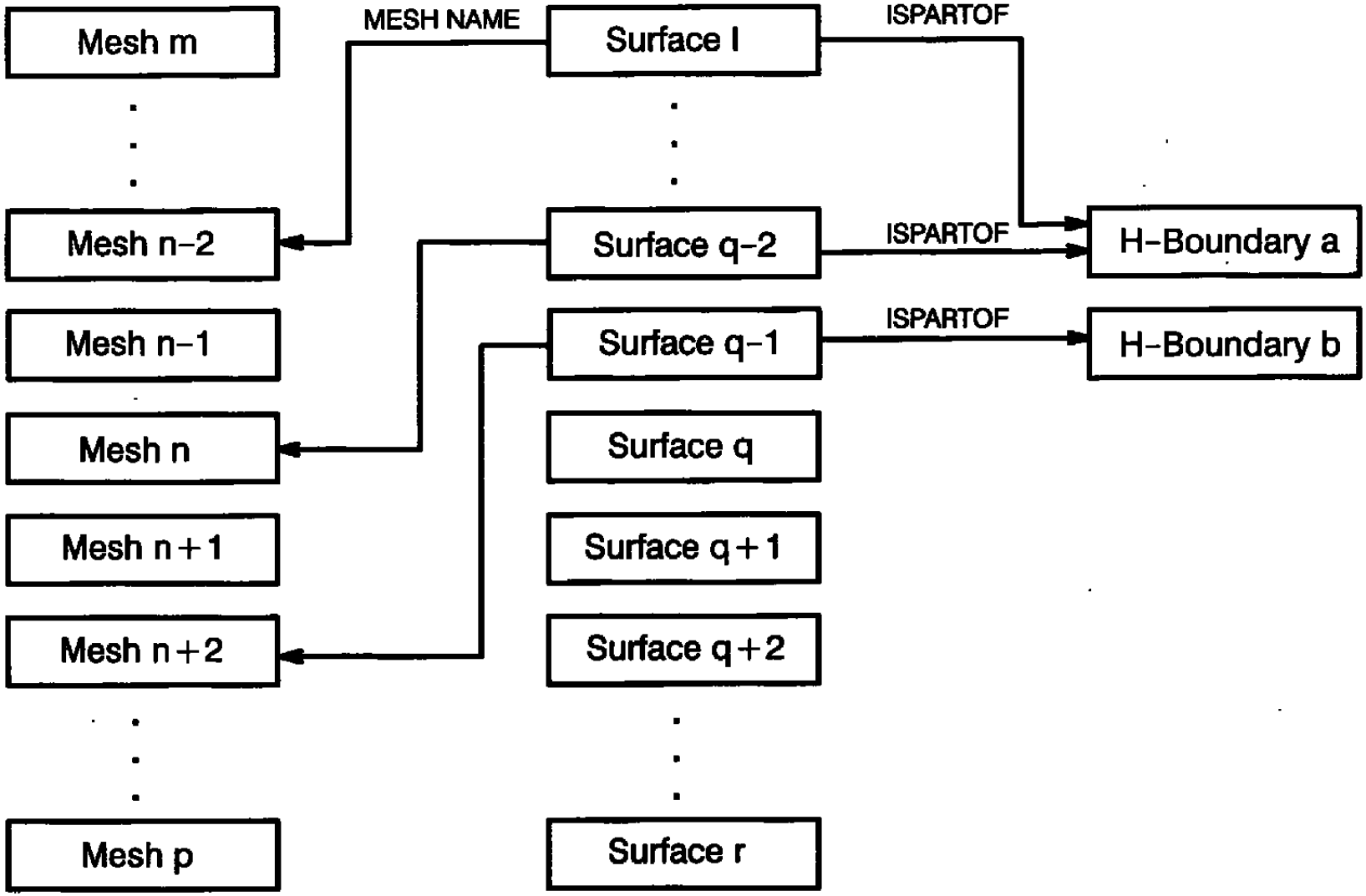
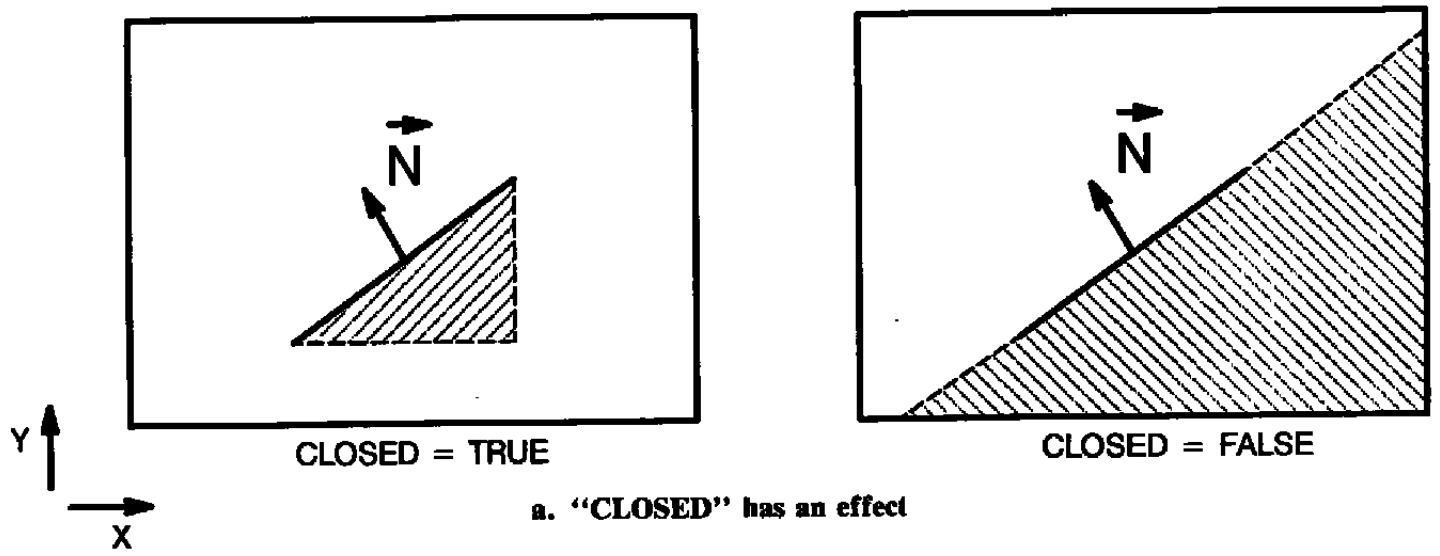
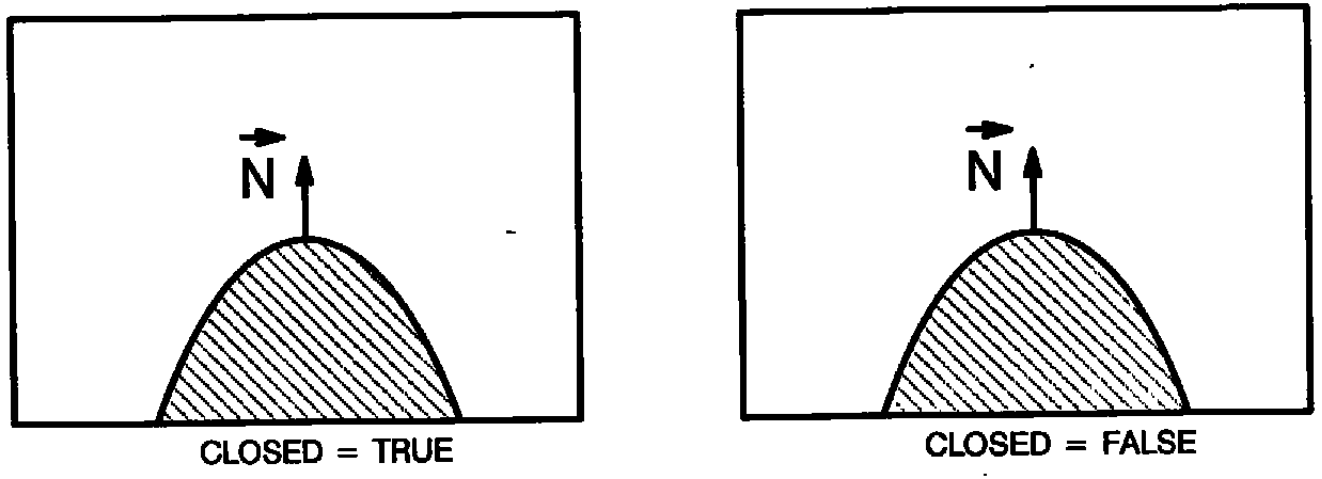


Figure 8. Surface-hole creation boundary connections.



a. "CLOSED" has an effect



b. "CLOSED" has no effect

Figure 9. Effects of the "CLOSED" user input on hole boundaries.

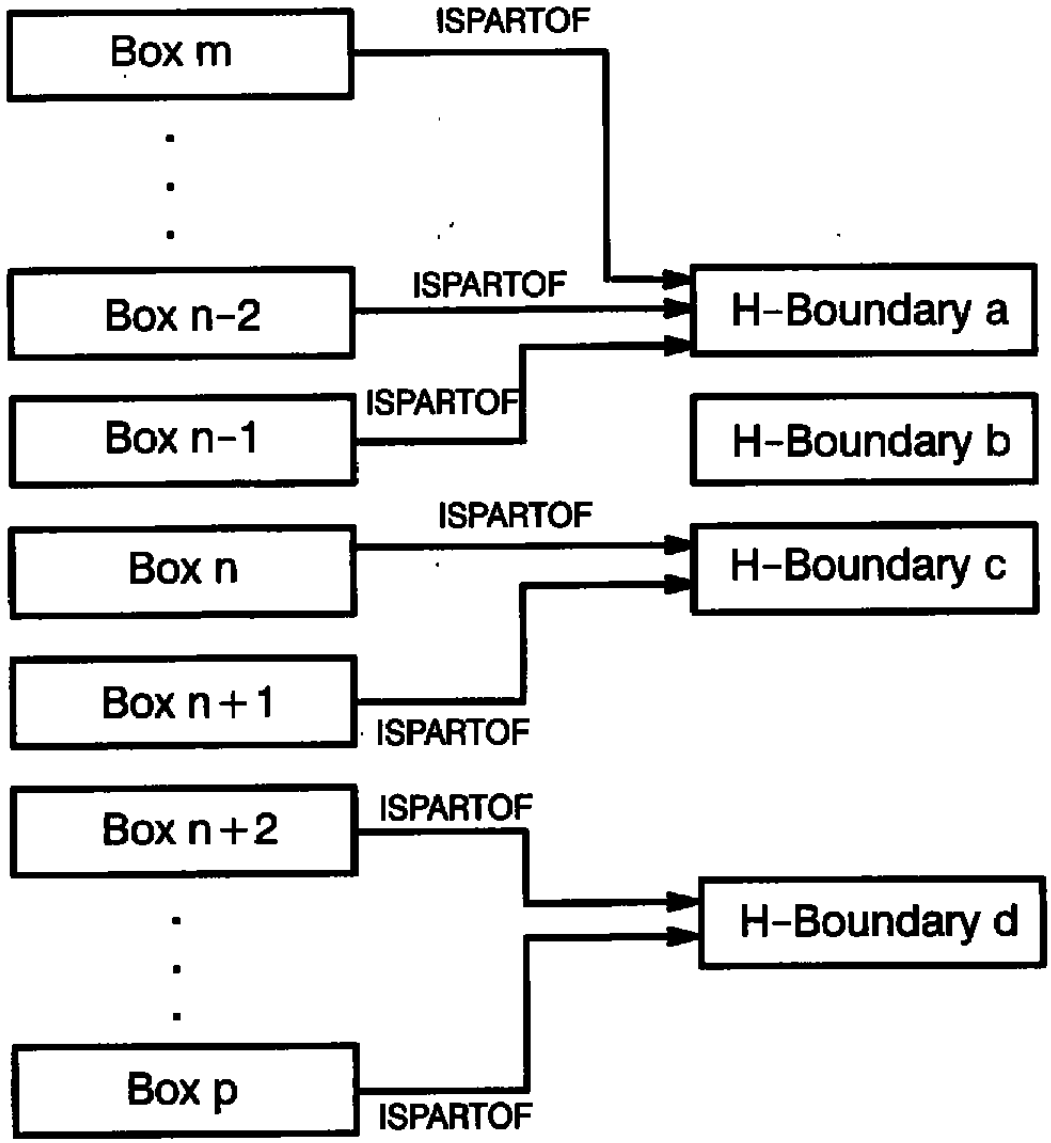


Figure 10. Box-hole creation boundary connections.

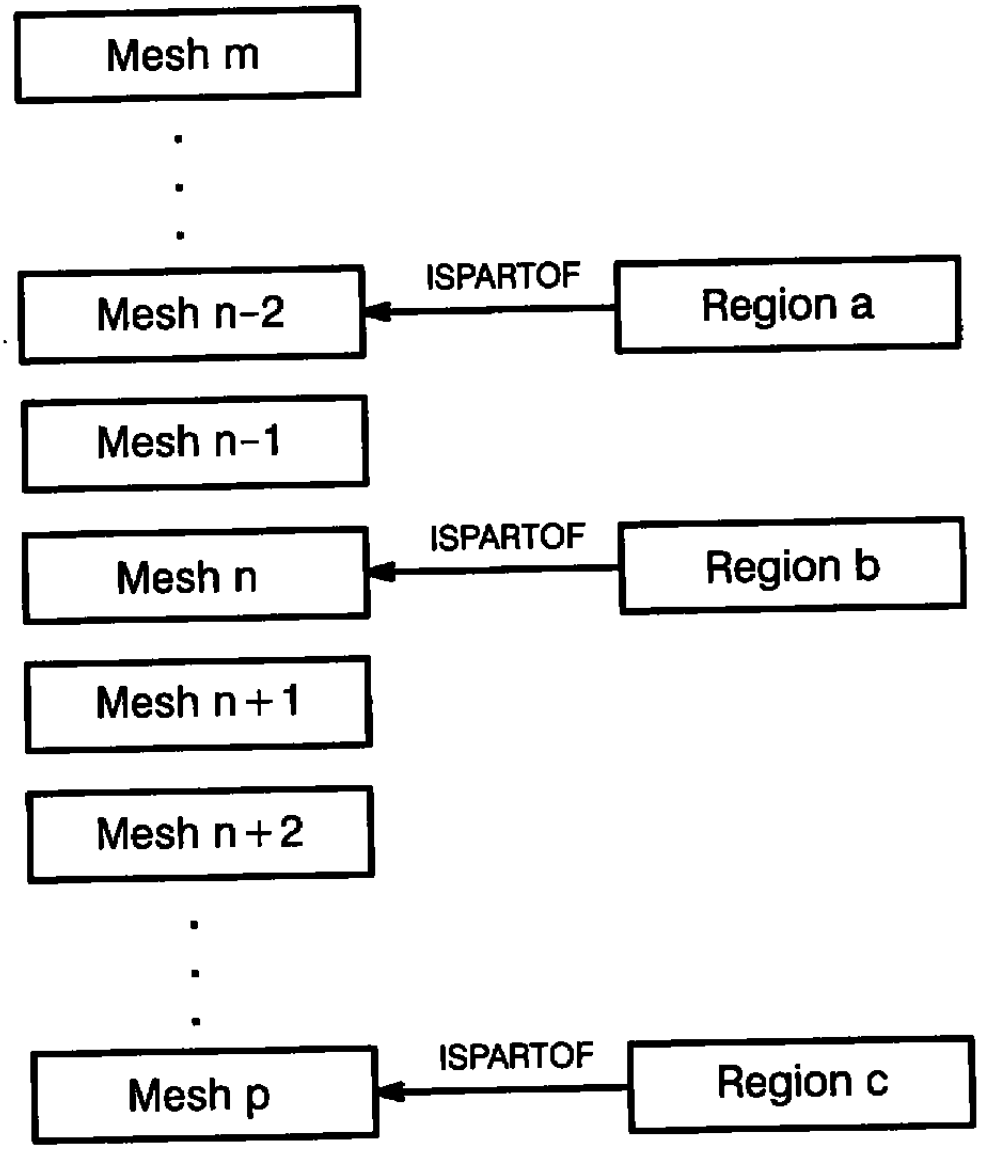


Figure 11. Region-mesh connections.

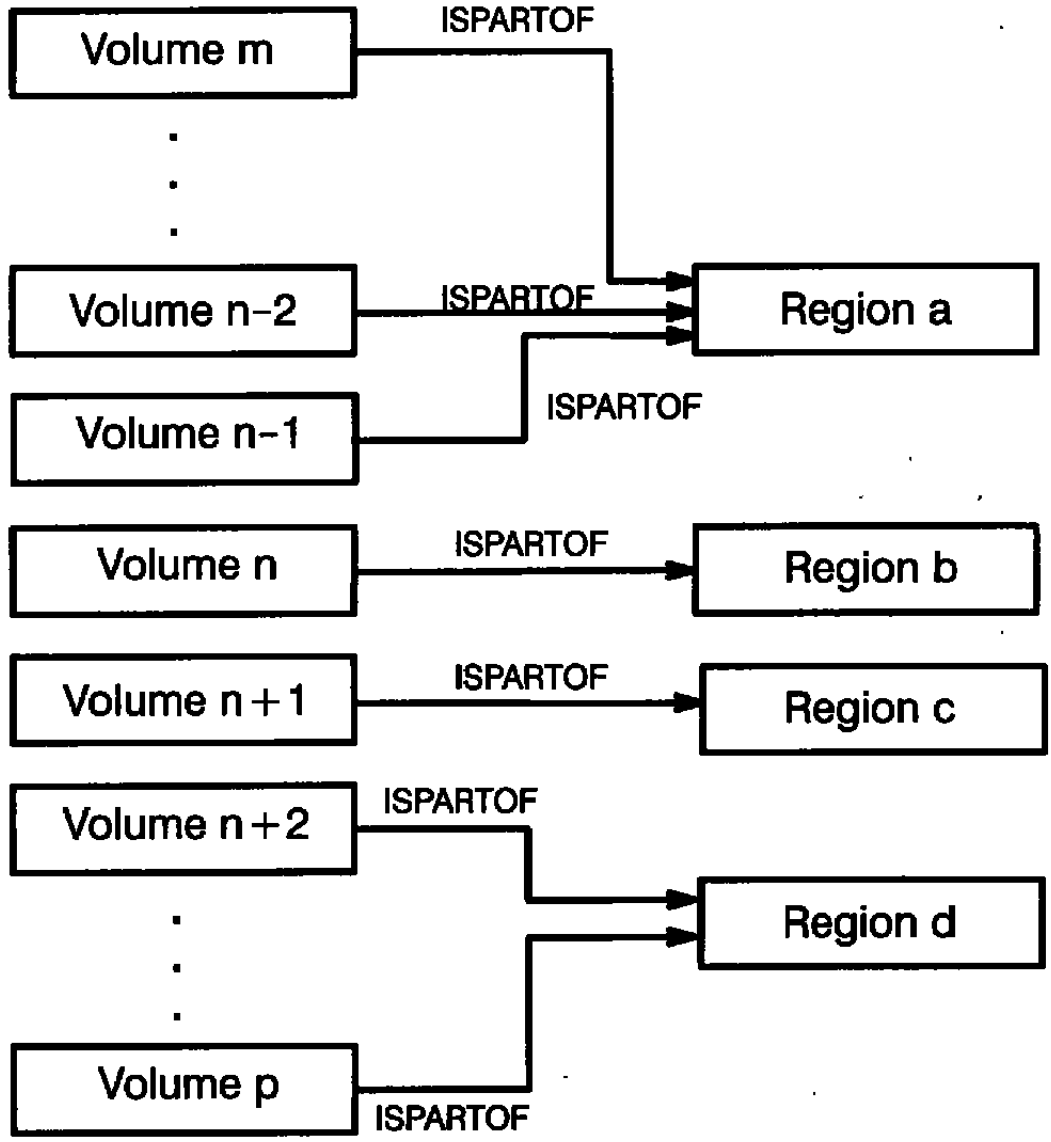
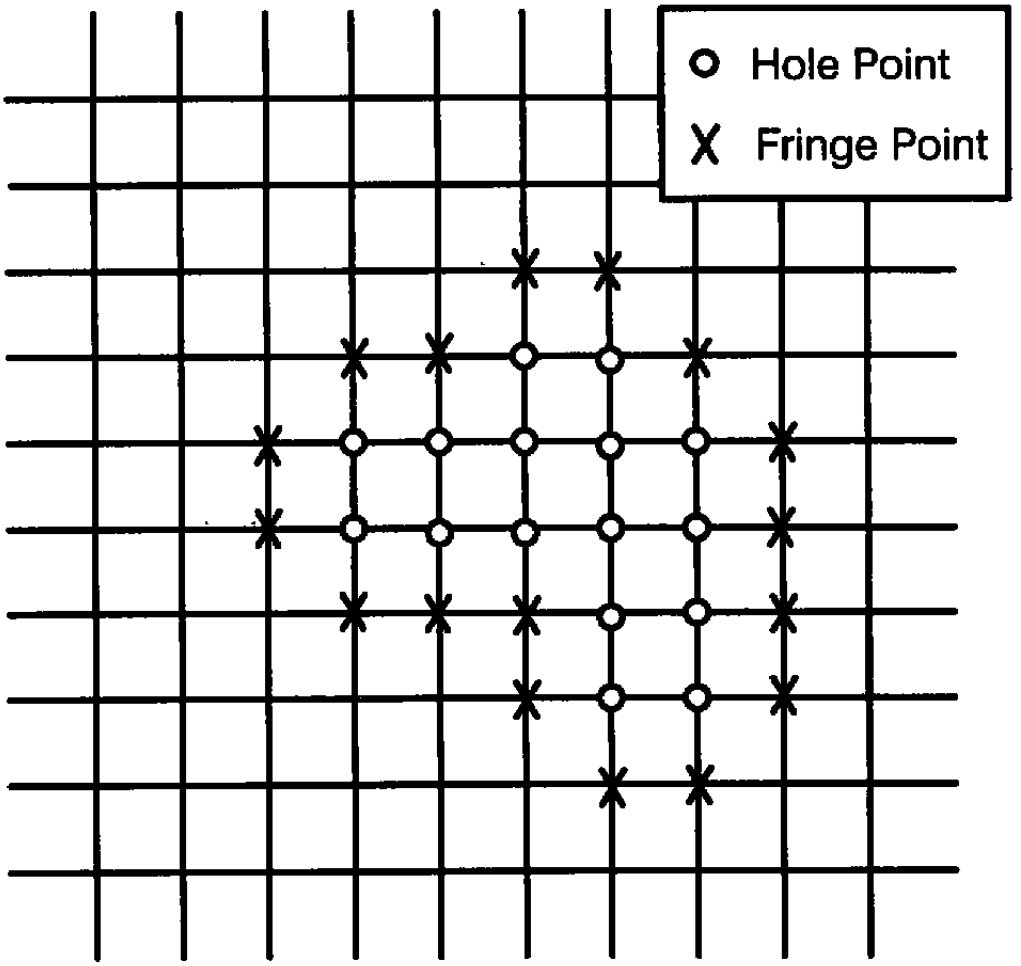
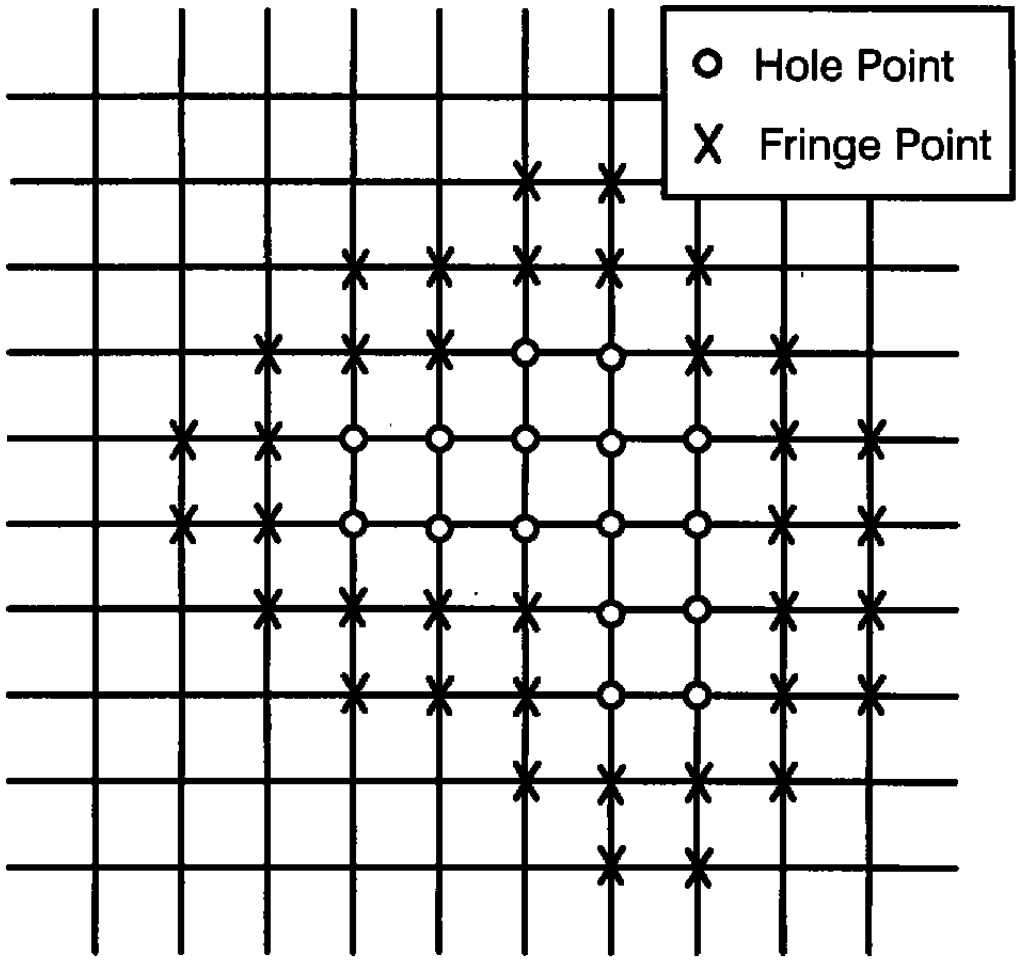


Figure 12. Volume-region connections.

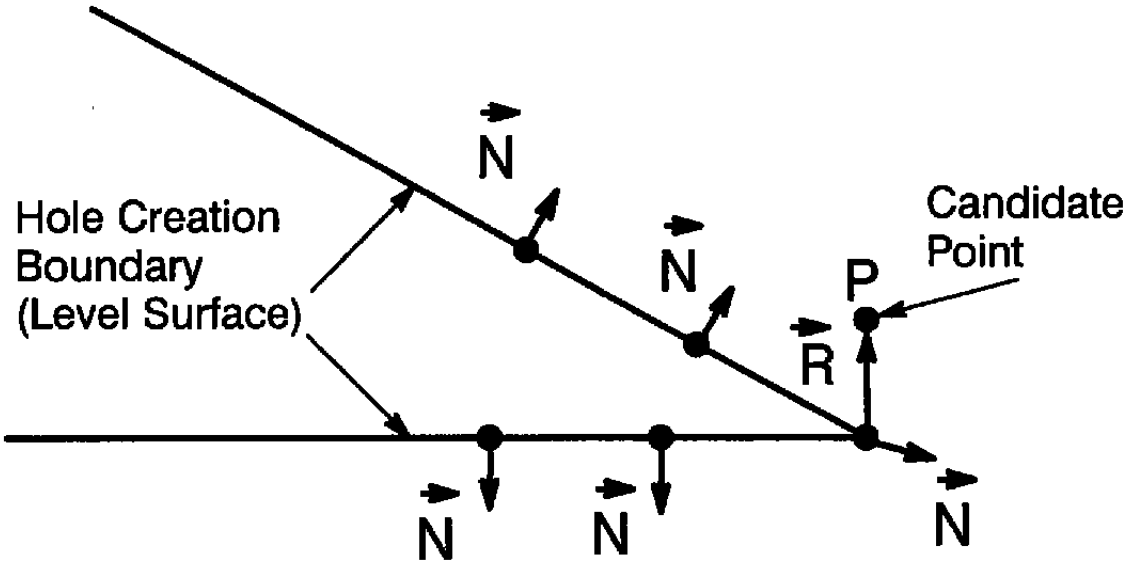


a. Single fringe

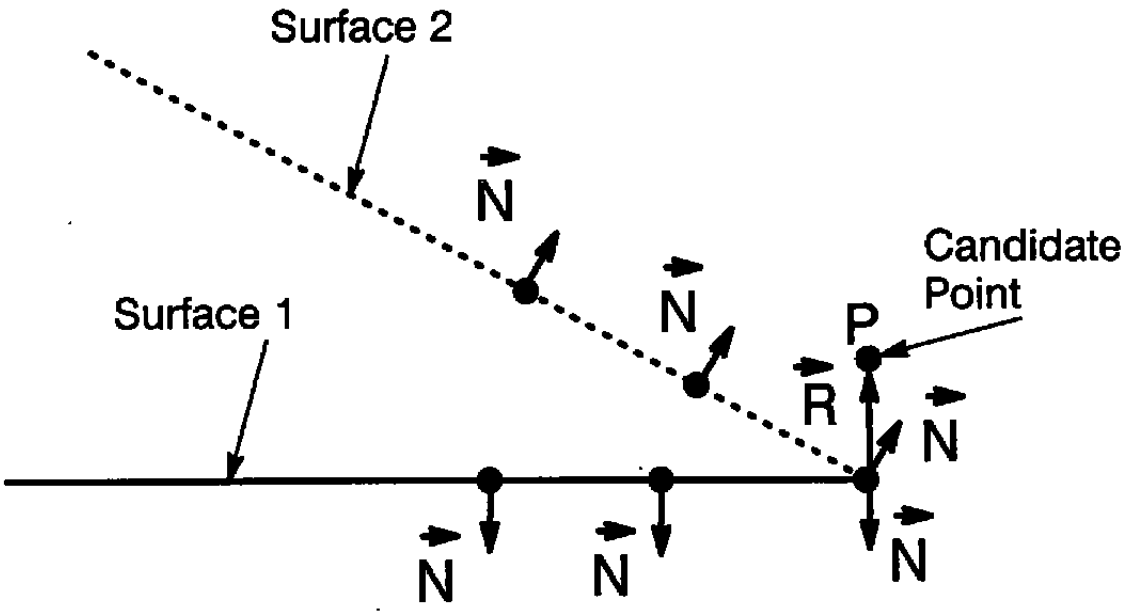
Figure 13. Hole boundary generation.



b. Double fringe
Figure 13. Concluded.

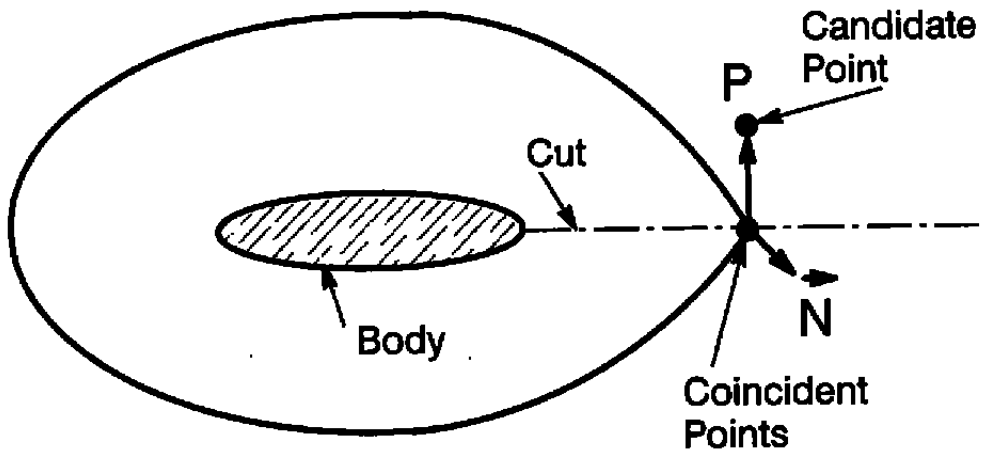


a. Problem: spurious hole points near sharp corner

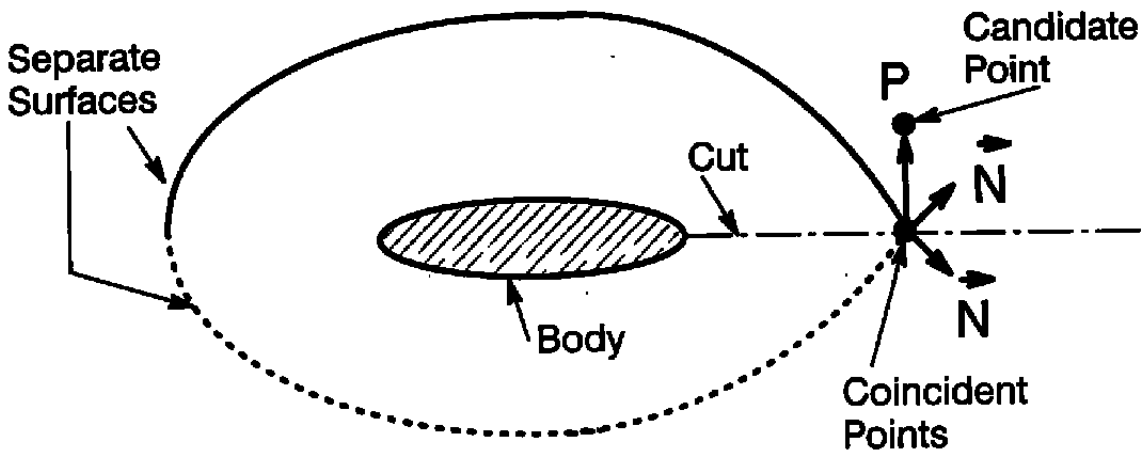


b. Solution: divide hole creation boundary into two surfaces

Figure 14. Hole creation boundary with sharp corner.

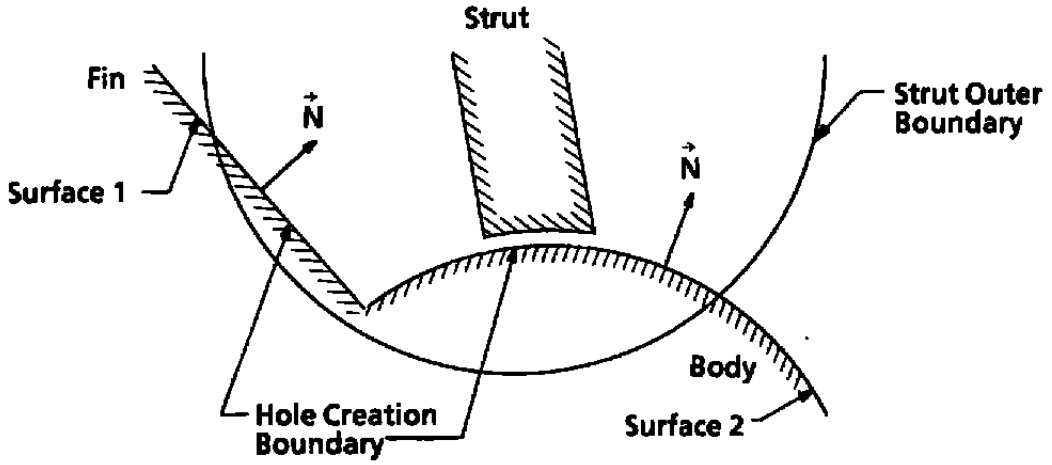


a. Problem: spurious hole points near coincident points

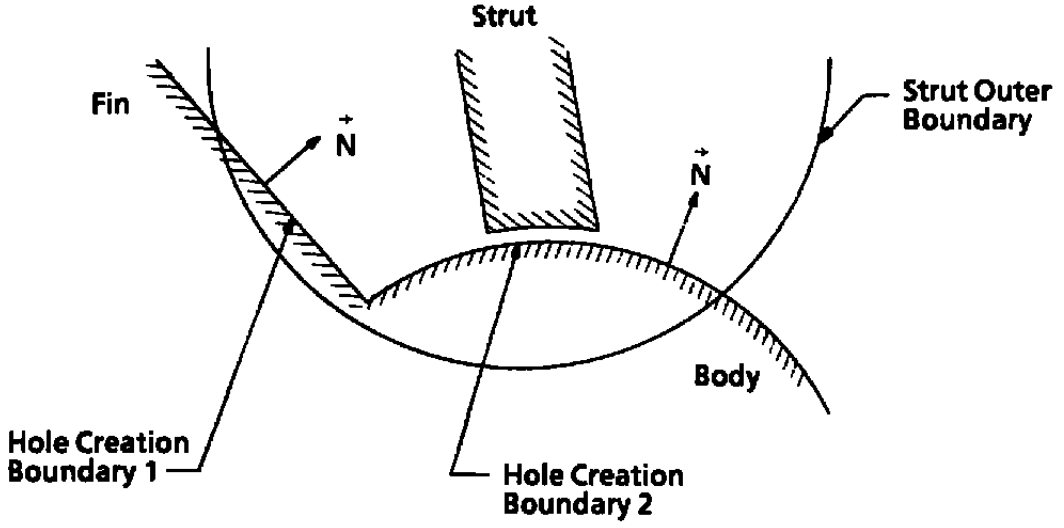


b. Solution: divide hole creation boundary into two surfaces

Figure 15. Single boundary with coincident points.



a. Problem: improper hole blanking



b. Solution: divide hole creation boundary into two boundaries

Figure 16. Concave hole creation boundary.

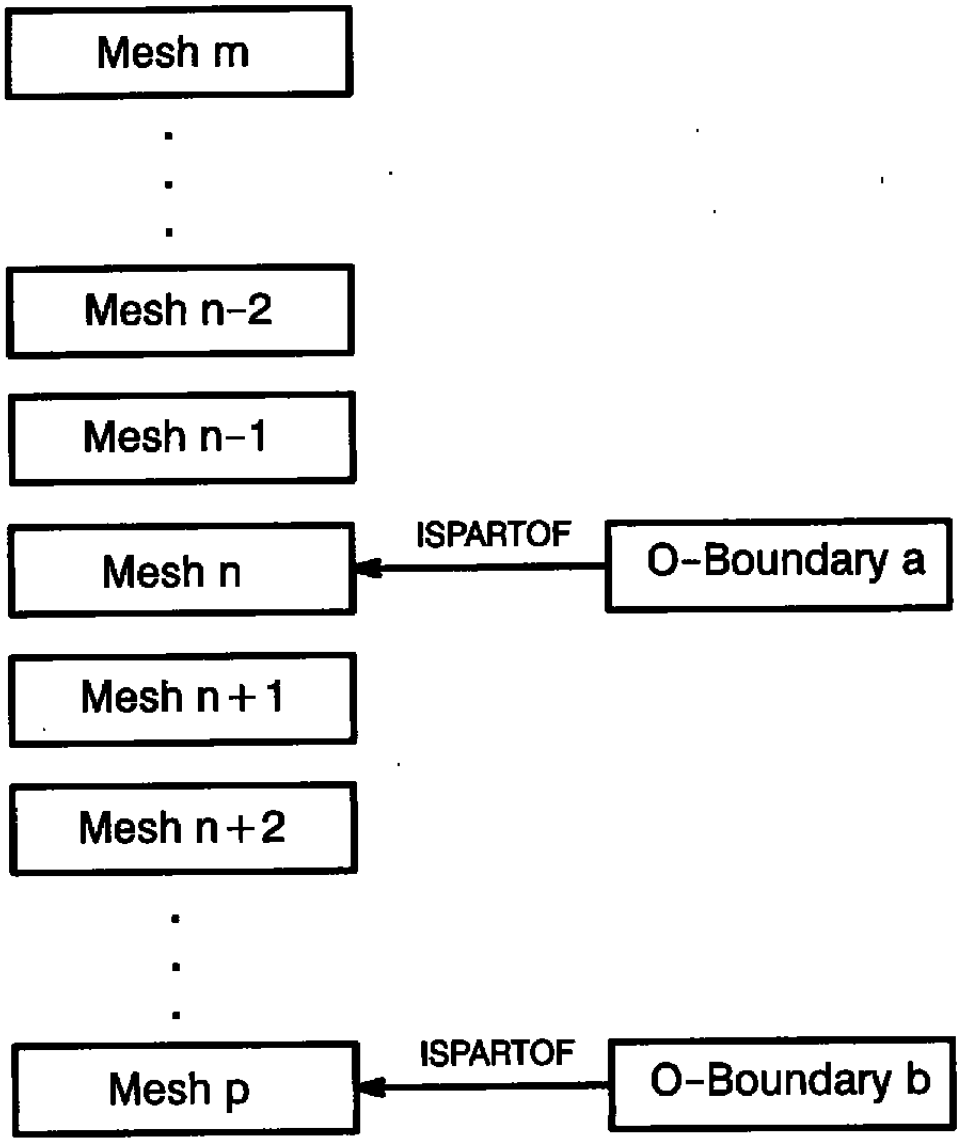


Figure 17. Outer boundary-mesh connections.

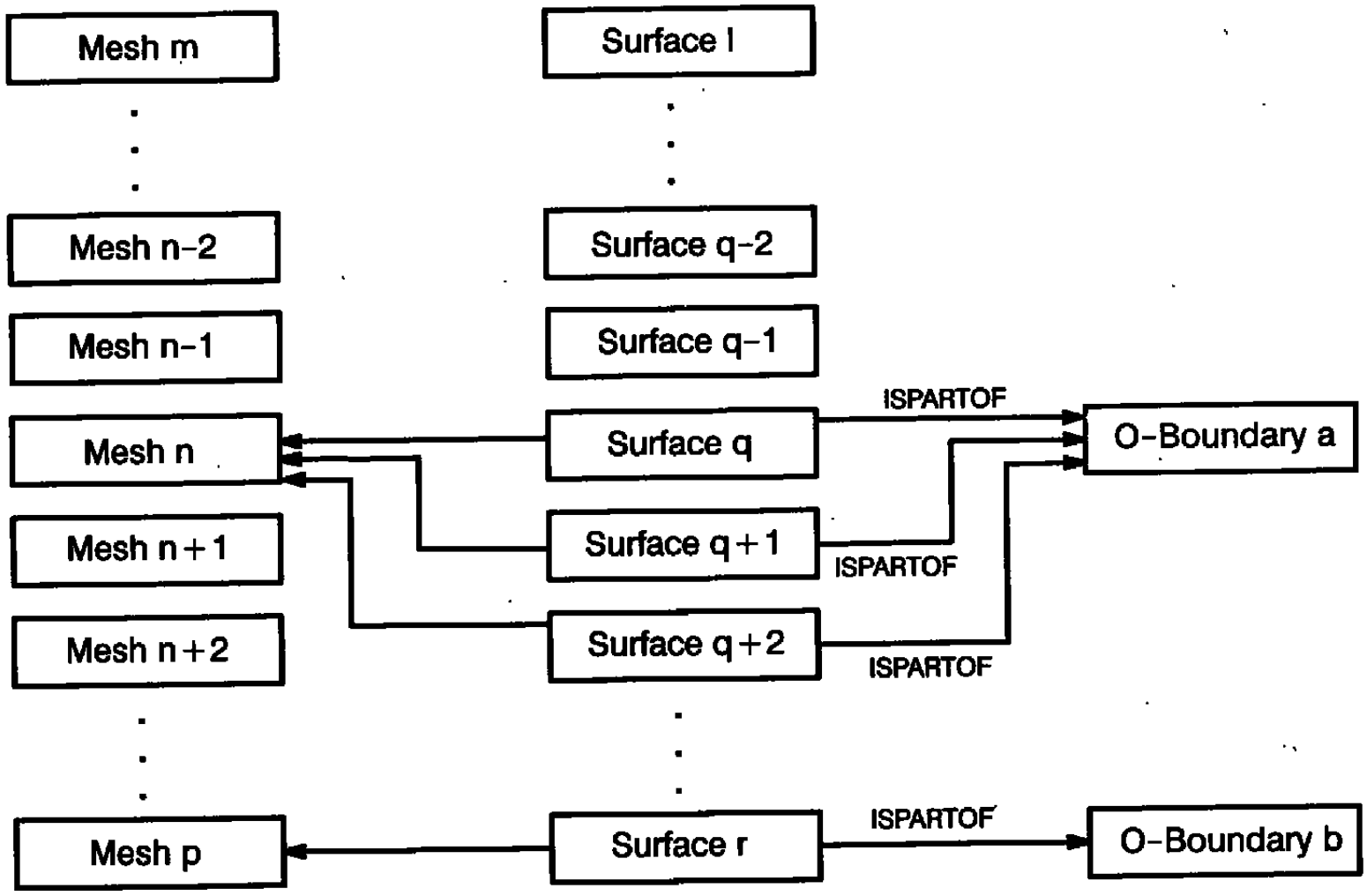


Figure 18. Surface-outer boundary connections.

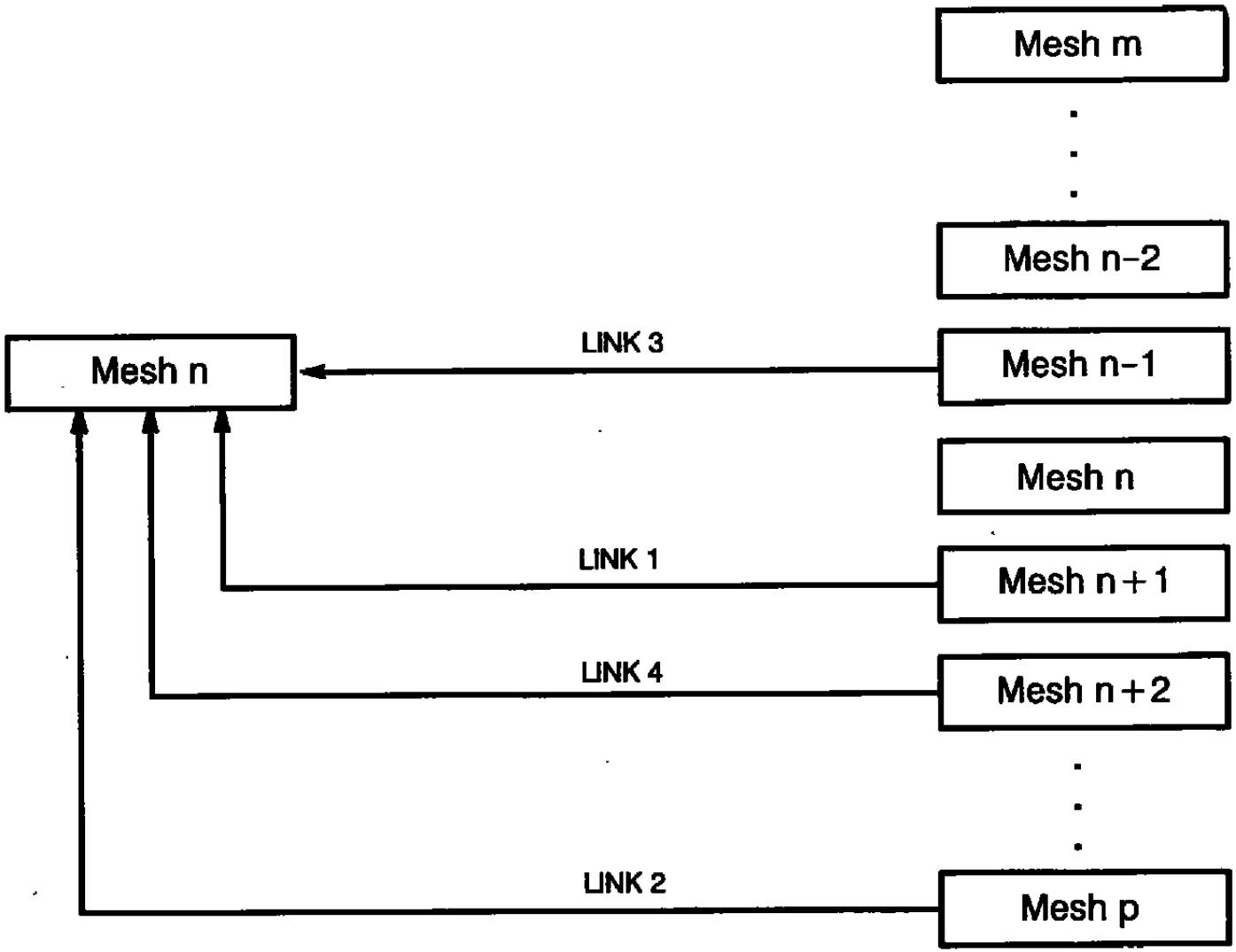


Figure 19. Mesh-mesh connections.

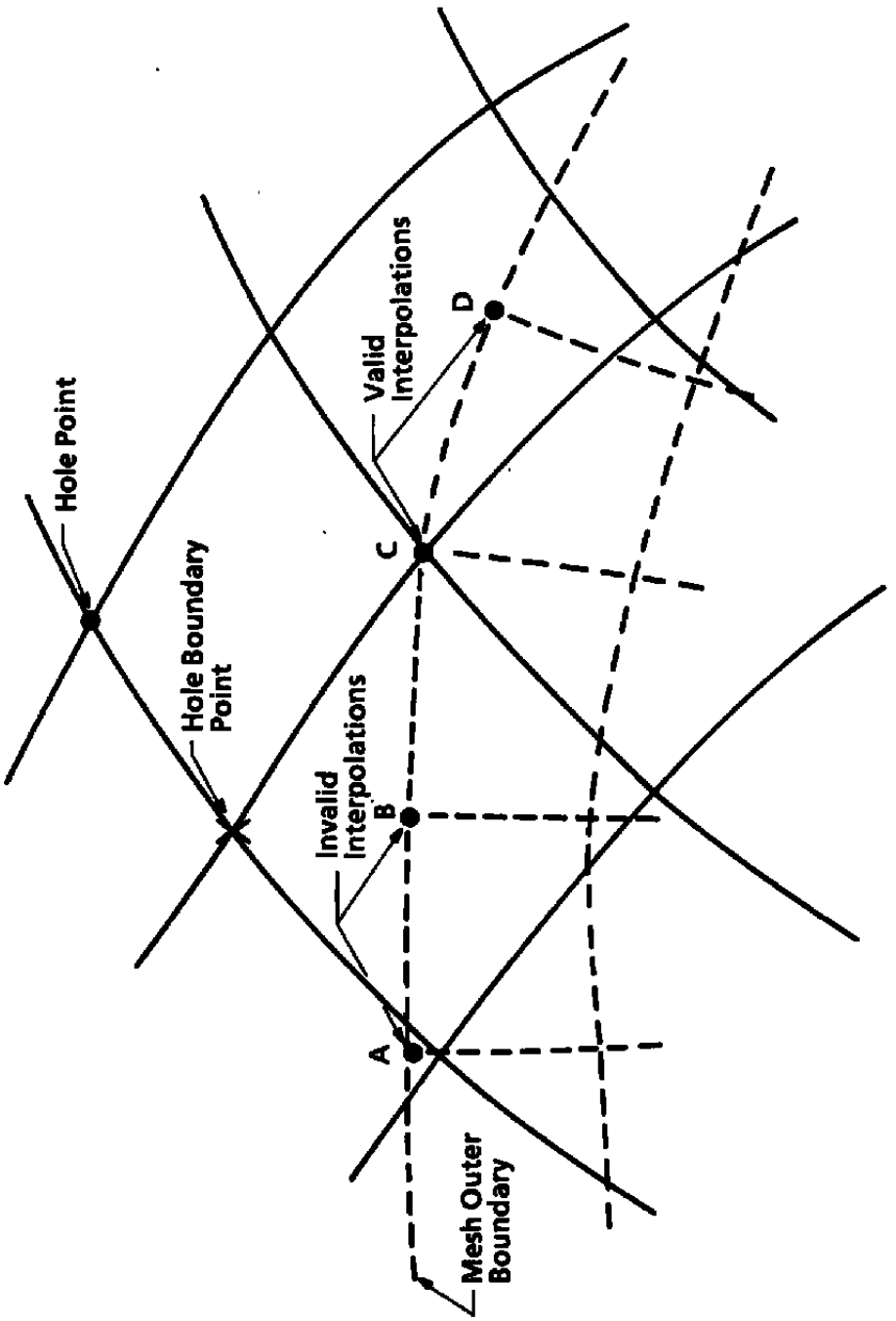
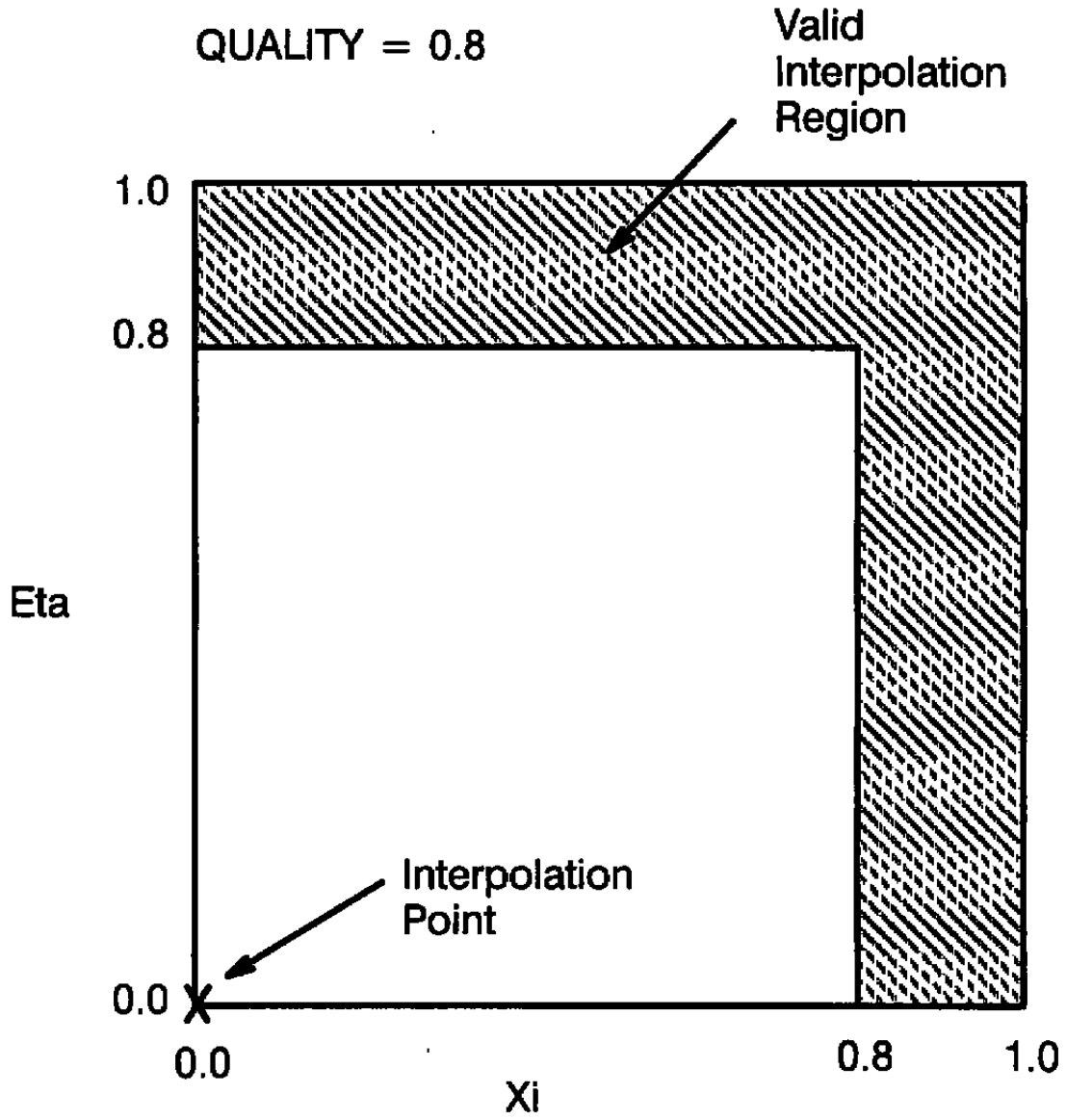
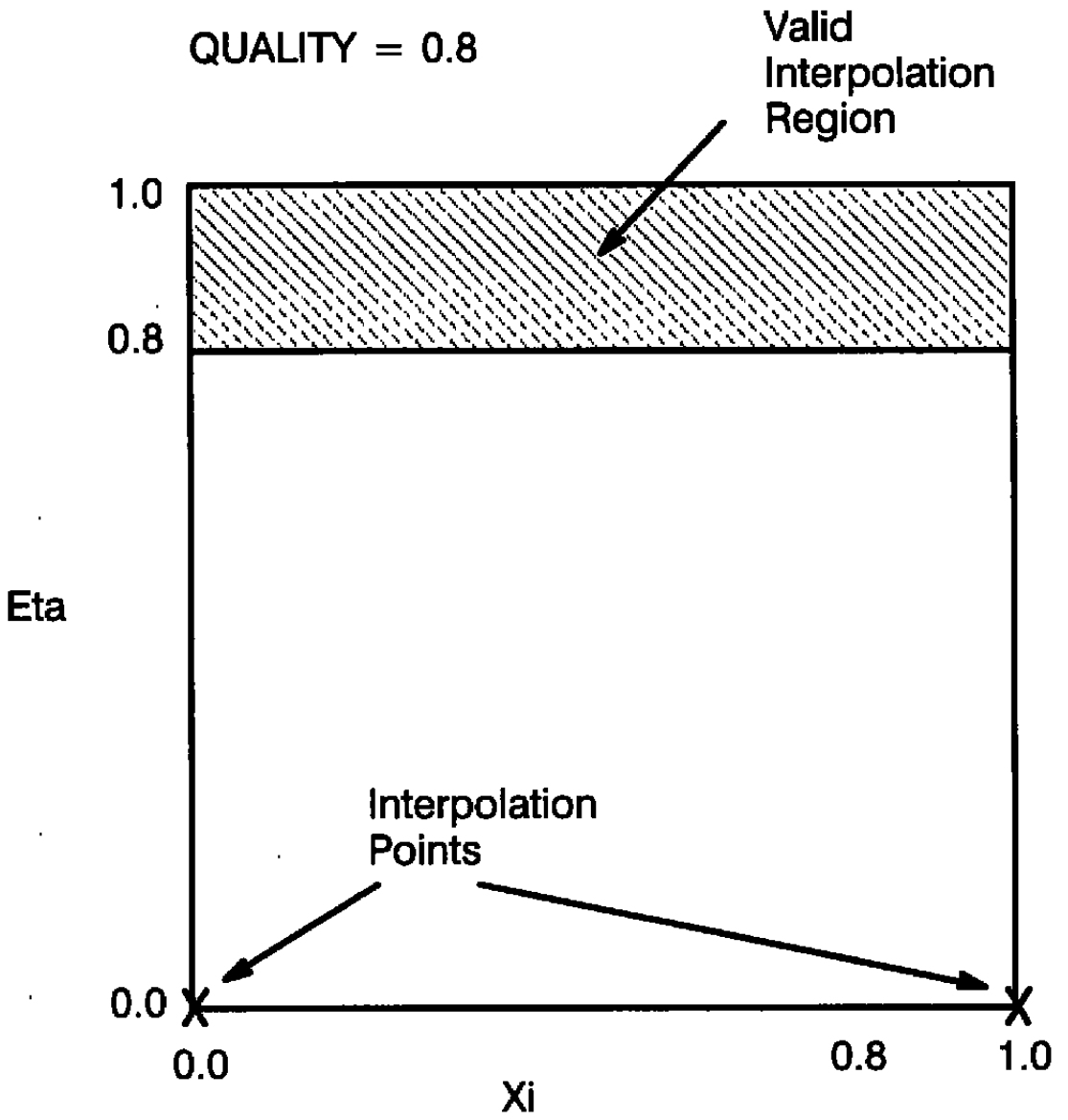


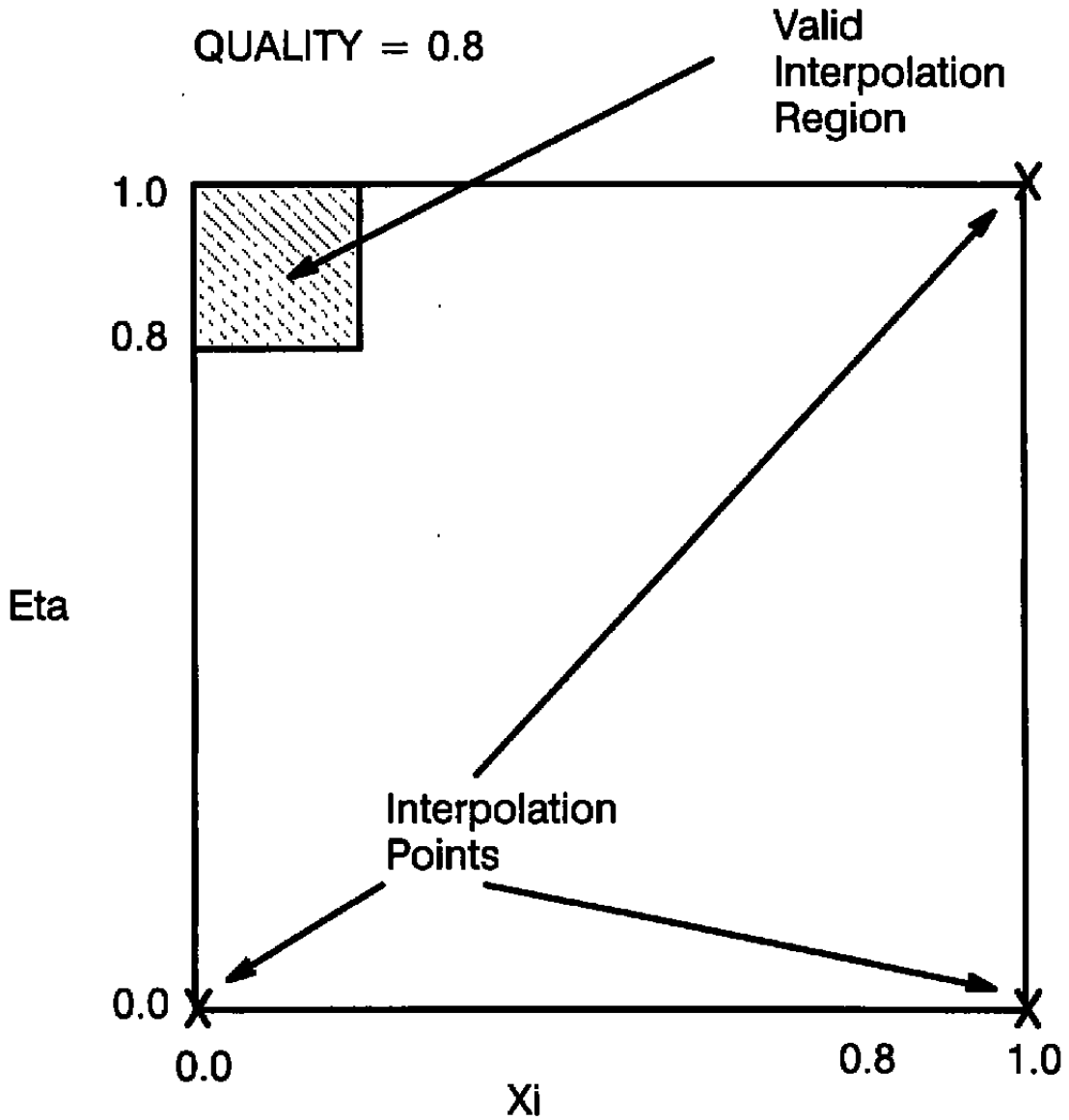
Figure 20. Valid and invalid interpolations.



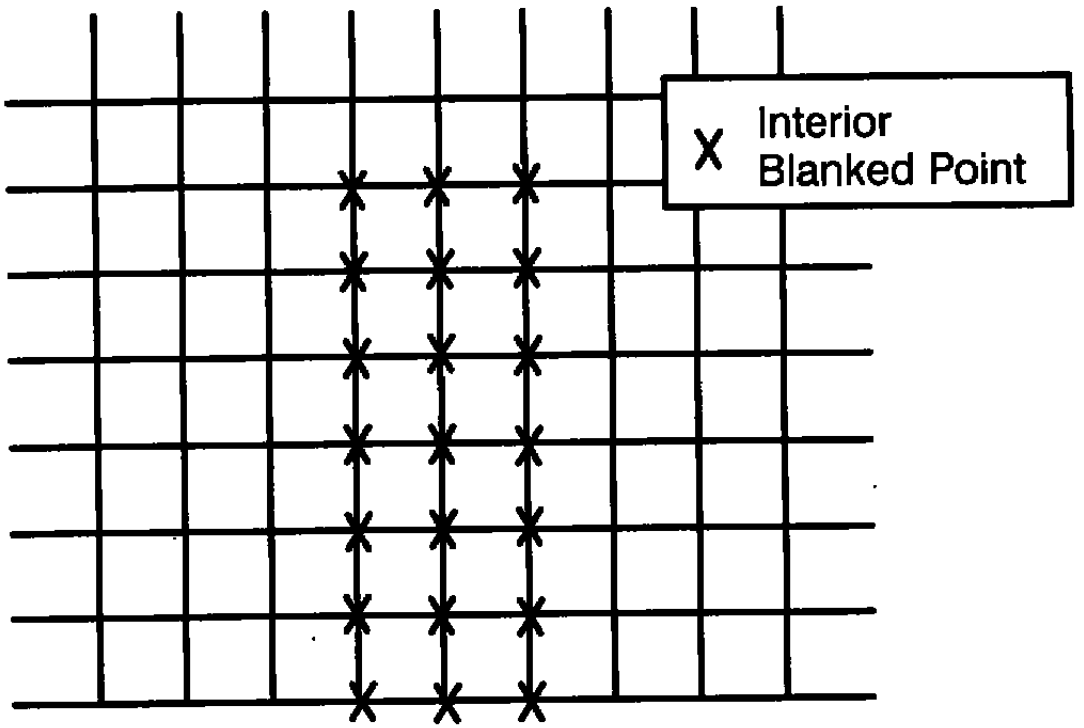
a. One blanked point in stencil
Figure 21. Interpolation quality.



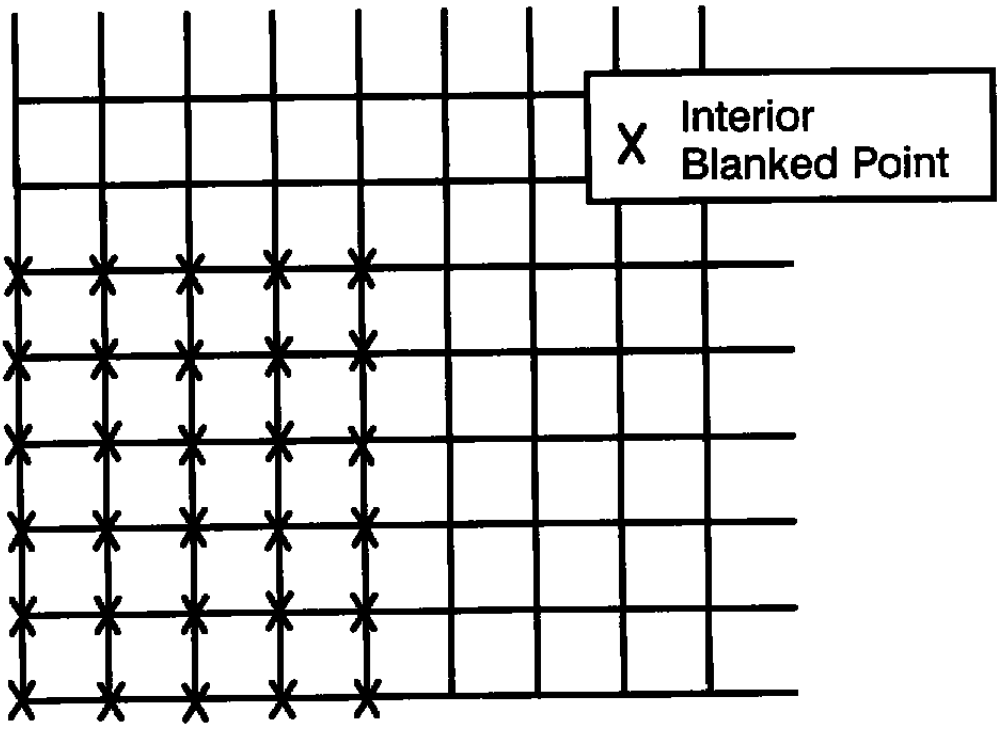
b. Two blanked points in stencil
Figure 21. Continued.



c. Three blanked points in stencil
Figure 21. Concluded.



a. Store fin



b. Corner

Figure 22. Interior blanking.

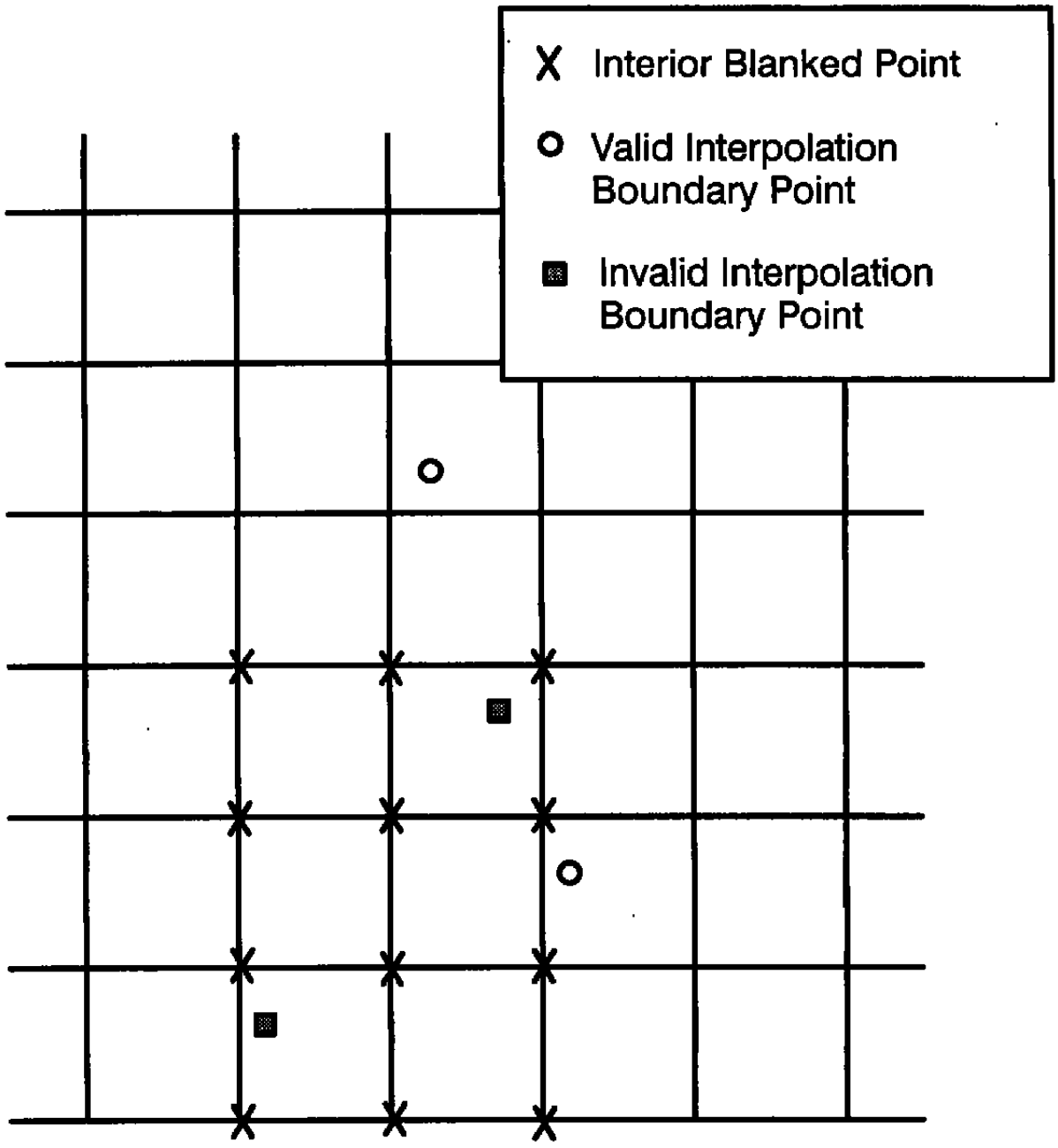
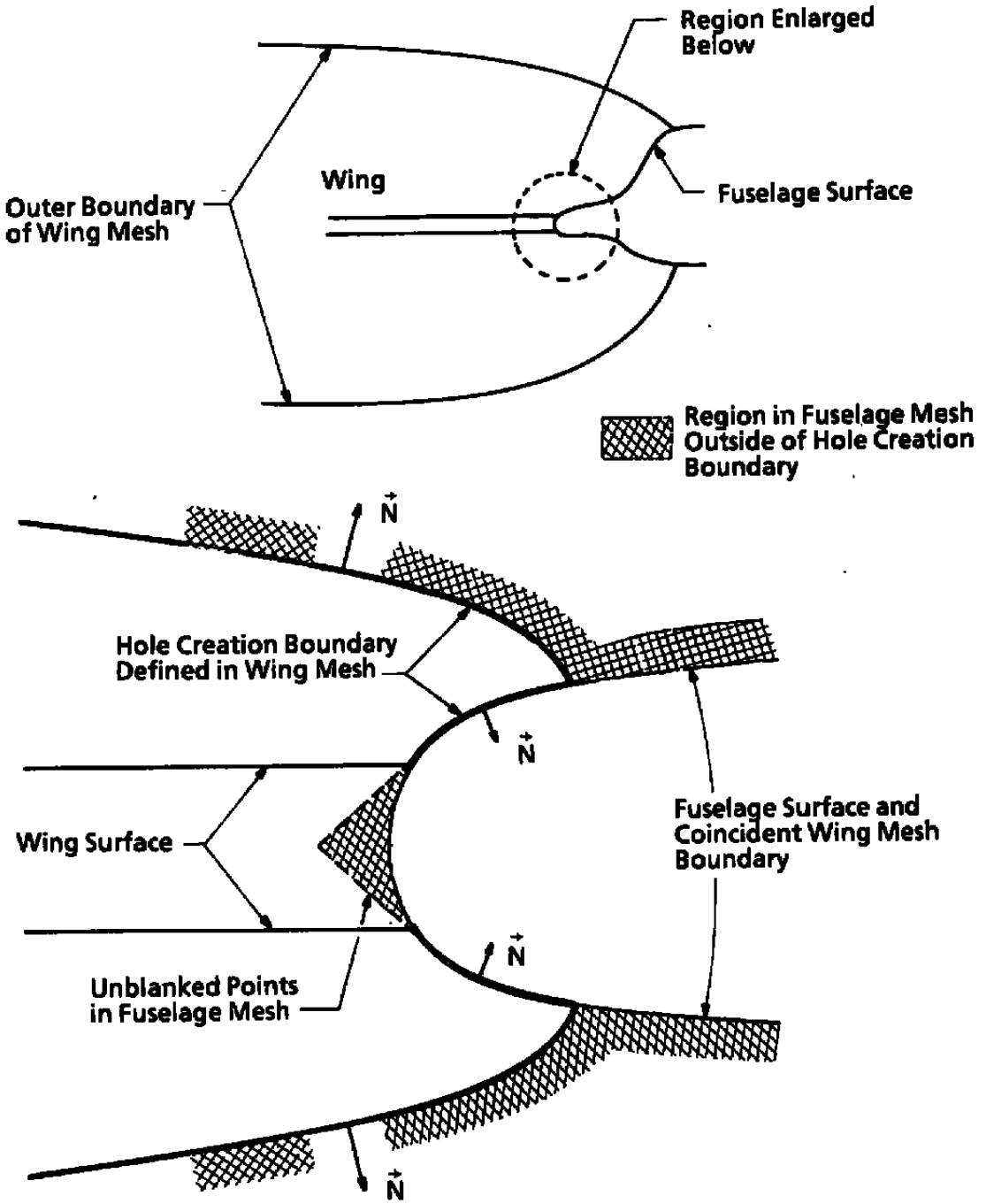
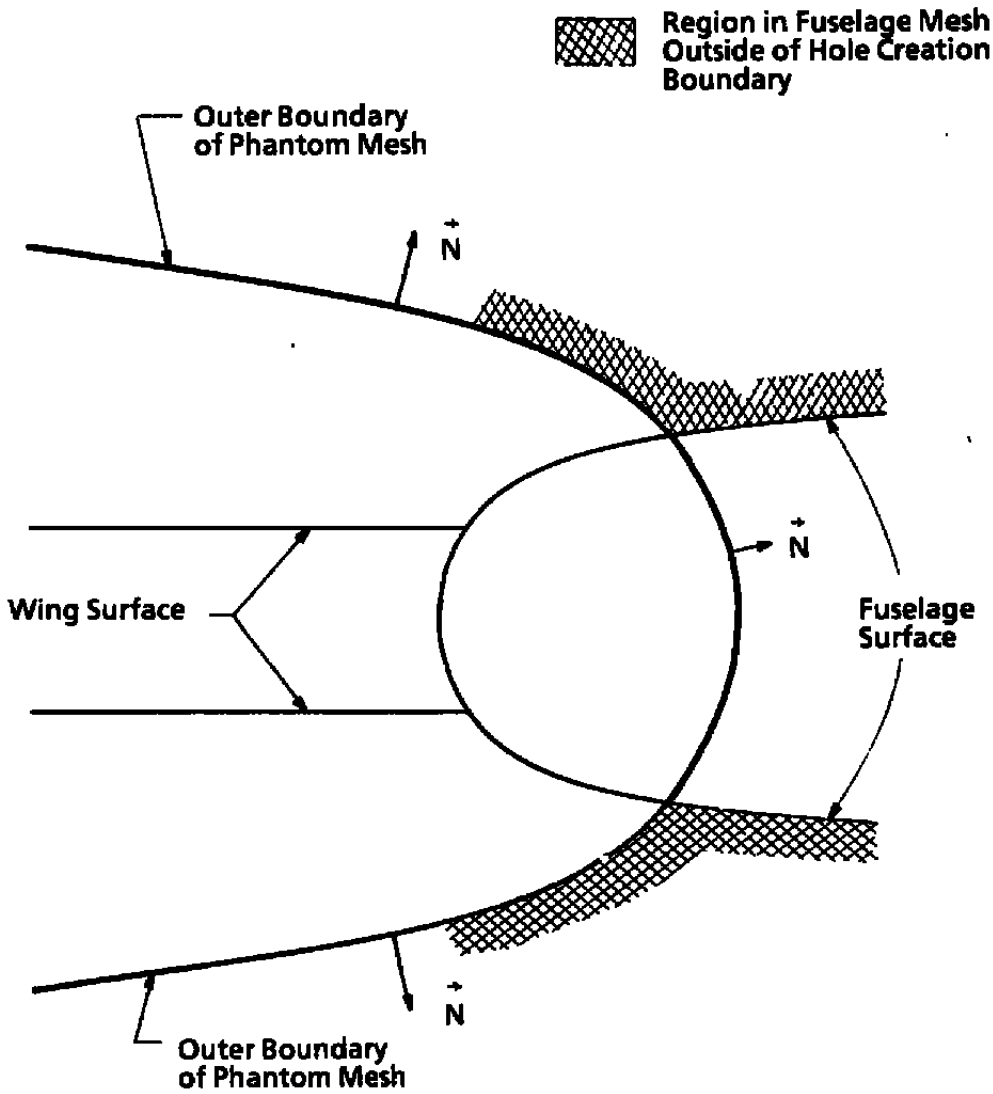


Figure 23. Valid and invalid interpolation near an interior blanked region.



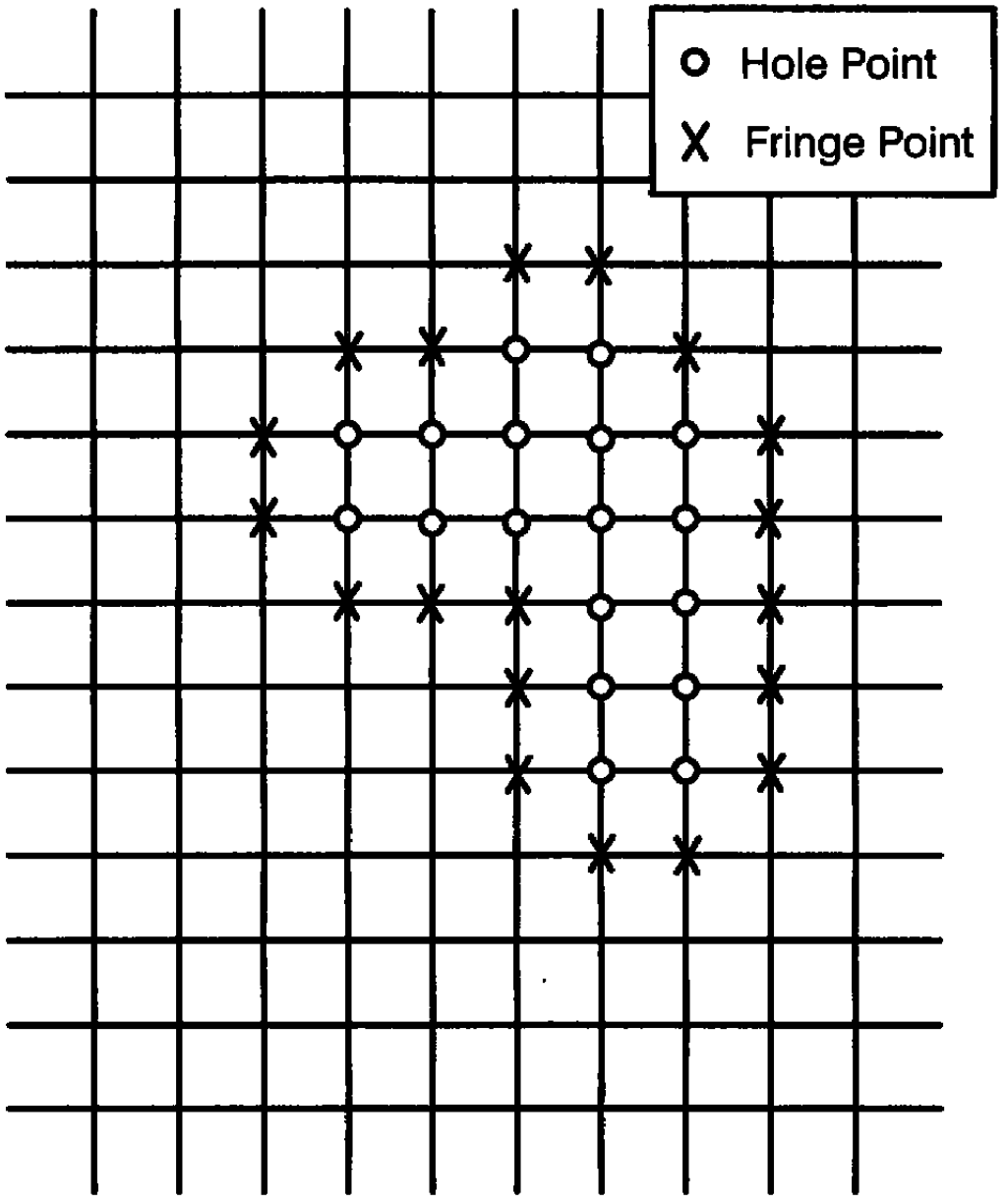
a. Problem: unblanked hole points near wing/body junction

Figure 24. Complex wing/body junction.



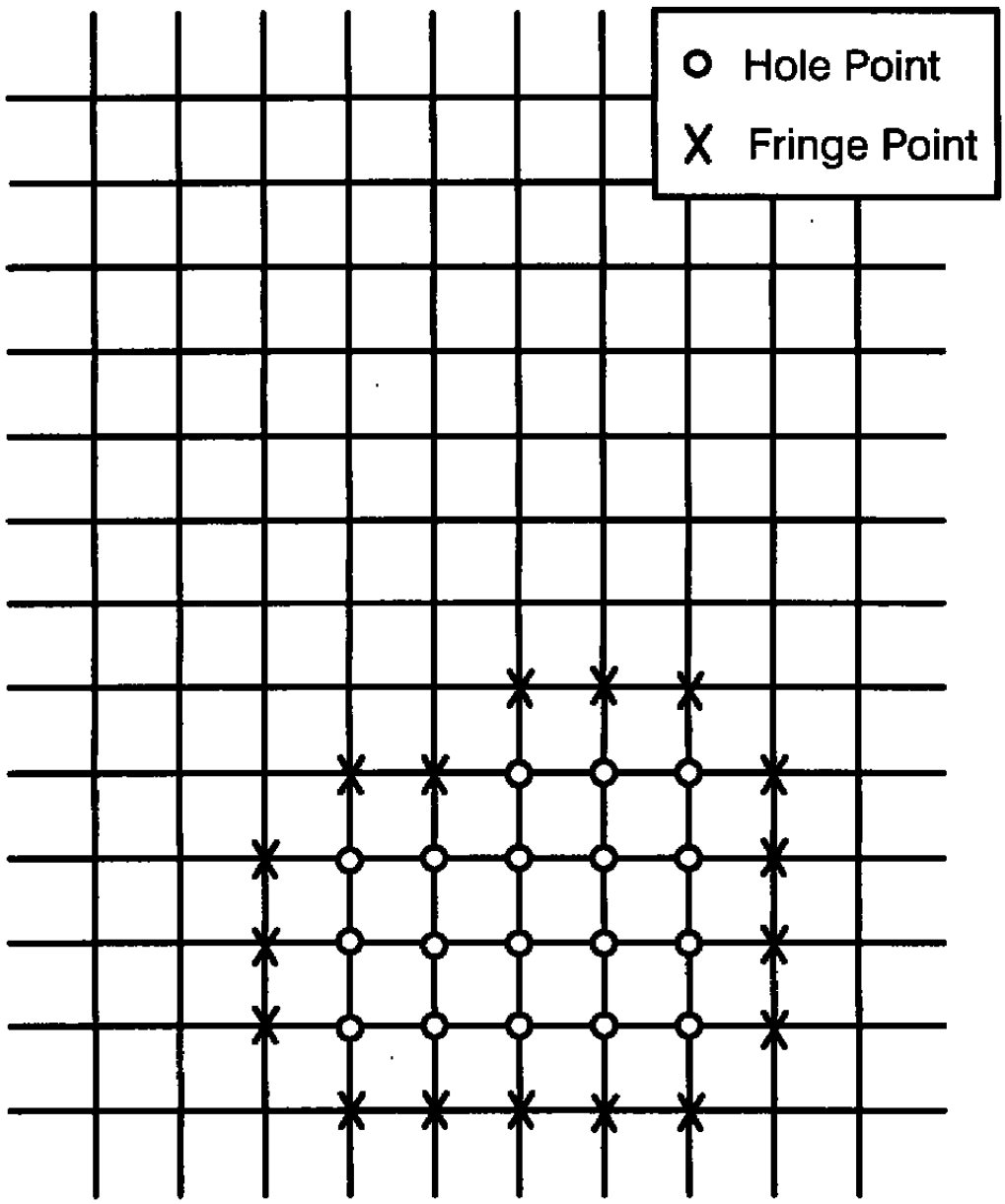
b. Solution: use of phantom mesh to create holes

Figure 24. Concluded.

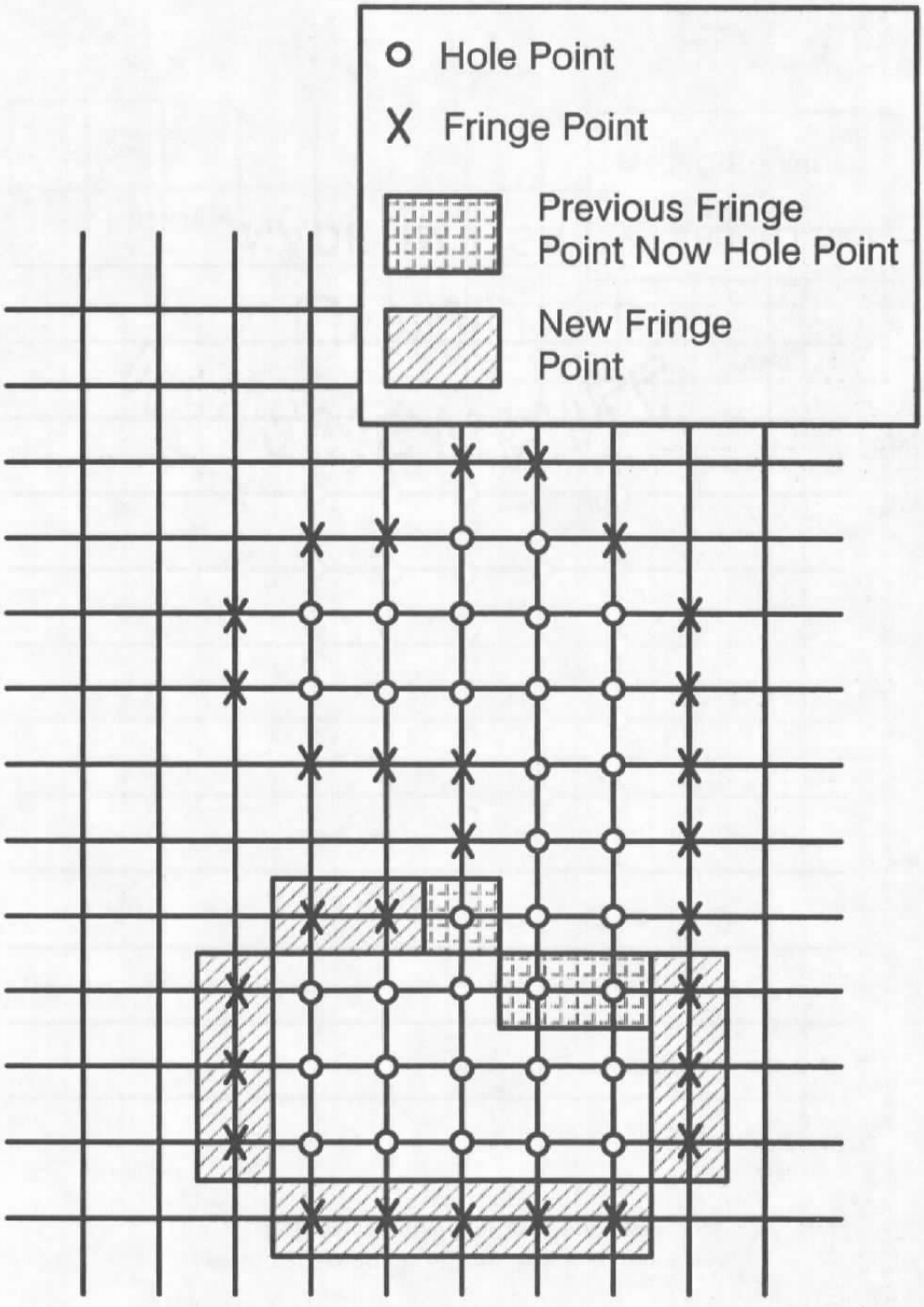


a. Previous hole and hole boundary

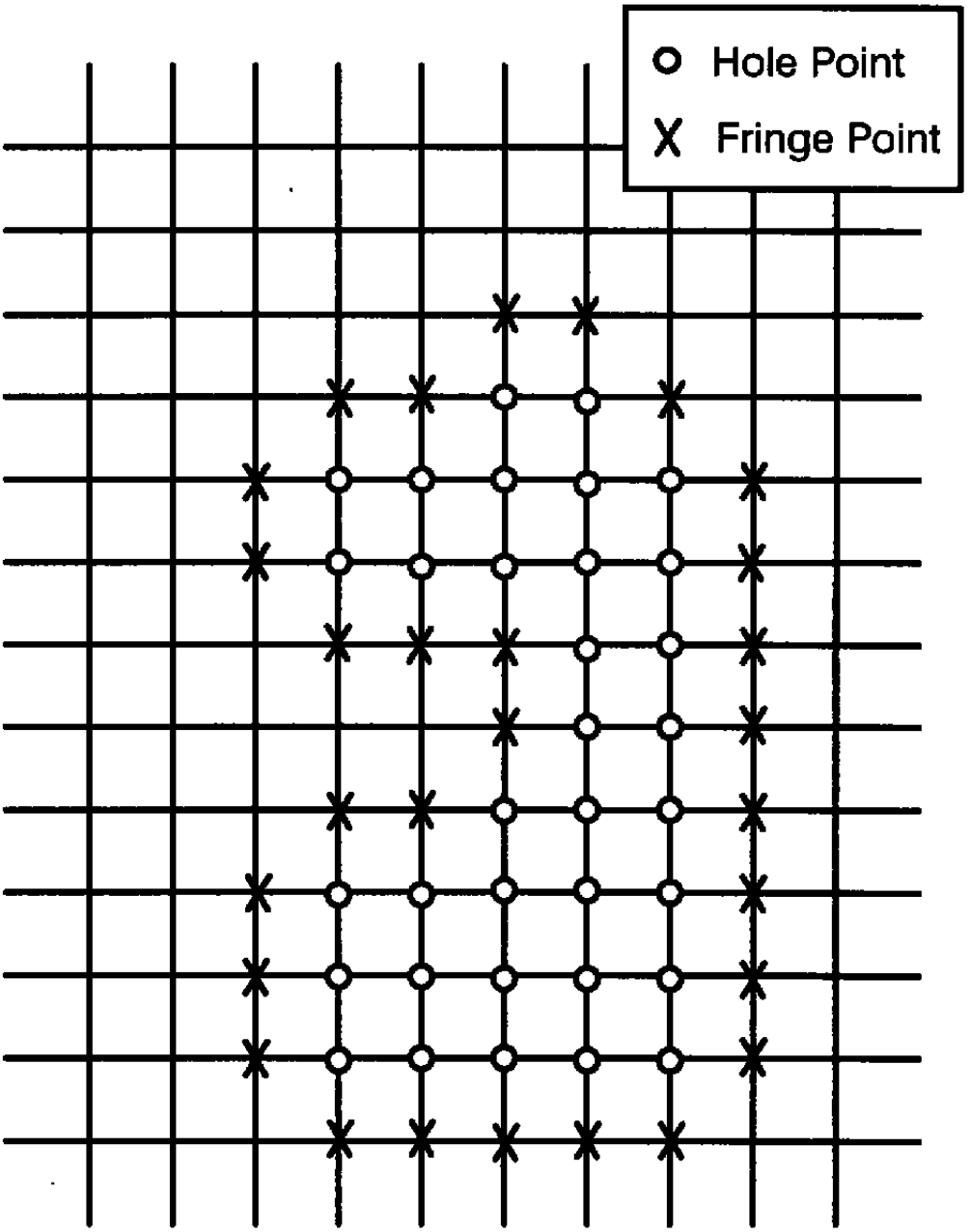
Figure 25. Addition of hole and hole boundary.



b. New hole and hole boundary
Figure 25. Continued.

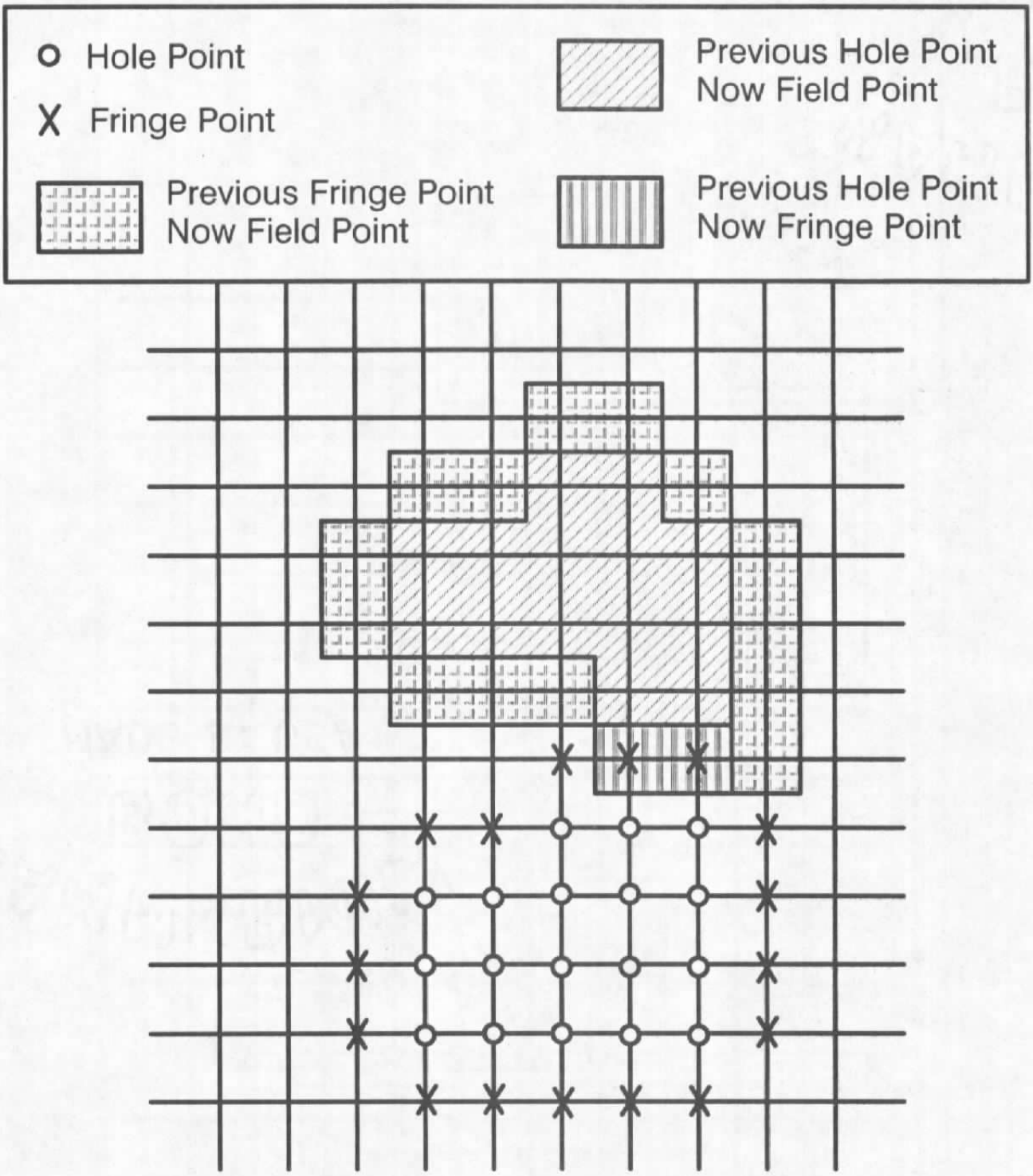


c. Combined hole and hole boundary
Figure 25. Concluded.



a. Hole and hole boundary

Figure 26. Subtraction of hole and hole boundary.



b. Resulting hole and hole boundary
Figure 26. Concluded.

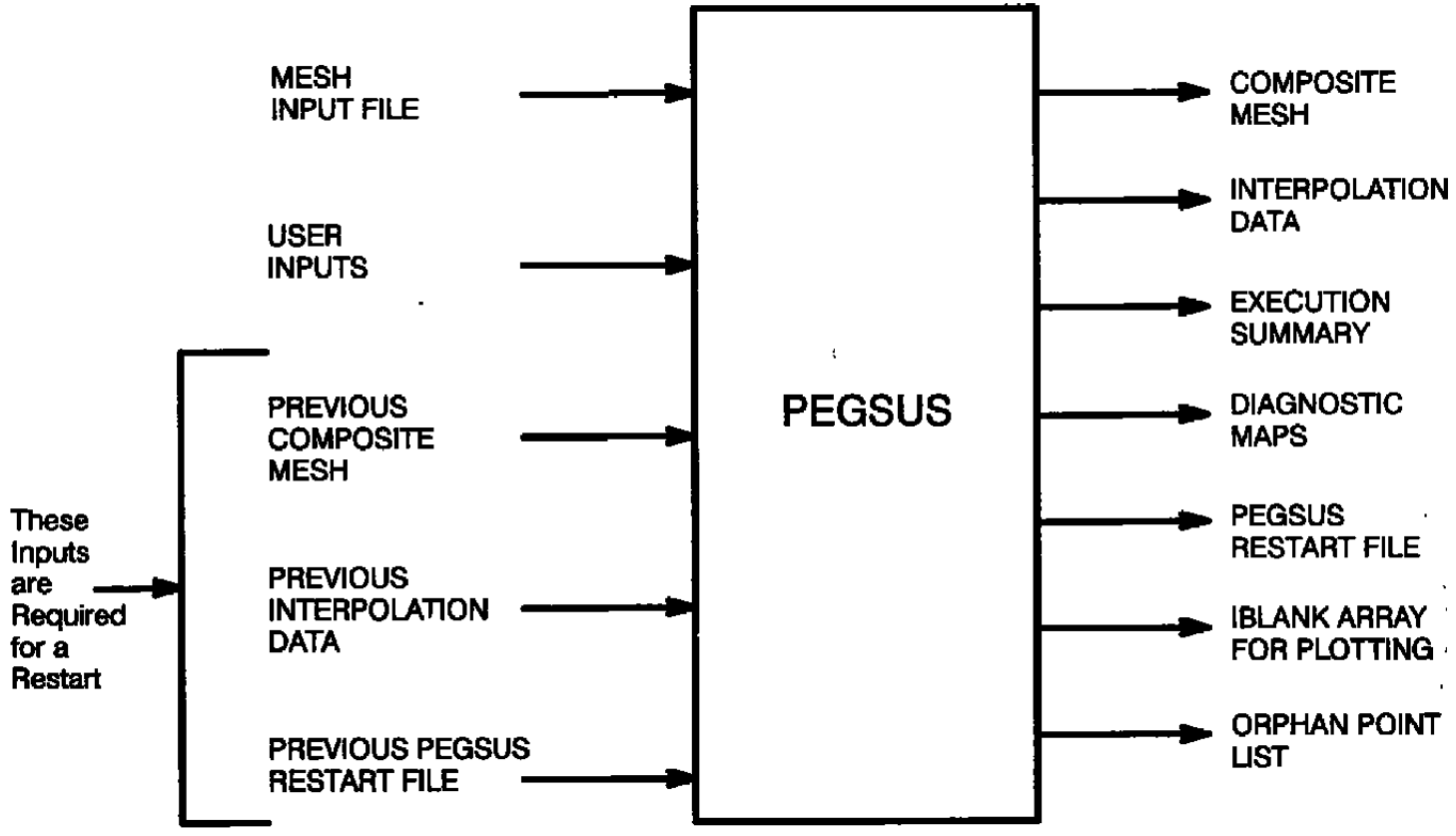


Figure 27. PEGSUS 4.0 input and output files.

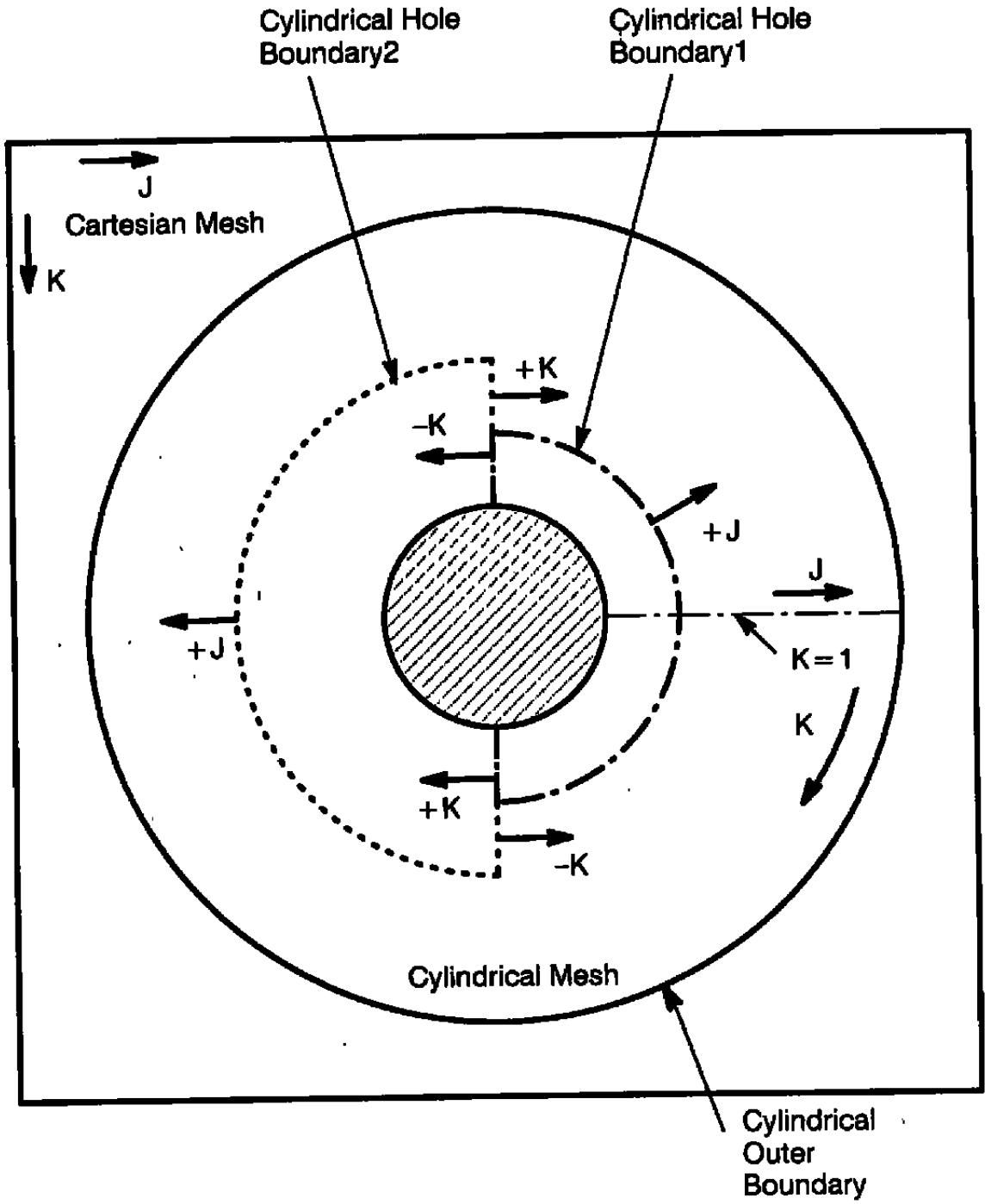


Figure 28. Graphical depiction of indirect hole creation using $BOUNDARY and $SURFACE.

```
C              PEGSUS 4.0 INPUT
C            CARTESIAN GRID / CYLINDER
C    EXAMPLE 4.1.1 - INDIRECT HOLE SPECIFICATION

C  DEFINITION OF GLOBAL PARAMETERS
 $GLOBAL
   FRINGE = 2,
   $END

C MESH DEFINITIONS

 $MESH NAME = 'CARTESIAN GRID',
   LINK = ,
   $END

 $MESH NAME = 'CYLINDER GRID',
   LINK = 'CARTESIAN GRID',
   X0   = 20.0,
   Y0   = 20.0,
   $END

C BOUNDARY DEFINITIONS CONSIST OF TWO HOLE
C CREATION BOUNDARIES AND AN OUTER BOUNDARY.

  $BOUNDARY NAME = 'CYLINDER HOLE BOUNDARY1',
   ISPARTOF = 'CYLINDER GRID',
   MHOLEIN = 'CARTESIAN GRID',
   $END

  $BOUNDARY NAME = 'CYLINDER HOLE BOUNDARY2',
   ISPARTOF = 'CYLINDER GRID',
   MHOLEIN = 'CARTESIAN GRID',
   $END

  $BOUNDARY NAME = 'CYLINDER OUTER BOUNDARY',
   ISPARTOF = 'CYLINDER GRID',
   $END

C SURFACE DEFINITIONS

C FOUR SURFACES ARE NEEDED TO DESCRIBE
C CYLINDER HOLE BOUNDARY1.

C THE FIRST TWO SURFACES DESCRIBE THE HALF
C CYLINDER OF SMALLER RADIUS.
C THE REMAINING TWO SURFACES DESCRIBE THE
C RECTANGULAR SEGMENTS.

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY1',
   JRANGE = 11,11,
   KRANGE = 1,11,
   LRANGE = 1,5,
   NVOUT = '+J',
   $END

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY1',
   JRANGE = 11,11,
   KRANGE = 31,41,
   LRANGE = 1,5,
   NVOUT = '+J',
   $END

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY1',
   JRANGE = 1,11,
   KRANGE = 11,11,
   LRANGE = 1,5,
```

a. Page 1

Figure 29. Input for indirect hole creation using $BOUNDARY and $SURFACE.

```
   NVOUT = '+K',
   $END

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY1',
   JRANGE = 1,11,
   KRANGE = 31,31,
   LRANGE = 1,5,
   NVOUT = '-K',
   $END

C THREE SURFACES ARE NEEDED TO DESCRIBE
C CYLINDER HOLE BOUNDARY2.

C THE FIRST SURFACE DESCRIBES THE HALF
C CYLINDER OF LARGER RADIUS.
C THE REMAINING TWO SURFACES DESCRIBE THE
C RECTANGULAR SEGMENTS.

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY2',
   JRANGE = 21,21,
   KRANGE = 11,31,
   LRANGE = 1,5,
   NVOUT = '+J',
   $END

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY2',
   JRANGE = 1,21,
   KRANGE = 11,11,
   LRANGE = 1,5,
   NVOUT = '-K',
   $END

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY2',
   JRANGE = 1,21,
   KRANGE = 31,31,
   LRANGE = 1,5,
   NVOUT = '+K',
   $END

C THE LAST SURFACE IS THE OUTER BOUNDARY OF
C THE CYLINDER GRID. NO NORMAL VECTOR IS
C SPECIFIED FOR SURFACES WHICH ARE CONNECTED
C TO OUTER BOUNDARIES THROUGH THE ISPARTOF
C USER INPUT.

  $SURFACE ISPARTOF = 'CYLINDER OUTER BOUNDARY',
   JRANGE = 40,41,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END

C END OF PEGSUS 4.0 INPUT FOR EXAMPLE 4.1.1
```

b. Page 2

Figure 29. Concluded.

```
                    DIAGNOSTIC MAPS
            LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  CARTESIAN GRID
 B  -  CYLINDER GRID


 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  CARTESIAN GRID
 b  -  CYLINDER GRID

 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT

MAP FOR CARTESIAN GRID

          PLANE  L =   1

     J         1         2         3         4         5         6
      123456789012345678901234567890123456789012345678901234567890
   K
   1 ..........................................
   2 ..........................................
   3 ..............b..*.bb.*..b................
   4 ............b..b..........b..b............
   5 ............b..............b.............
   6 ..........*.....................*.........
   7 ..........................................
   8 ........*.........BBBB...........*........
   9 ................BBBBBBB.....................
  10 .....*........BBBB####BB...........*.......
  11 ............BBB######BB...................
  12 ...bb.......BB########BBBB..........bb.....
  13 ..........BB##########BBB................
  14 .........BB##################BB..............
  15 ..bb.....BB##################BB........bb...
  16 ........BB####################BB.............
  17 ........BB####################BB.............
  18 ..*.....BB####################BB........*....
  19 .......BB######################BB...........
  20 .......BB####################BB.......b....
  21 ..*.....BB####################BB.......b*...
  22 .......BB######################BB...........
  23 ..*.....BB####################BB.......*....
  24 .......BB######################BB...........
  25 ........BB####################BB.............
  26 ..bb....BB####################BB........bb...
  27 ..........BB################BB...............
  28 .........BB##################BB..............
  29 ...bb.....BB##########BBB..........bb.....
  30 ............BB########BBBB...................
  31 .....*......BBB######BB............*.........
  32 ..............BBBB####BB.....................
  33 .......*........BBBBBBB...........*..........
  34 ..................BBBB.....................
  35 ..........*.....................*...........
  36 ............b..............b.............
  37 ............b..b..........b..b.............
  38 ..............b..*..b.*..b................
  39 ....................b.....................
  40 ..........................................
  41 ..........................................
```

Figure 30. Diagnostic map for indirect hole creation using $BOUNDARY and $SURFACE.

```
        PPPPPPPPP  EEEEEEEEE  GGGGGGGGG  SSSSSSSSS  UUU   UUU  SSSSSSSSS
       PPPPPPPPP  EEEEEEEEE  GGGGGGGGG  SSSSSSSSS  UUU   UUU  SSSSSSSSS
      PPP   PPP  EEE        GGG        SSS        UUU   UUU  SSS
     PPP   PPP  EEE        GGG        SSS        UUU   UUU  SSS
    PPPPPPPPP  EEEEEEEEE  GGG        SSSSSSSSS  UUU   UUU  SSSSSSSSS
   PPPPPPPPP  EEEEEEEEE  GGG        SSSSSSSSS  UUU   UUU  SSSSSSSSS
  PPP        EEE        GGG  GGGG        SSS  UUU   UUU        SSS
 PPP        EEE        GGG   GG         SSS  UUU   UUU        SSS
PPP        EEEEEEEEE  GGGGGGGGG  SSSSSSSSS  UUUUUUUUU  SSSSSSSSS
PPP        EEEEEEEEE  GGGGGGGGG  SSSSSSSSS  UUUUUUUUU  SSSSSSSSS

                          VERSION  4.0

                            1991


                  GLOBAL DESCRIPTION READ
                2 MESH DESCRIPTIONS READ
                3 BOUNDARY DESCRIPTIONS READ
                8 SURFACE DESCRIPTIONS READ
                0 BOX DESCRIPTIONS READ
                0 REGION DESCRIPTIONS READ
                0 VOLUME DESCRIPTIONS READ

READING MESH: CARTESIAN GRID
READING MESH: CYLINDER GRID

***** GLOBAL PARAMETER SUMMARY *****

MIN/MAX OF DOMAIN
MIN X =   0.0000E+00     MAX X =   0.4000E+02
MIN Y =   0.0000E+00     MAX Y =   0.4000E+02
MIN Z =   0.0000E+00     MAX Z =   0.4000E+01

NO INCLUDE RANGES SPECIFIED

NO. OF FRINGE POINTS = 2
QUALITY(1)           =   1.0000
QUALITY(2)           =   1.0000
QUALITY(3)           =   1.0000
EPS                  =   0.1000E-02
```

a. Page 1

Figure 31. Execution summary for indirect hole creation using $BOUNDARY and $SURFACE.

```
MESH NAME: CARTESIAN GRID

PEGSUS MESH NO.=    1
COMPOSITE MESH NO.=    1

INDEX RANGES:  JMAX =    41  KMAX =    41  LMAX =     5

TRANSLATION: X0    =  0.0000E+00  Y0    =  0.0000E+00  Z0   =  0.0000E+00
ROTATION:    XR    =  0.0000E+00  YR    =  0.0000E+00  ZR   =  0.0000E+00
             ALPHA =  0.0000E+00  BETA  =  0.0000E+00  GAM  =  0.0000E+00
SCALING:     SCALE =  0.1000E+01

NO. OF FRINGE POINTS = 2
QUALITY(1)           =  1.0000
QUALITY(2)           =  1.0000
QUALITY(3)           =  1.0000
EPS                  =  0.1000E-02

MIN/MAX OF MESH
MIN X =   0.0000E+00     MAX X =   0.4000E+02
MIN Y =   0.0000E+00     MAX Y =   0.4000E+02
MIN Z =   0.0000E+00     MAX Z =   0.4000E+01

NO INCLUDE RANGES SPECIFIED


LINKS
 NO.     NAME
  1      CYLINDER GRID


MESH NAME: CYLINDER GRID

PEGSUS MESH NO.=    2
COMPOSITE MESH NO.=    2

INDEX RANGES:  JMAX =    41  KMAX =    41  LMAX =     5

TRANSLATION: X0    =  0.2000E+02  Y0    =  0.2000E+02  Z0   =  0.0000E+00
ROTATION:    XR    =  0.0000E+00  YR    =  0.0000E+00  ZR   =  0.0000E+00
             ALPHA =  0.0000E+00  BETA  =  0.0000E+00  GAM  =  0.0000E+00
SCALING:     SCALE =  0.1000E+01

NO. OF FRINGE POINTS = 2
QUALITY(1)           =  1.0000
QUALITY(2)           =  1.0000
QUALITY(3)           =  1.0000
EPS                  =  0.1000E-02

MIN/MAX OF MESH
MIN X =   0.2000E+01     MAX X =   0.3800E+02
MIN Y =   0.2000E+01     MAX Y =   0.3800E+02
MIN Z =   0.0000E+00     MAX Z =   0.4000E+01

NO INCLUDE RANGES SPECIFIED


LINKS
 NO.     NAME
  1      CARTESIAN GRID
```

b. Page 2

FIgure 31. Continued.

```
***** BOUNDARY PARAMETER SUMMARY *****


BOUNDARY NAME: CYLINDER HOLE BOUNDARY1

BOUNDARY TYPE    = HOLE
CLOSED           =     T
IS PART OF MESH = CYLINDER GRID

MIN/MAX OF HOLE BOUNDARY
MIN X =   0.2000E+02      MAX X =   0.2825E+02
MIN Y =   0.1175E+02      MAX Y =   0.2825E+02
MIN Z =   0.0000E+00      MAX Z =   0.4000E+01

MAKE HOLE IN:
 NO.     NAME
  1      CARTESIAN GRID


BOUNDARY NAME: CYLINDER HOLE BOUNDARY2

BOUNDARY TYPE    = HOLE
CLOSED           =     T
IS PART OF MESH = CYLINDER GRID

MIN/MAX OF HOLE BOUNDARY
MIN X =   0.8500E+01      MAX X =   0.2000E+02
MIN Y =   0.8500E+01      MAX Y =   0.3150E+02
MIN Z =   0.0000E+00      MAX Z =   0.4000E+01

MAKE HOLE IN:
 NO.     NAME
  1      CARTESIAN GRID


BOUNDARY NAME: CYLINDER OUTER BOUNDARY

BOUNDARY TYPE    = OUTER
CLOSED           =     T
IS PART OF MESH = CYLINDER GRID


SURFACE NO. =     1
ISPARTOF : CYLINDER HOLE BOUNDARY1
SURFACE RANGES
MIN J =    11   MAX J =    11
MIN K =     1   MAX K =    11
MIN L =     1   MAX L =     5
NVOUT = +J
CONNECTING MESH = CYLINDER GRID


SURFACE NO. =     2
ISPARTOF : CYLINDER HOLE BOUNDARY1
SURFACE RANGES
MIN J =    11   MAX J =    11
MIN K =    31   MAX K =    41
MIN L =     1   MAX L =     5
NVOUT = +J
CONNECTING MESH = CYLINDER GRID
```

c. Page 3

Figure 31. Continued.

```
SURFACE NO. =     3
ISPARTOF : CYLINDER HOLE BOUNDARY1
SURFACE RANGES
MIN J =     1    MAX J =    11
MIN K =    11    MAX K =    11
MIN L =     1    MAX L =     5
NVOUT = +K
CONNECTING MESH = CYLINDER GRID


SURFACE NO. =     4
ISPARTOF : CYLINDER HOLE BOUNDARY1
SURFACE RANGES
MIN J =     1    MAX J =    11
MIN K =    31    MAX K =    31
MIN L =     1    MAX L =     5
NVOUT = -K
CONNECTING MESH = CYLINDER GRID


SURFACE NO. =     5
ISPARTOF : CYLINDER HOLE BOUNDARY2
SURFACE RANGES
MIN J =    21    MAX J =    21
MIN K =    11    MAX K =    31
MIN L =     1    MAX L =     5
NVOUT = +J
CONNECTING MESH = CYLINDER GRID


SURFACE NO. =     6
ISPARTOF : CYLINDER HOLE BOUNDARY2
SURFACE RANGES
MIN J =     1    MAX J =    21
MIN K =    11    MAX K =    11
MIN L =     1    MAX L =     5
NVOUT = -K
CONNECTING MESH = CYLINDER GRID


SURFACE NO. =     7
ISPARTOF : CYLINDER HOLE BOUNDARY2
SURFACE RANGES
MIN J =     1    MAX J =    21
MIN K =    31    MAX K =    31
MIN L =     1    MAX L =     5
NVOUT = +K
CONNECTING MESH = CYLINDER GRID


SURFACE NO. =     8
ISPARTOF : CYLINDER OUTER BOUNDARY
SURFACE RANGES
MIN J =    40    MAX J =    41
MIN K =     1    MAX K =    41
MIN L =     1    MAX L =     5
NVOUT =
CONNECTING MESH = CYLINDER GRID
```

d. Page 4

Figure 31. Continued.

```
INTERPOLATING BOUNDARY POINTS FOR
MESH: CARTESIAN GRID
LINK: CYLINDER GRID
QUALITY= 1.00000      STENCILS FOUND:       630

NO ORPHAN POINTS LEFT IN MESH - CARTESIAN GRID


INTERPOLATING BOUNDARY POINTS FOR
MESH: CYLINDER GRID
LINK: CARTESIAN GRID
QUALITY= 1.00000      STENCILS FOUND:       410

NO ORPHAN POINTS LEFT IN MESH - CYLINDER GRID


***** QUALITY SUMMARY *****

MINIMUM        NO. OF INTERPOLATED
QUALITY        BOUNDARY POINTS FOUND
  =1.0                  1040
 >=0.9                     0
 >=0.8                     0
 >=0.7                     0
 >=0.6                     0
 >=0.5                     0
 >=0.4                     0
 >=0.3                     0
 >=0.2                     0
 >=0.1                     0
 >=0.0                     0

MESH: CARTESIAN GRID
NUMBER OF BOUNDARY POINTS:       630
NUMBER OF INTERPOLATION STENCILS:       410

MESH: CYLINDER GRID
NUMBER OF BOUNDARY POINTS:       410
NUMBER OF INTERPOLATION STENCILS:       630
```

e. Page 5

Figure 31. Concluded.

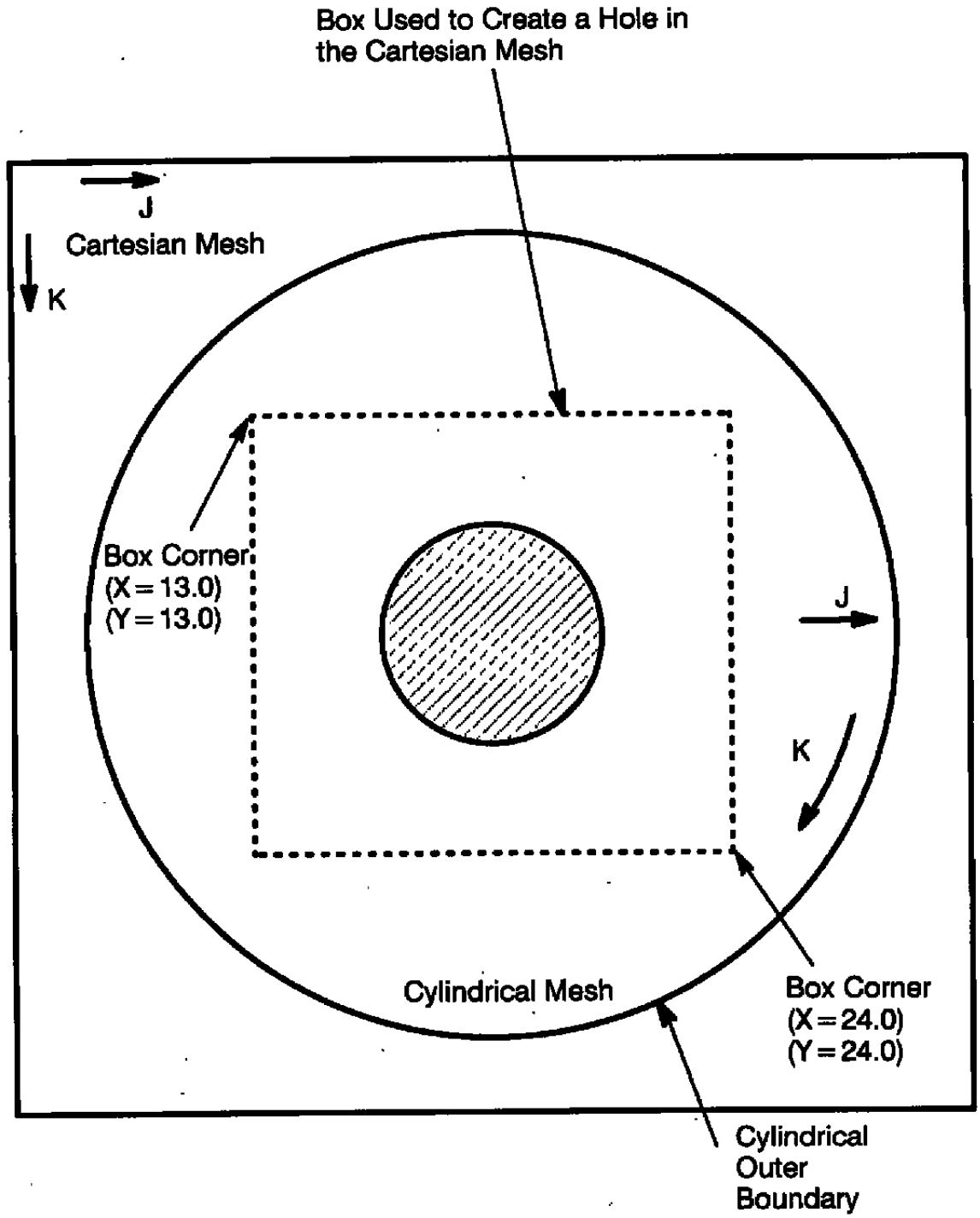


Figure 32. Graphical depiction of indirect hole creation using $BOUNDARY and $BOX.

```
C              PEGSUS 4.0 INPUT
C            CARTESIAN GRID / CYLINDER
C    EXAMPLE 4.1.2 - DIRECT HOLE SPECIFICATION

C  DEFINITION OF GLOBAL PARAMETERS
 $GLOBAL
   FRINGE = 2,
   $END

C MESH DEFINITIONS

 $MESH NAME = 'CARTESIAN GRID',
   LINK = ,
   $END

 $MESH NAME = 'CYLINDER GRID',
   LINK = 'CARTESIAN GRID',
   X0   = 20.0,
   Y0   = 20.0,
   $END

C BOUNDARY DEFINITIONS CONSIST OF ONE HOLE
C CREATION BOUNDARY AND AN OUTER BOUNDARY.

  $BOUNDARY NAME = 'CYLINDER HOLE BOUNDARY',
   ISPARTOF = 'CYLINDER GRID',
   MHOLEIN = 'CARTESIAN GRID',
   $END

  $BOUNDARY NAME = 'CYLINDER OUTER BOUNDARY',
   ISPARTOF = 'CYLINDER GRID',
   $END

C BOX DEFINITION

C ONLY ONE BOX IS DEFINED. THIS BOX IS
C PART OF THE CYLINDER HOLE BOUNDARY

  $BOX ISPARTOF = 'CYLINDER HOLE BOUNDARY'
   XRANGE = 13.0,24.0,
   YRANGE = 13.0,24.0,
   ZRANGE = 0.0,4.0,
   $END

C SURFACE DEFINITION

C THE ONLY SURFACE IS THE OUTER BOUNDARY OF
C THE CYLINDER GRID. NO NORMAL VECTOR IS
C SPECIFIED FOR SURFACES WHICH ARE CONNECTED
C TO OUTER BOUNDARIES THROUGH THE ISPARTOF
C USER INPUT.

  $SURFACE ISPARTOF = 'CYLINDER OUTER BOUNDARY',
   JRANGE = 40,41,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END

C END OF PEGSUS 4.0 INPUT FOR EXAMPLE 4.1.2
```

Figure 33. Input for indirect hole creation using $BOUNDARY and $BOX.

```
                    DIAGNOSTIC MAPS
          LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  CARTESIAN GRID
 B  -  CYLINDER GRID

 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  CARTESIAN GRID
 b  -  CYLINDER GRID

 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT


MAP FOR CARTESIAN GRID

          PLANE  L =   1

     J         1         2         3         4         5         6
      12345678901234567890123456789012345678901234567890123456789O
   K
   1 ..........................................
   2 ..........................................
   3 ..............b..*.bb.*..b...............
   4 ...........b..b..........b..b.............
   5 ...........b..................b...........
   6 .........*......................*.........
   7 ..........................................
   8 .......*............................*.......
   9 ..........................................
  10 .....*................................*.....
  11 ..........................................
  12 ...bb........BBBBBBBBBBBB..........bb.....
  13 ............BBBBBBBBBBBBBB..............
  14 ...........BB##############BB.............
  15 ..bb.......BB##############BB.........bb...
  16 ...........BB##############BB.............
  17 ...........BB##############BB.............
  18 ..*........BB##############BB..........*...
  19 ...........BB##############BB.............
  20 ...........BB##############BB..........b...
  21 ..*........BB##############BB..........b*..
  22 ...........BB##############BB.............
  23 ..*........BB##############BB..........*...
  24 ...........BB##############BB.............
  25 ...........BB##############BB.............
  26 ..bb........BBBBBBBBBBBBBB..........bb...
  27 ..............BBBBBBBBBBBB................
  28 ..........................................
  29 ...bb..................................bb.....
  30 ..........................................
  31 .....*................................*.....
  32 ..........................................
  33 .......*............................*.......
  34 ..........................................
  35 .........*......................*.........
  36 ...........b..................b...........
  37 ...........b..b..........b..b.............
  38 ..............b..*..b.*..b................
  39 ....................b.....................
  40 ..........................................
  41 ..........................................
```

Figure 34. Diagnostic map for indirect hole creation using $BOUNDARY and $BOX.

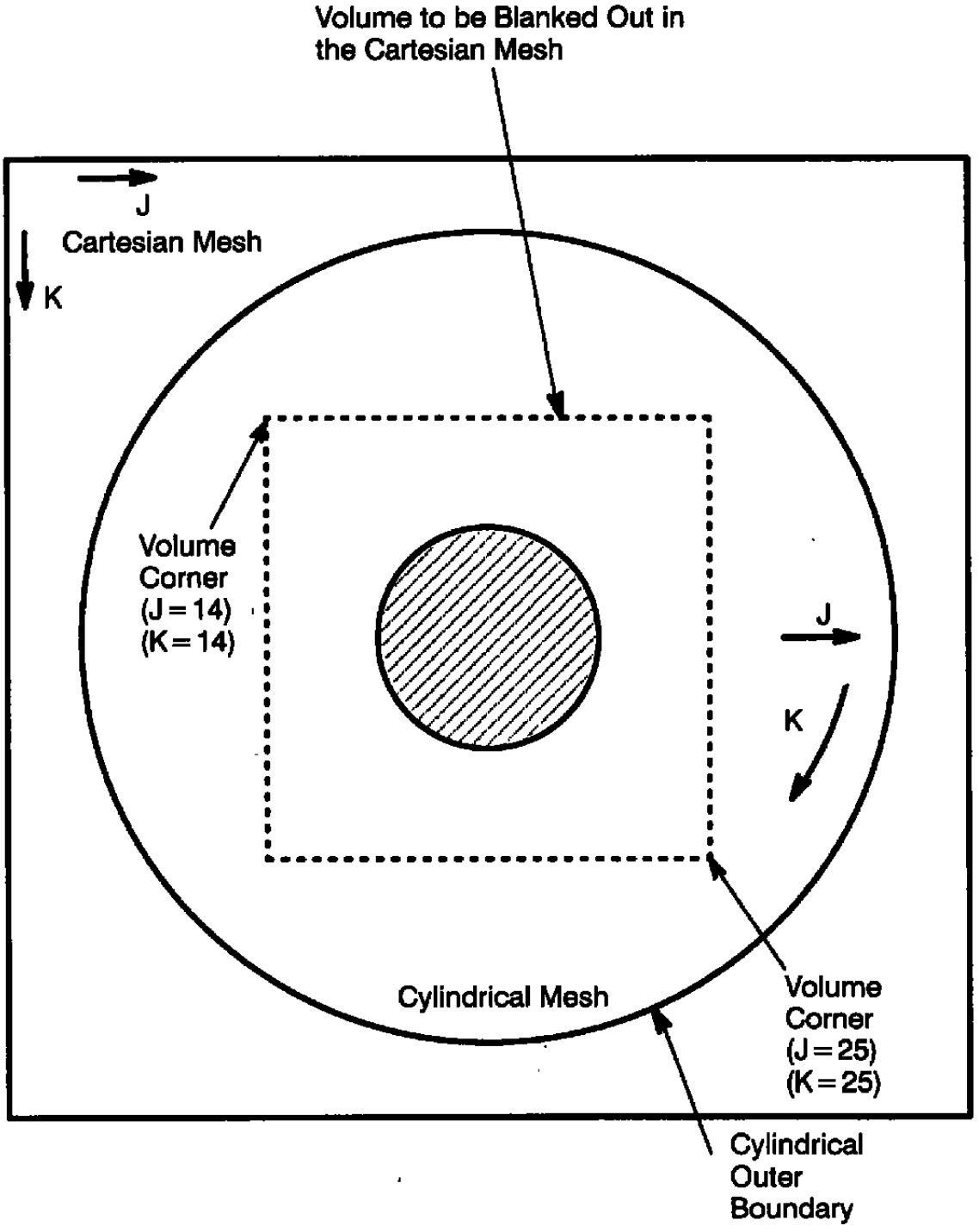


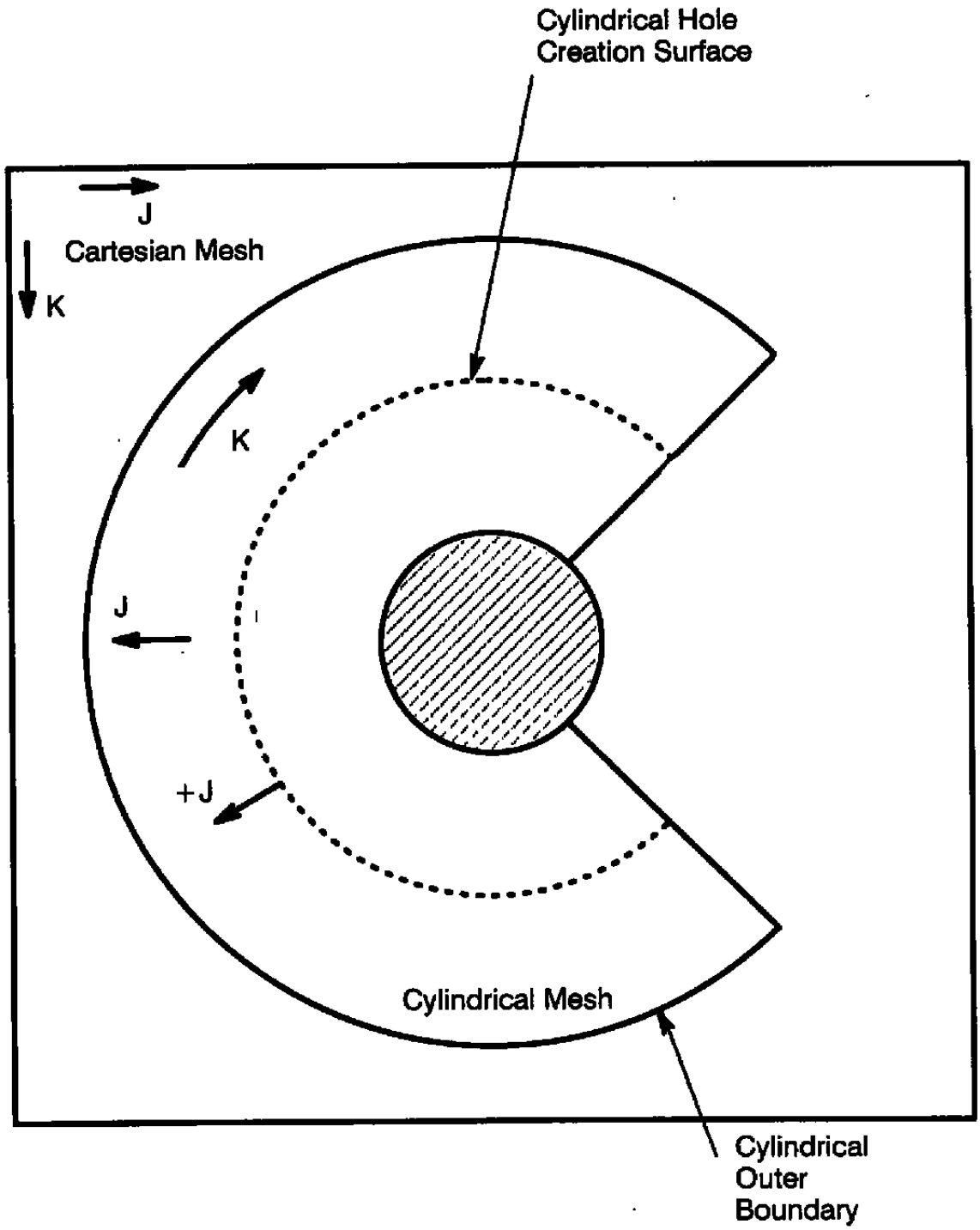
Figure 35. Graphical depiction of direct hole creation using $REGION and $VOLUME.

```
C              PEGSUS 4.0 INPUT
C            CARTESIAN GRID / CYLINDER
C    EXAMPLE 4.1.3 - DIRECT HOLE SPECIFICATION

C  DEFINITION OF GLOBAL PARAMETERS
 $GLOBAL
   FRINGE = 2,
   $END

C MESH DEFINITIONS

 $MESH NAME = 'CARTESIAN GRID',
   LINK = ,
   $END

 $MESH NAME = 'CYLINDER GRID',
   LINK = 'CARTESIAN GRID',
   X0   = 20.0,
   Y0   = 20.0,
   $END

C BOUNDARY DEFINITION CONSISTS OF ONE OUTER BOUNDARY.

  $BOUNDARY NAME = 'CYLINDER OUTER BOUNDARY',
   ISPARTOF = 'CYLINDER GRID',
   $END

C REGION DEFINITION

C ONLY ONE REGION IS DEFINED. THIS REGION
C WILL MAKE A HOLE IN THE CARTESIAN GRID.

  $REGION NAME = 'CARTESIAN HOLE BOUNDARY',
   ISPARTOF = 'CARTESIAN GRID',
   $END


C SURFACE DEFINITION

C THE ONLY SURFACE IS THE OUTER BOUNDARY OF
C THE CYLINDER GRID. NO NORMAL VECTOR IS
C SPECIFIED FOR SURFACES WHICH ARE CONNECTED
C TO OUTER BOUNDARIES THROUGH THE ISPARTOF
C USER INPUT.

  $SURFACE ISPARTOF = 'CYLINDER OUTER BOUNDARY',
   JRANGE = 40,41,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END

C VOLUME DEFINITION

C ONLY ONE VOLUME IS DEFINED. THE VOLUME
C CONSISTS OF THE J , K AND L RANGES OF
C THE POINTS IN THE CARTESIAN GRID WHICH
C ARE TO BE BLANKED OUT.

  $VOLUME ISPARTOF = 'CARTESIAN HOLE BOUNDARY'
   JRANGE = 14,25,
   KRANGE = 14,25,
   LRANGE = 1,5,
   $END


C END OF PEGSUS 4.0 INPUT FOR EXAMPLE 4.1.3
```

Figure 36. Input for direct hole creation using $REGION and $VOLUME.

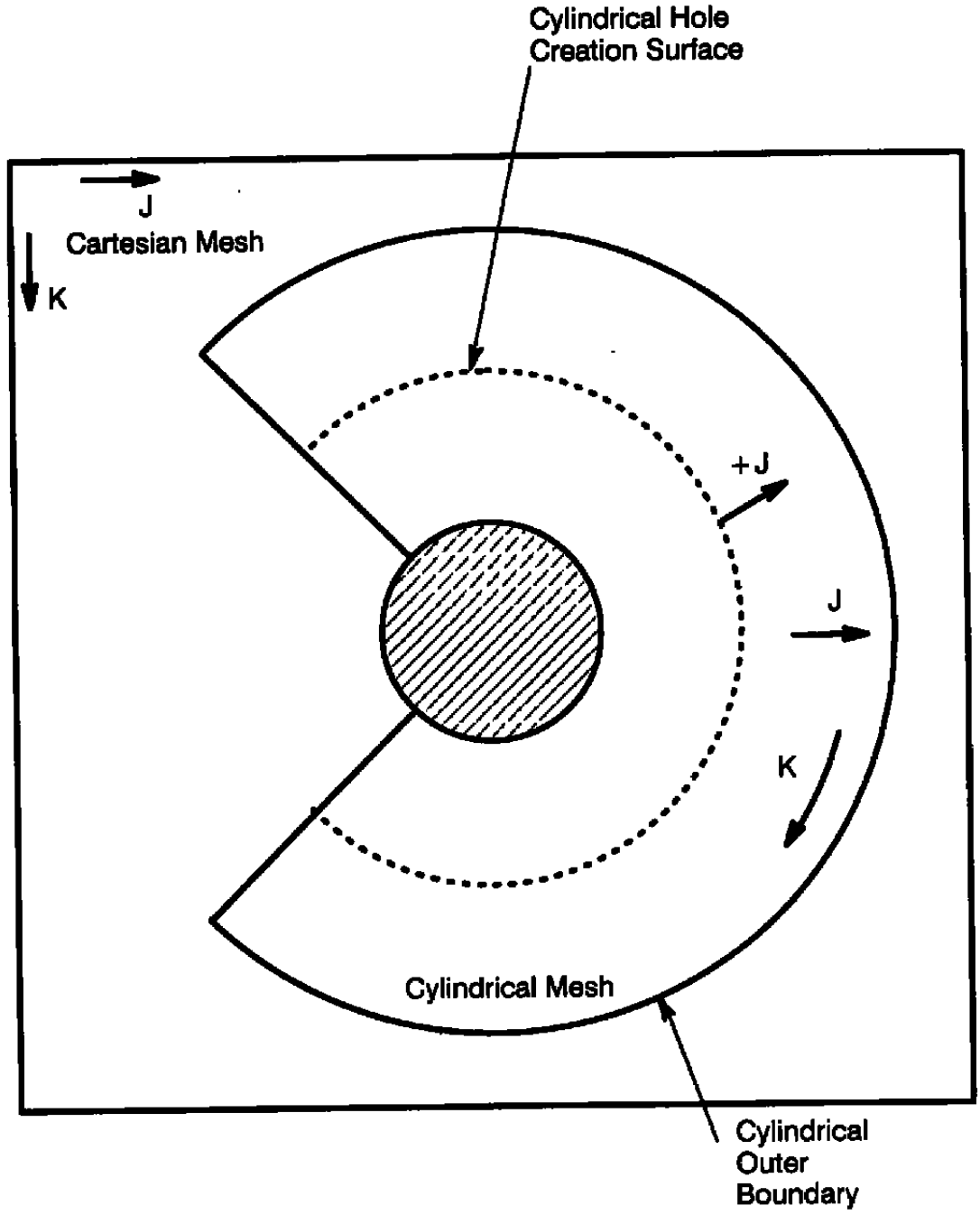
```
                    DIAGNOSTIC MAPS
            LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  CARTESIAN GRID
 B  -  CYLINDER GRID

 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  CARTESIAN GRID
 b  -  CYLINDER GRID

 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT


MAP FOR CARTESIAN GRID

           PLANE  L =   1

     J         1         2         3         4         5         6
       12345678901234567890123456789012345678901234567890123456789
   K
   1 .........................................
   2 .........................................
   3 ..............b..*.bb.*..b..............
   4 ...........b..b..........b..b............
   5 ...........b..............b.............
   6 .........*.....................*.........
   7 .........................................
   8 .......*.........................*.......
   9 .........................................
  10 .....*...............................*.....
  11 .........................................
  12 ...bb........BBBBBBBBBBBB..........bb....
  13 ............BBBBBBBBBBBBBB.............
  14 ...........BB##############BB............
  15 ..bb.......BB##############BB.........bb...
  16 ...........BB##############BB............
  17 ...........BB##############BB............
  18 ..*........BB##############BB..........*...
  19 ...........BB##############BB............
  20 ...........BB##############BB..........b...
  21 ..*........BB##############BB.........b*..
  22 ...........BB##############BB............
  23 ..*........BB##############BB..........*...
  24 ...........BB##############BB............
  25 ...........BB##############BB............
  26 ..bb........BBBBBBBBBBBBBB..........bb...
  27 ............BBBBBBBBBBBB...............
  28 .........................................
  29 ...bb.................................bb....
  30 .........................................
  31 .....*...............................*.....
  32 .........................................
  33 .......*.........................*.........
  34 .........................................
  35 .........*.....................*...........
  36 ...........b..............b...............
  37 ...........b..b..........b..b.............
  38 .............b..*..b.*..b.................
  39 ...................b.....................
  40 .........................................
  41 .........................................
```

Figure 37. Diagnostic map for direct hole creation using $REGION and $VOLUME.



a. 'CYLINDER GRID 1' and 'CARTESIAN GRID'

Figure 38. Graphical depiction of indirect hole creation using surfaces from two separate meshes.



b. 'CYLINDER GRID 2' and 'CARTESIAN GRID'
Figure 38. Concluded.

```
C              PEGSUS 4.0 INPUT
C            CARTESIAN GRID / CYLINDER
C    EXAMPLE 4.1.4 - HOLE CREATION BOUNDARY COMPRISED
C                    OF SURFACES FROM TWO MESHS

C  DEFINITION OF GLOBAL PARAMETERS
 $GLOBAL
   FRINGE = 2,
   $END


C MESH DEFINITIONS

 $MESH NAME = 'CARTESIAN GRID',
   LINK = ,
   $END

 $MESH NAME = 'CYLINDER GRID 1',
   LINK = 'CYLINDER GRID 2', 'CARTESIAN GRID',
   X0   = 20.0,
   Y0   = 20.0,
   $END

 $MESH NAME = 'CYLINDER GRID 2',
   LINK = 'CYLINDER GRID 1', 'CARTESIAN GRID',
   X0   = 20.0,
   Y0   = 20.0,
   $END


C BOUNDARY DEFINITIONS CONSIST OF ONE HOLE
C CREATION BOUNDARY AND TWO OUTER BOUNDARIES.

  $BOUNDARY NAME = 'CYLINDER HOLE BOUNDARY',
   ISPARTOF = 'CYLINDER GRID 1',
   MHOLEIN = 'CARTESIAN GRID',
   $END


  $BOUNDARY NAME = 'CYLINDER 1 OUTER BOUNDARY',
   ISPARTOF = 'CYLINDER GRID 1',
   $END

  $BOUNDARY NAME = 'CYLINDER 2 OUTER BOUNDARY',
   ISPARTOF = 'CYLINDER GRID 2',
   $END
```

a. Page 1

Figure 39. Input for hole creation using surfaces from two separate meshes.

```
C SURFACE DEFINITIONS

C TWO SURFACES ARE NEEDED TO DESCRIBE
C THE CYLINDER HOLE BOUNDARY. ONE SURFACE
C RESPECTIVELY FROM EACH OF THE CYLINDER
C MESHS.


  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY',
   JRANGE = 21,21,
   KRANGE = 1,41,
   LRANGE = 1,5,
   NVOUT = '+J',
   $END

  $SURFACE ISPARTOF = 'CYLINDER HOLE BOUNDARY',
   MESH NAME = 'CYLINDER GRID 2',
   JRANGE = 21,21,
   KRANGE = 1,41,
   LRANGE = 1,5,
   NVOUT = '+J',
   $END


C THE LAST SIX SURFACES ARE THE OUTER BOUNDARIES
C OF THE CYLINDER GRIDS. NO NORMAL VECTORS ARE
C SPECIFIED FOR SURFACES WHICH ARE CONNECTED TO
C OUTER BOUNDARIES THROUGH THE ISPARTOF USER INPUT.

  $SURFACE ISPARTOF = 'CYLINDER 1 OUTER BOUNDARY',
   JRANGE = 40,41,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END
  $SURFACE ISPARTOF = 'CYLINDER 1 OUTER BOUNDARY',
   JRANGE = 1,41,
   KRANGE = 1,2,
   LRANGE = 1,5,
   $END
  $SURFACE ISPARTOF = 'CYLINDER 1 OUTER BOUNDARY',
   JRANGE = 1,41,
   KRANGE = 40,41,
   LRANGE = 1,5,
   $END

  $SURFACE ISPARTOF = 'CYLINDER 2 OUTER BOUNDARY',
   JRANGE = 40,41,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END
  $SURFACE ISPARTOF = 'CYLINDER 2 OUTER BOUNDARY',
   JRANGE = 1,41,
   KRANGE = 1,2,
   LRANGE = 1,5,
   $END
  $SURFACE ISPARTOF = 'CYLINDER 2 OUTER BOUNDARY',
   JRANGE = 1,41,
   KRANGE = 40,41,
   LRANGE = 1,5,
   $END

C END OF PEGSUS 4.0 INPUT FOR EXAMPLE 4.1.4
```

b. Page 2
Figure 39. Concluded.

```
                    DIAGNOSTIC MAPS
            LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  CARTESIAN GRID
 B  -  CYLINDER GRID 1
 C  -  CYLINDER GRID 2

 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  CARTESIAN GRID
 b  -  CYLINDER GRID 1
 c  -  CYLINDER GRID 2

 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT

MAP FOR CARTESIAN GRID

          PLANE  L =   1

     J         1         2         3         4         5         6
      1234567890123456789012345678901234567890123456789012345678901234567890
   K
   1 ........................................
   2 ........................................
   3 ..............c**********b..............
   4 ............**c..........b**............
   5 ..........**..................**..........
   6 ........c*......................*b........
   7 .......b.*......................*.c.......
   8 .......*b........BBBBBBB.......c*........
   9 ......*........BBBBBBBBBBB.......*.......
  10 ..............BBBB#######BBBC.............
  11 .....*......BBB###########BCC.....*......
  12 ...bb......BB#################CC.....cc....
  13 ..........BB###################CC..........
  14 ...*.....BB###################CC....*....
  15 .........BB###################CC.........
  16 ..*.....BB#####################CC....*...
  17 ........BB#####################CC........
  18 ..*....BB#######################CC...*...
  19 .......BB#######################CC.......
  20 .......BB#######################CC...c...
  21 ..*....BB#######################CC....c..
  22 .......BB#######################CC.......
  23 ..*....BB#######################CC...*...
  24 .......BB#######################CC.......
  25 ..*.....BB#####################CC....*...
  26 ........BB#####################CC........
  27 ...*.....BB###################CC....*....
  28 .........BB###################CC.........
  29 ...bb.....BB#################CC....cc....
  30 .....*.....BB###############CC....*......
  31 ............BBB###########BCC............
  32 ......*......BBBB#######BBBC.....*.......
  33 .......*b......BBBBBBBBBBB.....c*........
  34 .......b.*.......BBBBBBB......*.c........
  35 ........c*......................*b........
  36 ..........**..................**..........
  37 ............**c..........b**..............
  38 ..............c**********b................
  39 ........................................
  40 ........................................
```

a. Page 1

Figure 40. Diagnostic maps for hole creation using surfaces from two separate meshes.

```
MAP FOR CYLINDER GRID 1

          PLANE  L =   1

     J         1         2         3         4         5         6
      12345678901234567890123456789012345678901234567890123456789
   K
   1 ccccccccccccccccccccccccccccccccccccccccccCAAA
   2 ccccccccccccccccccccccccccccccccccccccccccCAAA
   3 ....................a..a..............AA
   4 ....................aa.aa.............AA
   5 ....................a..a..............AA
   6 ....................aa..a.............AA
   7 ....................a..*..............AA
   8 ....................a..aa.............AA
   9 ....................*..a..............AA
  10 ....................a..a..............AA
  11 ....................a.aa..............AA
  12 ....................a.a.a.............AA
  13 ccccccccccccccccccc*cc*ccccccccccccc.AA
  14 ccccccccccccccccccc*c*ccccccccccccc.AA
  15 ....................a.a..............AA
  16 ....................a..aa.............AA
  17 ....................a..a..............AA
  18 ....................a.aa..............AA
  19 ....................a..*..............AA
  20 ....................*..a..............AA
  21 ....................a..*..............AA
  22 ....................a..*..............AA
  23 ....................a.aa..............AA
  24 ....................a..a..............AA
  25 ....................a..aa.............AA
  26 ....................a.a...............AA
  27 ccccccccccccccccccc*c*ccccccccccccc.AA
  28 ccccccccccccccccccc*cc*ccccccccccccc.AA
  29 ....................a.a.a.............AA
  30 ....................a.aa..............AA
  31 ....................a..a..............AA
  32 ....................*..a..............AA
  33 ....................a..aa.............AA
  34 ....................a..*..............AA
  35 ....................aa..a.............AA
  36 ....................a..a..............AA
  37 ....................aa.aa.............AA
  38 ....................a..a..............AA
  39 ..........................................AA
  40 ccccccccccccccccccccccccccccccccccccccccCAAA
  41 ccccccccccccccccccccccccccccccccccccccccCAAA
```

b. Page 2
Figure 40. Continued.

```
MAP FOR CYLINDER GRID 2

          PLANE  L =   1

     J         1         2         3         4         5         6
      12345678901234567890123456789012345678901234567890123456789
   K
   1 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAAA
   2 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAAA
   3 .....................................................AA
   4 .....................................................AA
   5 .....................................................AA
   6 .....................................................AA
   7 .....................................................AA
   8 .....................................................AA
   9 .....................................................AA
  10 .....................................................AA
  11 .....................................................AA
  12 ......................a.a............................AA
  13 bbbbbbbbbbbbbbbbbbbb*bb*bbbbbbbbbbbbb.AA
  14 bbbbbbbbbbbbbbbbbbbb*b*bbbbbbbbbbbbb.AA
  15 ......................a.a............................AA
  16 ......................a..aa..........................AA
  17 ......................a..a...........................AA
  18 .....................a.aa............................AA
  19 .....................a..*...........................AA
  20 .....................*..*...........................AA
  21 .....................a..a...........................AA
  22 .....................a..*...........................AA
  23 .....................a.aa............................AA
  24 .....................a..a...........................AA
  25 .....................a..aa..........................AA
  26 .....................a.a............................AA
  27 bbbbbbbbbbbbbbbbbbbb*b*bbbbbbbbbbbbb.AA
  28 bbbbbbbbbbbbbbbbbbbb*bb*bbbbbbbbbbbbb.AA
  29 .....................a.a............................AA
  30 .....................................................AA
  31 .....................................................AA
  32 .....................................................AA
  33 .....................................................AA
  34 .....................................................AA
  35 .....................................................AA
  36 .....................................................AA
  37 .....................................................AA
  38 .....................................................AA
  39 .....................................................AA
  40 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAAA
  41 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBAAA
```

c. Page 3

Figure 40. Concluded.

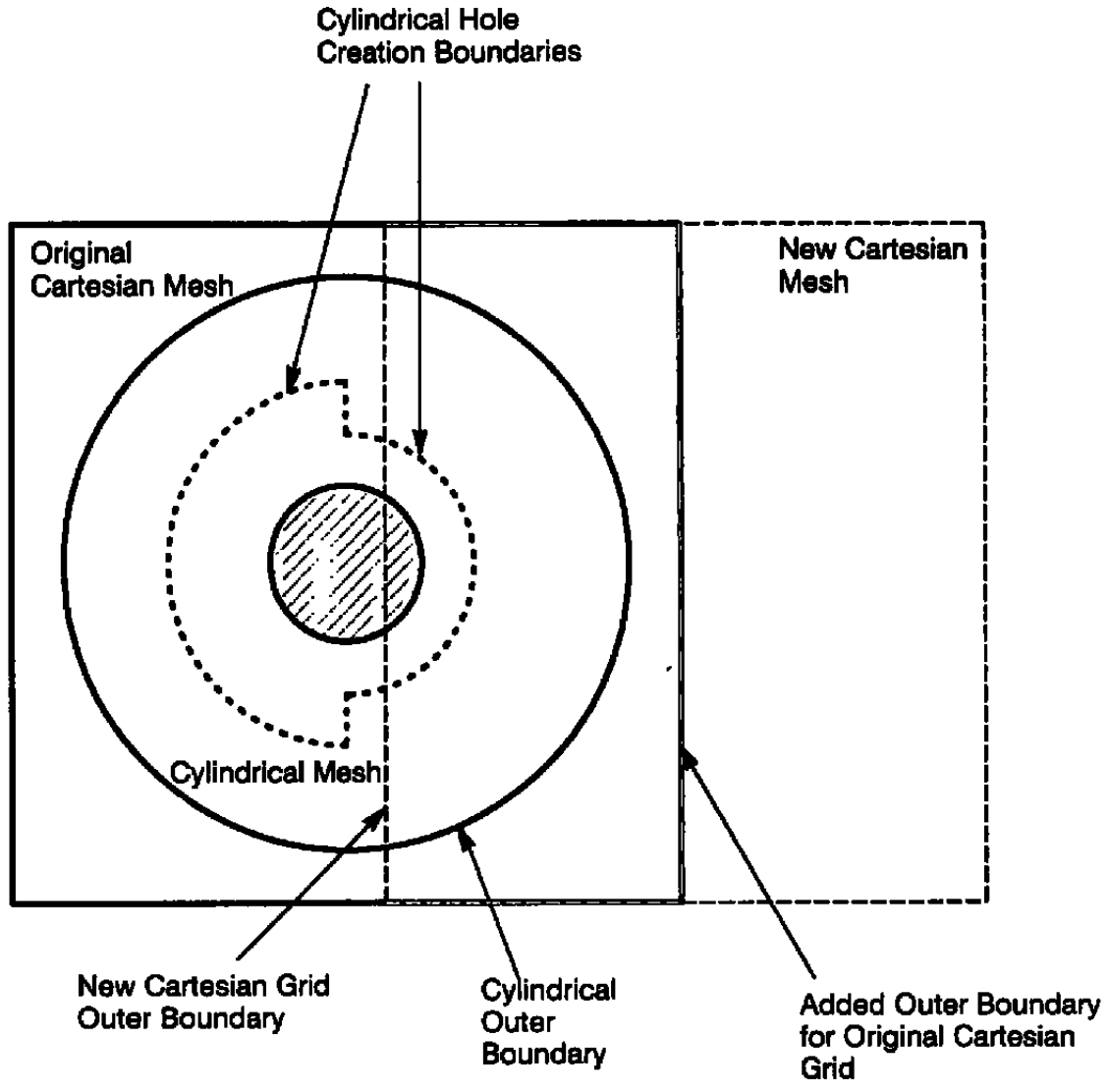


Figure 41. Addition of a mesh and boundary.

```
C              PEGSUS 4.0 INPUT
C            CARTESIAN GRID / CYLINDER
C     EXAMPLE 4.2.1 - ADDITION


C MESH DEFINITIONS
C THE NEW CARTESIAN MESH IS THE ONLY MESH HERE.
C THIS MESH WILL BE ADDED TO THE CONFIGURATION
C OF EXAMPLE 4.1.1 .
C THE 'ADDLINK' ENTRY WILL ADD THE NEW MESH TO
C THE LINK LISTS OF THE OTHER GRIDS.
C THE 'ADDHOLE' ENTRY WILL ADD THE NEW MESH TO
C THE MAKE HOLE IN LISTS OF THE HOLE CREATION
C BOUNDARY.

 $MESH NAME = 'NEW CARTESIAN GRID',
   MODE = 'ADD',
   LINK = ,
   ADDLINK = 'CARTESIAN GRID','CYLINDER GRID',
   ADDHOLE = 'CYLINDER HOLE BOUNDARY1','CYLINDER HOLE BOUNDARY2',
   X0   = 25.0,
   $END


C BOUNDARY DEFINITIONS CONSIST OF TWO OUTER BOUNDARIES.
C AN OUTER BOUNDARY WILL BE ADDED TO THE ORIGINAL CARTESIAN
C MESH AND AND OUTER BOUNDARY WILL BE DEFINED IN THE
C NEW CARTESIAN MESH.


  $BOUNDARY NAME = 'CARTESIAN OUTER BOUNDARY',
   ISPARTOF = 'CARTESIAN GRID',
   MODE = 'ADD',
   $END

  $BOUNDARY NAME = 'NEW CARTESIAN OUTER BOUNDARY',
   ISPARTOF = 'NEW CARTESIAN GRID',
   $END

C SURFACE DEFINITIONS

C THE LAST TWO SURFACES ARE THE OUTER BOUNDARIES
C OF THE CARTESIAN GRIDS. NO NORMAL VECTORS ARE
C SPECIFIED FOR SURFACES WHICH ARE CONNECTED TO
C OUTER BOUNDARIES THROUGH THE ISPARTOF USER INPUT.

  $SURFACE ISPARTOF = 'CARTESIAN OUTER BOUNDARY',
   JRANGE = 40,41,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END

  $SURFACE ISPARTOF = 'NEW CARTESIAN OUTER BOUNDARY',
   JRANGE = 1,2,
   KRANGE = 1,41,
   LRANGE = 1,5,
   $END

C END OF PEGSUS 4.0 INPUT FOR EXAMPLE 4.2.1
```

Figure 42. Input for addition of mesh and boundary.

```
                    DIAGNOSTIC MAPS
            LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  CARTESIAN GRID
 B  -  CYLINDER GRID
 C  -  NEW CARTESIAN GRID

 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  CARTESIAN GRID
 b  -  CYLINDER GRID
 c  -  NEW CARTESIAN GRID

 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT

MAP FOR CARTESIAN GRID

          PLANE  L =   1

     J         1         2         3         4         5         6
      1234567890123456789012345678901234567890123456789012345678901234567890
   K
   1 ..........................cc............CC
   2 ..........................cc............CC
   3 ..............b..*.bb.*..*c............CC
   4 ............b..b..........*c.b..........CC
   5 ............b.............cc.b..........CC
   6 .........*................cc...*.........CC
   7 ..........................cc............CC
   8 ........*.........BBBB....cc.....*.......CC
   9 ................BBBBBBB...cc............CC
  10 .....*.......BBBB####BB..cc.......*....CC
  11 .............BBB######BB..cc.............CC
  12 ...bb.......BB########BBBBcc.........bb..CC
  13 ...........BB##########BBBc.............CC
  14 ..........BB################BB............CC
  15 ..bb.....BB##################BB........bb.CC
  16 ........BB####################BB..........CC
  17 ........BB####################BB..........CC
  18 ..*....BB####################BB.......*.CC
  19 .......BB####################BB.........CC
  20 .......BB######################BB......b.CC
  21 ..*....BB######################BB......b*CC
  22 .......BB####################BB.........CC
  23 ..*...:BB####################BB......*.CC
  24 .......BB####################BB.........CC
  25 ........BB##################BB..........CC
  26 ..bb....BB##################BB.......bb.CC
  27 ..........BB################BB...........CC
  28 ..........BB################BB...........CC
  29 ...bb.....BB##############BBBc.........bb..CC
  30 ............BB########BBBBcc.............CC
  31 .....*......BBB######BB..cc.......*....CC
  32 .............BBBB####BB..cc.............CC
  33 .......*.......BBBBBBB...cc.....*.......CC
  34 ..................BBBB....cc.............CC
  35 .........*................cc...*.........CC
  36 ............b.............cc.b..........CC
  37 ............b..b..........*c.b..........CC
  38 ..............b..*..b.*..*c.............CC
  39 ......................b....cc.............CC
  40 ..........................*c.............CC
```

a. Page 1

Figure 43. Diagnostic maps for addition of a mesh and boundary.

```
MAP FOR CYLINDER GRID

        PLANE  L =   1

   J         1         2         3         4         5         6
    12345678901234567890123456789012345678901234567890123456789
 K
 1 ...........*..*.....................AA
 2 ...........*..*.....................AA
 3 ..........*.**.*....................AA
 4 ..........*.*.......................AA
 5 ..........*..*......................AA
 6 ...........*........................AA
 7 ...........*.*......................AA
 8 .........a.aa.a.....................AA
 9 ............a..a..a..................AA
10 ............a..a..a..a...............AA
11 ......................*..*a..........AA
12 ......................*..a..........AA
13 .....................aa.aa...........AA
14 .....................a.aaa...........AA
15 .....................a.aa............AA
16 .....................a..a............AA
17 .....................a.aaa...........AA
18 .....................aa.aa...........AA
19 ......................*..a..........AA
20 ......................*..*a..........AA
21 .....................a..aa...........AA
22 ......................*..a..........AA
23 .....................aa.aa...........AA
24 .....................a.aaa...........AA
25 .....................a.aa............AA
26 .....................a..a............AA
27 .....................a.aaa...........AA
28 .....................aa.aa...........AA
29 ......................*..a..........AA
30 ......................*..*a..........AA
31 ............a..a..a..a...............AA
32 ..............a..a..a................AA
33 .........a.aa.a.....................AA
34 ..........*.*.......................AA
35 .........c..*.......................AA
36 .........a..*.......................AA
37 ...........*.*......................AA
38 ..........*.**.*....................AA
39 ............*..*.....................AA
40 ............*..*.....................AA
41 ....................................AA
```

b. Page 2
Figure 43. Continued.

```
MAP FOR NEW CARTESIAN GRID

          PLANE  L =   1

     J         1         2         3         4         5         6
      12345678901234567890123456789012345678901234567890123456789
   K
    1 AA.........aa.......................
    2 AA.........aa.......................
    3 AA.........aa.......................
    4 AA.........aa.......................
    5 AA.........aa.......................
    6 AA.........aa.......................
    7 AA.........aa.......................
    8 AA.........aa.......................
    9 AA.........aa.......................
   10 AA.........aa.......................
   11 AA.........aa.......................
   12 AA.........aa.......................
   13 BA.........aa.......................
   14 BB.........aa.......................
   15 #BB........aa.......................
   16 ##BB.......aa.......................
   17 ###BB......aa.......................
   18 ###BB......aa.......................
   19 ####BB.....aa.......................
   20 ####BB.....aa.......................
   21 ####BB.....aa.......................
   22 ####BB.....aa.......................
   23 ####BB.....aa.......................
   24 ###BB......aa.......................
   25 ###BB......aa.......................
   26 ##BB.......aa.......................
   27 #BB........aa.......................
   28 BB.........aa.......................
   29 BA.........aa.......................
   30 AA.........aa.......................
   31 AA.........aa.......................
   32 AA.........aa.......................
   33 AA.........aa.......................
   34 AA.........aa.......................
   35 AA.........aa.......................
   36 AA.........aa.......................
   37 AA.........aa.......................
   38 AA.........aa.......................
   39 AA.........aa.......................
   40 AA.........*a.......................
   41 AA.....................................
```

c. Page 3

Figure 43. Concluded.

```
C                PEGSUS 4.0 INPUT
C              CARTESIAN GRID / CYLINDER
C     EXAMPLE 4.2.2 - SUBTRACTION


C MESH DEFINITIONS
C THE CYLINDER MESH WILL BE SUBTRACTED
C FROM THE MULTIPLE-MESH CONFIGURATION
C USED IN EXAMPLE 4.2.1 .


 $MESH NAME = 'CYLINDER GRID',
   MODE = 'SUB'
   $END


C END OF PEGSUS 4.0 INPUT FOR EXAMPLE 4.2.2
```

Figure 44. Input for subtraction of a mesh and boundary.

```
                    DIAGNOSTIC MAPS
          LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  CARTESIAN GRID
 B  -  NEW CARTESIAN GRID

 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  CARTESIAN GRID
 b  -  NEW CARTESIAN GRID

 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT

MAP FOR CARTESIAN GRID

          PLANE  L =   1

     J         1         2         3         4         5         6
      12345678901234567890123456789012345678901234567890123456789
   K
   1 ........................bb...........BB
   2 ........................bb...........BB
   3 ........................bb...........BB
   4 ........................bb...........BB
   5 ........................bb...........BB
   6 ........................bb...........BB
   7 ........................bb...........BB
   8 ........................bb...........BB
   9 ........................bb...........BB
  10 ........................bb...........BB
  11 ........................bb...........BB
  12 ........................bb...........BB
  13 ........................bb...........BB
  14 ........................bb...........BB
  15 ........................bb...........BB
  16 ........................bb...........BB
  17 ........................bb...........BB
  18 ........................bb...........BB
  19 ........................bb...........BB
  20 ........................bb...........BB
  21 ........................bb...........BB
  22 ........................bb...........BB
  23 ........................bb...........BB
  24 ........................bb...........BB
  25 ........................bb...........BB
  26 ........................bb...........BB
  27 ........................bb...........BB
  28 ........................bb...........BB
  29 ........................bb...........BB
  30 ........................bb...........BB
  31 ........................bb...........BB
  32 ........................bb...........BB
  33 ........................bb...........BB
  34 ........................bb...........BB
  35 ........................bb...........BB
  36 ........................bb...........BB
  37 ........................bb...........BB
  38 ........................bb...........BB
  39 ........................bb...........BB
  40 ........................*b...........BB
  41 ..................................... BB
```

a. Page 1

Figure 45. Diagnostic maps for subtraction of a mesh and boundary.

```
MAP FOR NEW CARTESIAN GRID

          PLANE  L =   1

     J.         1         2         3         4         5         6
      12345678901234567890123456789012345678901234567890123456789012345678901234567890 .
   K
    1 AA..........aa.....................
    2 AA..........aa.....................
    3 AA..........aa.....................
    4 AA..........aa.....................
    5 AA..........aa.....................
    6 AA..........aa.....................
    7 AA..........aa.....................
    8 AA..........aa.....................
    9 AA..........aa.....................
   10 AA..........aa.....................
   11 AA..........aa.....................
   12 AA..........aa.....................
   13 AA..........aa.....................
   14 AA..........aa.....................
   15 AA..........aa.....................
   16 AA..........aa.....................
   17 AA..........aa.....................
   18 AA..........aa.....................
   19 AA..........aa.....................
   20 AA..........aa.....................
   21 AA..........aa.....................
   22 AA..........aa.....................
   23 AA..........aa.....................
   24 AA..........aa.....................
   25 AA..........aa.....................
   26 AA..........aa.....................
   27 AA..........aa.....................
   28 AA..........aa.....................
   29 AA..........aa.....................
   30 AA..........aa.....................
   31 AA..........aa.....................
   32 AA..........aa.....................
   33 AA..........aa.....................
   34 AA..........aa.....................
   35 AA..........aa.....................
   36 AA..........aa.....................
   37 AA..........aa.....................
   38 AA..........aa.....................
   39 AA..........aa.....................
   40 AA..........*a.....................
   41 AA.....................................
```

b. Page 2
Figure 45. Concluded.

APPENDIX A
GLOSSARY OF TERMS

Blanked Point	A mesh point completely out of the computational domain (hole point), updated by interpolation (interpolation boundary point), or updated by boundary conditions (interior blanked point).
Boundary	Collection of one or more surfaces and/or one or more boxes.
Box	Group of points defined by X, Y, and Z coordinate ranges. Collections of one or more boxes comprise a boundary.
Candidate Point	Boundary point for which an interpolation stencil in another mesh is to be found.
Coincident Points	Points in different meshes which are closer than a predefined tolerance.
Composite Mesh	Single file containing all meshes in a multiple-mesh configuration.
Computational Domain	Mesh defined in the coordinate system of the rectangular box into which the physical domain has been mapped.
Donor Mesh	Mesh that sends information by interpolation to another mesh.
Field Point	Point within computational domain (IBLANK = 1) that is updated by the flow solver or boundary conditions.
Fringe Point	A hole boundary point, q.v.
Hole	Collection of hole points, i.e., points in a mesh that have been blanked.
Hole Boundary	Collection of hole boundary points (fringe points) surrounding a hole.
Hole Boundary Point	Point on a hole boundary, by definition a point between a hole point and field point.

Hole Creation Boundary	Collection of surfaces in one or more meshes that creates a hole in another mesh.
Hole Point	A mesh point that has been specified to be within a hole. This point is considered to be out of the computational domain (IBLANK = 0).
Interpolation Boundary Point	A hole boundary point or outer boundary point that is interpolated from another mesh (IBLANK = 0).
Interpolation File	File consisting of (1) boundary points, (2) pointers to interpolation stencils, and (3) interpolation coefficients for each mesh. The interpolation file is output by PEGSUS and is input to the flow solver.
Interpolation Stencil	The eight mesh points that surround a candidate boundary point.
Level Surface	A constant J, K, or L plane of a mesh.
Orphan Point	Boundary point in which a valid interpolation can not be found in any LINKed mesh.
Outer Boundary	Collection of outer boundary points.
Outer Boundary Point	A point in a mesh that has been specified directly by a surface to be interpolated (typically this point is on the outer boundary of a mesh, not the extreme boundary of the composite mesh).
Overlap	The field points lying between interpolation boundary points contained in adjacent meshes.
Physical Domain	Mesh defined in a coordinate system with shape and topology that is natural for the complex geometry of the configuration being represented.
Recipient Mesh	Mesh that receives information from another mesh.
Region	Collection of one or more volumes.

Surface	Part of a level surface of a mesh. Collections of one or more surfaces comprise a boundary.
Valid Interpolation	An interpolation where all eight mesh points of the interpolation stencil are field points (not hole or interpolation boundary points).
Volume	Group of points defined by J, K, and L index ranges. Collections of one or more volumes comprise a region.

APPENDIX B
PROGRAM VARIABLES

The following is a list of the variables used within the program. Listed are the variable name, type, and description. The CHARACTER variable type is abbreviated with the letters CH (i.e., CH*1 is CHARACTER*1). The variables are grouped within COMMON blocks as they are found within the program.

Name	Type	Description
COMMON/MESHPARM/		
NMESH	INTEGER	number of meshes
NAME(ICHAR,MDIM)	CH*1	mesh name
LINK(ICHAR,MDIM,LNDIM)	CH*1	name of meshes linked to this mesh
NLINK(MDIM)	INTEGER	number of meshes linked to this mesh
LINKALL(MDIM)	LOGICAL	=.TRUE. if all mesh links are to be connected to this mesh
MMODE(MDIM)	CH*3	='ADD' or 'SUB'
IFRINGE(MDIM)	INTEGER	number of fringe points (= 1 or 2)
QUALITY(3,MDIM)	REAL	values of interpolation quality for mesh
EPS(MDIM)	REAL	tolerance of interpolation coefficient
XINCLD(2,MDIM)	REAL	range of X values to be included for interpolation
YINCLD(2,MDIM)	REAL	range of Y values to be included for interpolation
ZINCLD(2,MDIM)	REAL	range of Z values to be included for interpolation
JINCLD(2,MDIM)	INTEGER	range of J indices to be included for interpolation
KINCLD(2,MDIM)	INTEGER	range of K indices to be included for interpolation
LINCLD(2,MDIM)	INTEGER	range of L indices to be included for interpolation
INCLD(2,MDIM)	LOGICAL	specifies whether include parameter specified for mesh
PHANTOM(MDIM)	LOGICAL	=.TRUE. if phantom mesh =.FALSE. if not a phantom mesh

Name	Type	Description
COMMON/BOUNPARM/		
BNAME(ICHAR,NBDIM)	CH*1	boundary name
NBOUN	INT	number of boundaries
IPARTMB(ICHAR,NBDIM)	CH*1	name of mesh that this boundary is part of
BMODE(NBDIM)	CH*3	= 'ADD' or 'SUB'
BTYPE(NBDIM)	CH*5	= 'HOLE' or 'OUTER'
MHOLEIN(ICHAR,NBDIM, MHDIM)	CH*1	name of mesh that this boundary makes a hole in
NHOLEIN(NBDIM)	INTEGER	number of meshes that this boundary makes holes in
CLOSED(NBDIM)	LOGICAL	= .TRUE.(default) if boundary described by surfaces is closed (used only for hole boundaries described by surfaces)
COMMON/SURFPARM/		
NSURF	INTEGER	total number of surfaces
JRANGE(2,NSDIM)	INTEGER	range of J indices of surface
KRANGE(2,NSDIM)	INTEGER	range of K indices of surface
LRANGE(2,NSDIM)	INTEGER	range of L indices of surface
NVOUT(NSDIM)	CH*2	outward normal vector of surface (required for hole boundaries)
IPARTBS(ICHAR,NSDIM)	CH*1	name of boundary which this surface is a part of
ISMESH(ICHAR,NSDIM)	CH*1	name of mesh that this surface is created from
COMMON/BOXPARM/		
NBOX	INTEGER	number of boxes
XRANGE(2,NXDIM)	REAL	minimum and maximum X values of box
YRANGE(2,NXDIM)	REAL	minimum and maximum Y values of box

Name	Type	Description
COMMON/BOXPARM/ (Continued)		
ZRANGE(2,NXDIM)	REAL	minimum and maximum Z values of box
IPARTBX(ICHAR,NXDIM)	CH*1	name of boundary that this box is a part of
COMMON/REGNPARM/		
RNAME(ICHAR,NRDIM)	CH*1	region name
IPARTMR(ICHAR,NRDIM)	CH*1	name of mesh that this region is a part of
RMODE(NRDIM)	CH*3	= 'ADD' or 'SUB'
RTYPE(NRDIM)	CH*4	= 'HOLE' or 'INTR'
NREGN	INTEGER	number of regions
COMMON/VOLPARM/		
NVOL	INTEGER	number of volumes
JRANGEV(2,NVDIM)	INTEGER	range of J indices defining the volume
KRANGEV(2,NVDIM)	INTEGER	range of K indices defining the volume
LRANGEV(2,NVDIM)	INTEGER	range of L indices defining the volume
IPARTR(ICHAR,NVDIM)	CH*1	name of region number that this volume is a part of
COMMON/MESHLINK/		
MLINK(MDIM,LNDIM)	INTEGER	mesh number that is linked
COMMON/BOUNLINK/		
IHNUM(MDIM)	INTEGER	number of holes made in mesh
IHOLE(MDIM,NBDIM)	INTEGER	boundaries that make holes in mesh
IONUM(MDIM)	INTEGER	number of outer boundaries in mesh
IOUTER(MDIM,NBDIM)	INTEGER	outer boundaries in mesh

Name	Type	Description
COMMON/SURFLINK/		
NUSURF(NBDIM)	INTEGER	number of surfaces that make up boundary
IDSURF(NBDIM,NSDIM)	INTEGER	surfaces that make up boundary
COMMON/BOXLINK/		
NUBOX(NBDIM)	INTEGER	number of boxes that make up boundary
IDBOX(NBDIM,NXDIM)	INTEGER	boxes that make up boundary
COMMON/REGNLINK/		
IRNUM(MDIM)	INTEGER	number of regions in mesh
IREGN(MDIM,NRDIM)	INTEGER	regions in mesh
COMMON/VOLLINK/		
NUVOL(NRDIM)	INTEGER	number of volumes that make up region
IDVOL(NRDIM,NVDIM)	INTEGER	volumes that make up region
COMMON/MESHCNTL/		
NEW(MDIM)	LOGICAL	= .TRUE. for initial run or if mesh is added
ADD(MDIM)	LOGICAL	= .TRUE. when boundary or region is added to mesh
SUB(MDIM)	LOGICAL	= .TRUE. when a boundary or region is subtracted from mesh
COMMON/POINTERS/		
IPIP(MDIM,MDIM)	INTEGER	array pointer for interpolation points
IPMS(MDIM)	INTEGER	array pointer for grids (based on the grid sizes)

Name	Type	Description
COMMON/POINTERS/ (Continued)		
IPSF(NSDIM)	INTEGER	array pointer for surfaces
IPIN(MDIM)	INTEGER	array pointer for points to be interpolated
COMMON/GLOBAL/		
IFRNGGL	INTEGER	global value for FRINGE no.
QUALGL	REAL	global value for QUALITY
EPSGL	REAL	global value for EPS
XINCGL(2)	REAL	global value for XINCLD
YINCGL(2)	REAL	global value for YINCLD
ZINCGL(2)	REAL	global value for ZINCLD
COMMON/RSTART/		
RESTART	LOGICAL	=.TRUE. if PEGSUS restart file read in
NMESHR	INTEGER	number of meshes in restart
NBOUNR	INTEGER	number of boundaries in restart
NSURFR	INTEGER	number of surfaces in restart
NBOXR	INTEGER	number of boxes in restart
NREGNR	INTEGER	number of regions in restart
NVOLR	INTEGER	number of volumes in restart
COMMON/RFLAGS/		
ADDM	LOGICAL	=.TRUE. if mesh added
ADDB	LOGICAL	=.TRUE. if boundary added
ADDR	LOGICAL	=.TRUE. if region added
SUBM	LOGICAL	=.TRUE. if mesh subtracted
SUBB	LOGICAL	=.TRUE. if boundary subtracted
SUBR	LOGICAL	=.TRUE. if region subtracted

Name	Type	Description
COMMON/SURF/		
ISURF(MSSMAX)	INTEGER	absolute index of surface point in mesh
VNX(MSSMAX)	REAL	magnitude of normal in X direction of surface point
VNY(MSSMAX)	REAL	magnitude of normal in Y direction of surface point
VNZ(MSSMAX)	REAL	magnitude of normal in Z direction of surface point
XS(MSSMAX)	REAL	X coordinate of surface point
YS(MSSMAX)	REAL	Y coordinate of surface point
ZS(MSSMAX)	REAL	Z coordinate of surface point
COMMON/ERROR/		
NUMERR	INTEGER	number of errors found
COMMON/NMLIST/		
AGLOBAL	CH*2000	global namelist input
AMESH(MDIM)	CH*2000	mesh namelist input
ABOUND(NBDIM)	CH*2000	boundary namelist input
ASURF(NSDIM)	CH*2000	surface namelist input
ABOX(NXDIM)	CH*2000	box namelist input
AREGN(NRDIM)	CH*2000	region namelist input
AVOL(NVDIM)	CH*2000	volume namelist input
COMMON/IWORK/		
IWORK1(MLEN)	INTEGER	integer work array
IWORK2(MLEN)	INTEGER	integer work array
IWORK3(MLEN)	INTEGER	integer work array

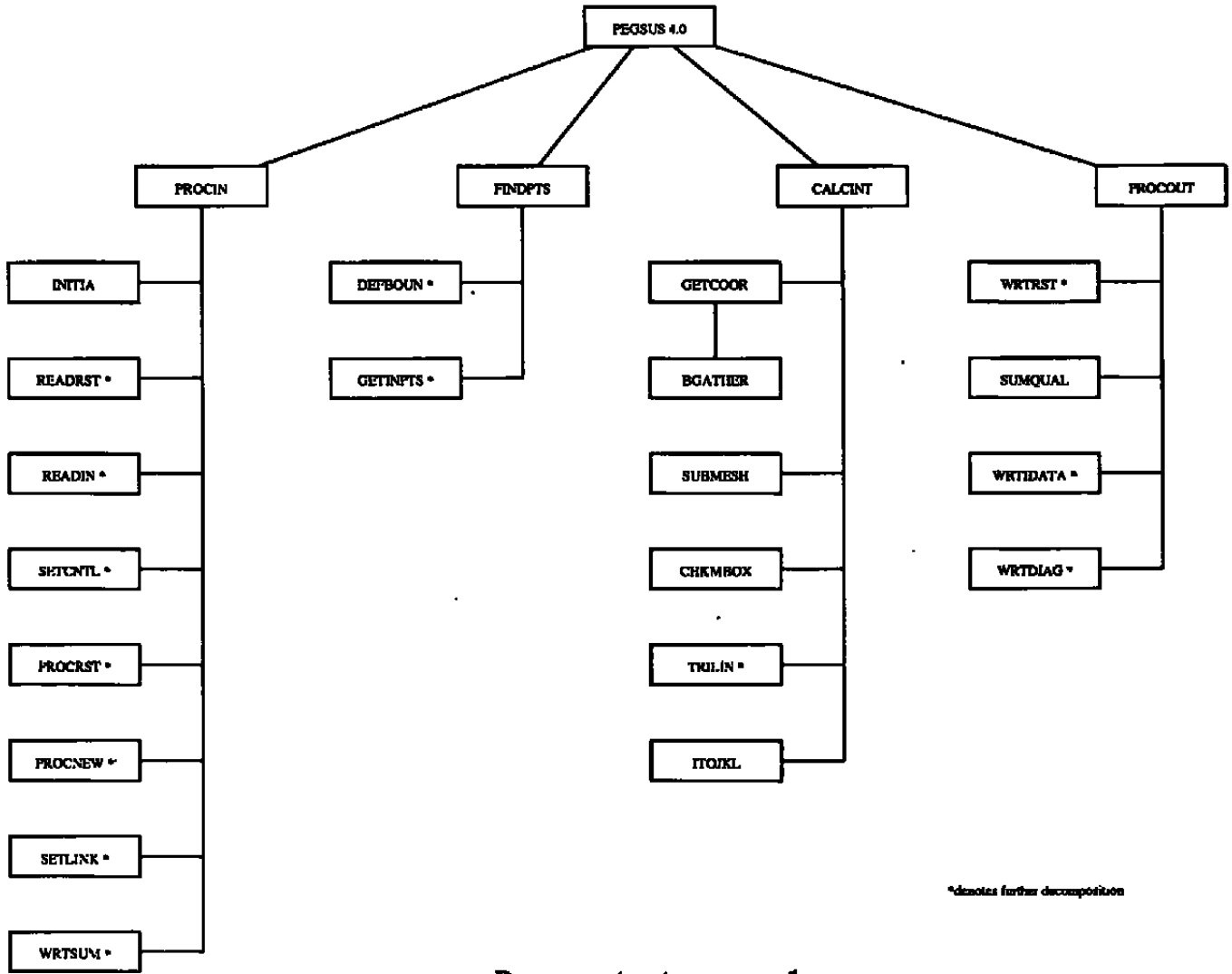
Name	Type	Description
COMMON/INVAL/		
NMESHI	INTEGER	number of meshes input
NBOUNI	INTEGER	number of boundaries input
NSURFI	INTEGER	number of surfaces input
NBOXI	INTEGER	number of boxes input
NREGNI	INTEGER	number of regions input
NVOLI	INTEGER	number of volumes input
COMMON/QUALTOT/		
NUMQUAL(11)	INTEGER	number of interpolated boundary points found for different ranges of minimum QUALITY
COMMON/MESH/		
X(MLEMAX)	REAL	X coordinate of grid
Y(MLEMAX)	REAL	Y coordinate of grid
Z(MLEMAX)	REAL	Z coordinate of grid
COMMON/BOXG/		
XMAXG	INTEGER	maximum X of domain
XMING	INTEGER	minimum X of domain
YMAXG	INTEGER	maximum Y of domain
YMING	INTEGER	minimum Y of domain
ZMAXG	INTEGER	maximum Z of domain
ZMING	INTEGER	minimum Z of domain
COMMON/BOXM/		
XMAXM(MDIM)	INTEGER	maximum X of mesh
XMINM(MDIM)	INTEGER	minimum X of mesh
YMAXM(MDIM)	INTEGER	maximum Y of mesh
YMINM(MDIM)	INTEGER	minimum Y of mesh
ZMAXM(MDIM)	INTEGER	maximum Z of mesh
ZMINM(MDIM)	INTEGER	minimum Z of mesh

Name	Type	Description
COMMON/BOXH/		
XMAXH(NBDIM)	INTEGER	maximum X of hole boundary
XMINH(NBDIM)	INTEGER	minimum X of hole boundary
YMAXH(NBDIM)	INTEGER	maximum Y of hole boundary
YMINH(NBDIM)	INTEGER	minimum Y of hole boundary
ZMAXH(NBDIM)	INTEGER	maximum Z of hole boundary
ZMINH(NBDIM)	INTEGER	minimum Z of hole boundary
COMMON/MESHVAL/		
MJMAX(MDIM)	INTEGER	maximum J for each mesh
MKMAX(MDIM)	INTEGER	maximum K for each mesh
MLMAX(MDIM)	INTEGER	maximum L for each mesh
X0(MDIM)	REAL	translation value of X
Y0(MDIM)	REAL	translation value of Y
Z0(MDIM)	REAL	translation value of Z
XR(MDIM)	REAL	X coordinate of rotation point
YR(MDIM)	REAL	Y coordinate of rotation point
ZR(MDIM)	REAL	Z coordinate of rotation point
ALPHA(MDIM)	REAL	Euler angle for rotation about Z axis
BETA(MDIM)	REAL	Euler angle for rotation about Y axis
GAMA(MDIM)	REAL	Euler angle for rotation about X axis
SCALE(MDIM)	REAL	scaling value
DELTA(MDIM)	REAL	minimum diagonal of a cell in each mesh multiplied by 0.0001
COMMON/INTRPTR/		
NUINTR(MDIM,MDIM)	INTEGER	number of boundary (or stencil) points for each mesh-to-mesh interface
COMMON/ILIST/		
NULIST(MDIM)	INTEGER	number of points for interpolation
INLIST(MIMAX1)	INTEGER	points for interpolation

Name	Type	Description
COMMON/PTRS/		
IIPNTS(MDIM)	INTEGER	number of interpolation stencils in each mesh
IBPNTS(MDIM)	INTEGER	number of interpolation boundary points in each mesh
IIEPTR(MDIM)	INTEGER	ending pointer for the interpolation stencil list for each mesh
IISPTR(MDIM)	INTEGER	starting pointer for the interpolation stencil list for each mesh
COMMON/INTERP/		
DXINT(MIMAX2)	REAL	X direction interpolation coefficient
DYINT(MIMAX2)	REAL	Y direction interpolation coefficient
DZINT(MIMAX2)	REAL	Z direction interpolation coefficient
JI(MIMAX2)	INTEGER	J index of stencil point
KI(MIMAX2)	INTEGER	K index of stencil point
LI(MIMAX2)	INTEGER	L index of stencil point
JB(MIMAX2)	INTEGER	J index of boundary point
KB(MIMAX2)	INTEGER	K index of boundary point
LB(MIMAX2)	INTEGER	L index of boundary point
IBC(MILEN)	INTEGER	interpolation boundary point pointer
COMMON/BLANK/		
IBLANK(MLEMAX)	INTEGER	iblank array

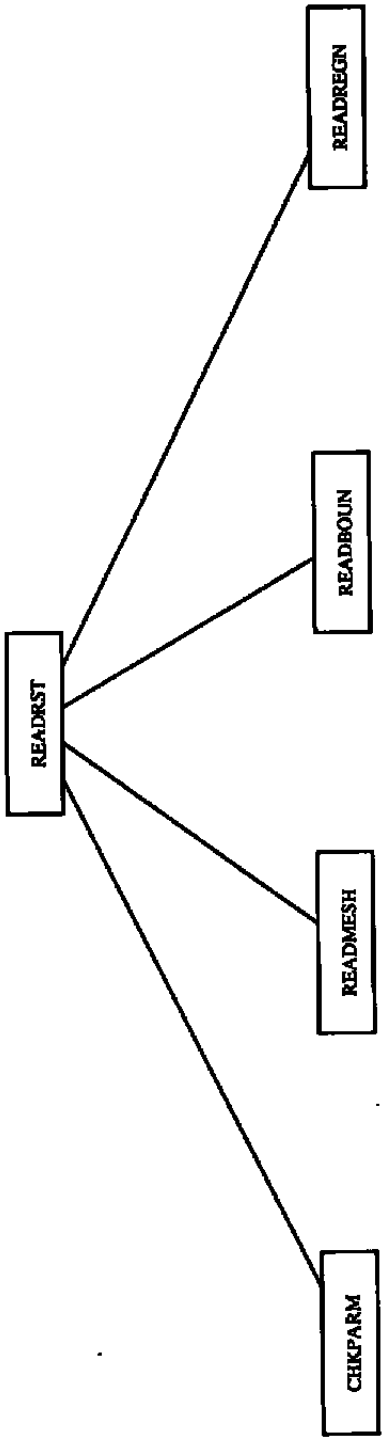
APPENDIX C
PROGRAM STRUCTURE

Figure C-1 depicts the hierarchical structure of PEGSUS 4.0. Each box (with the exception of the PEGSUS 4.0 box, which is the main program) represents a subroutine or function within PEGSUS 4.0. An asterisk(*) within a box denotes that the subroutine calls other subroutines or functions which are shown on subsequent pages.

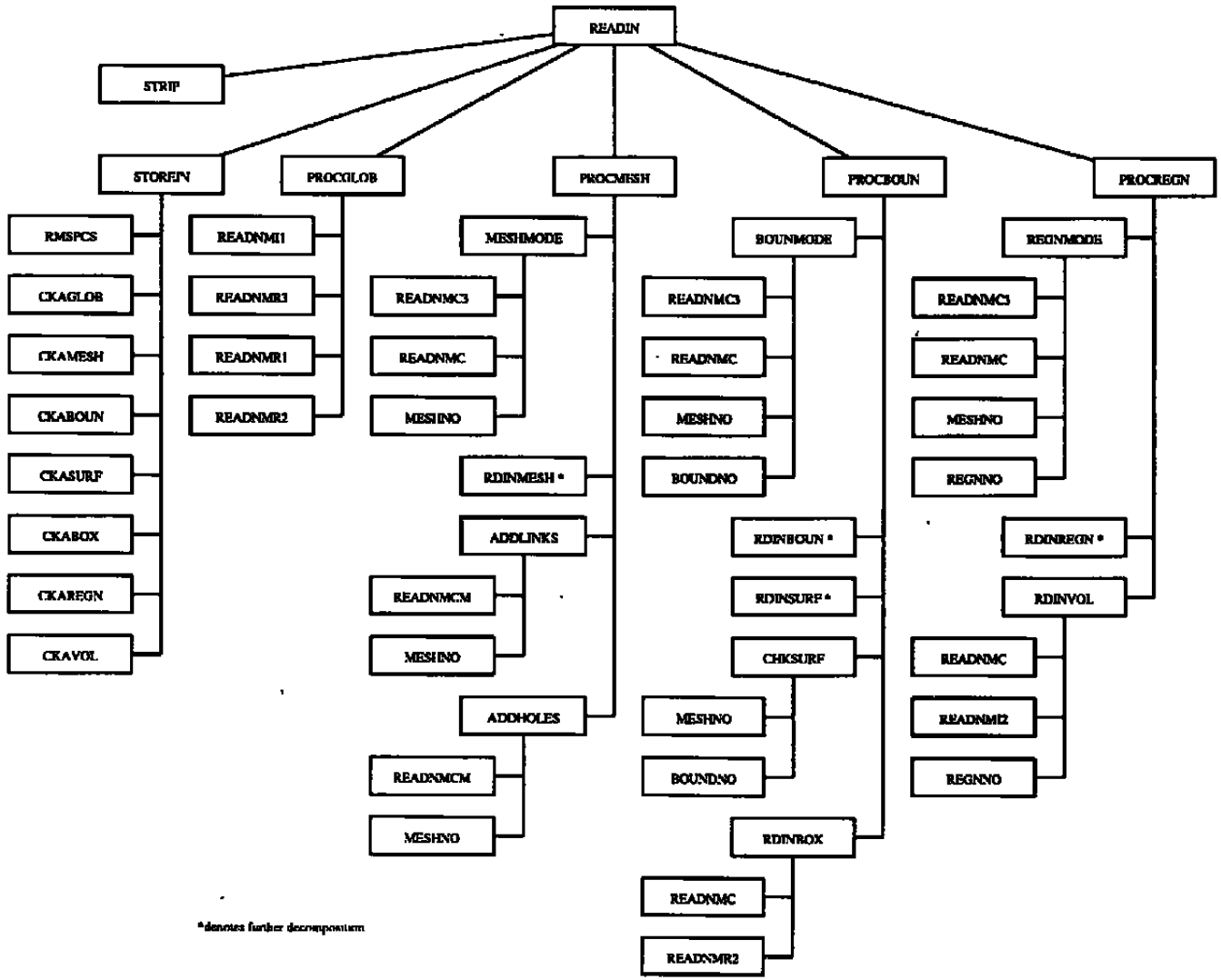


a. Program structure, page 1

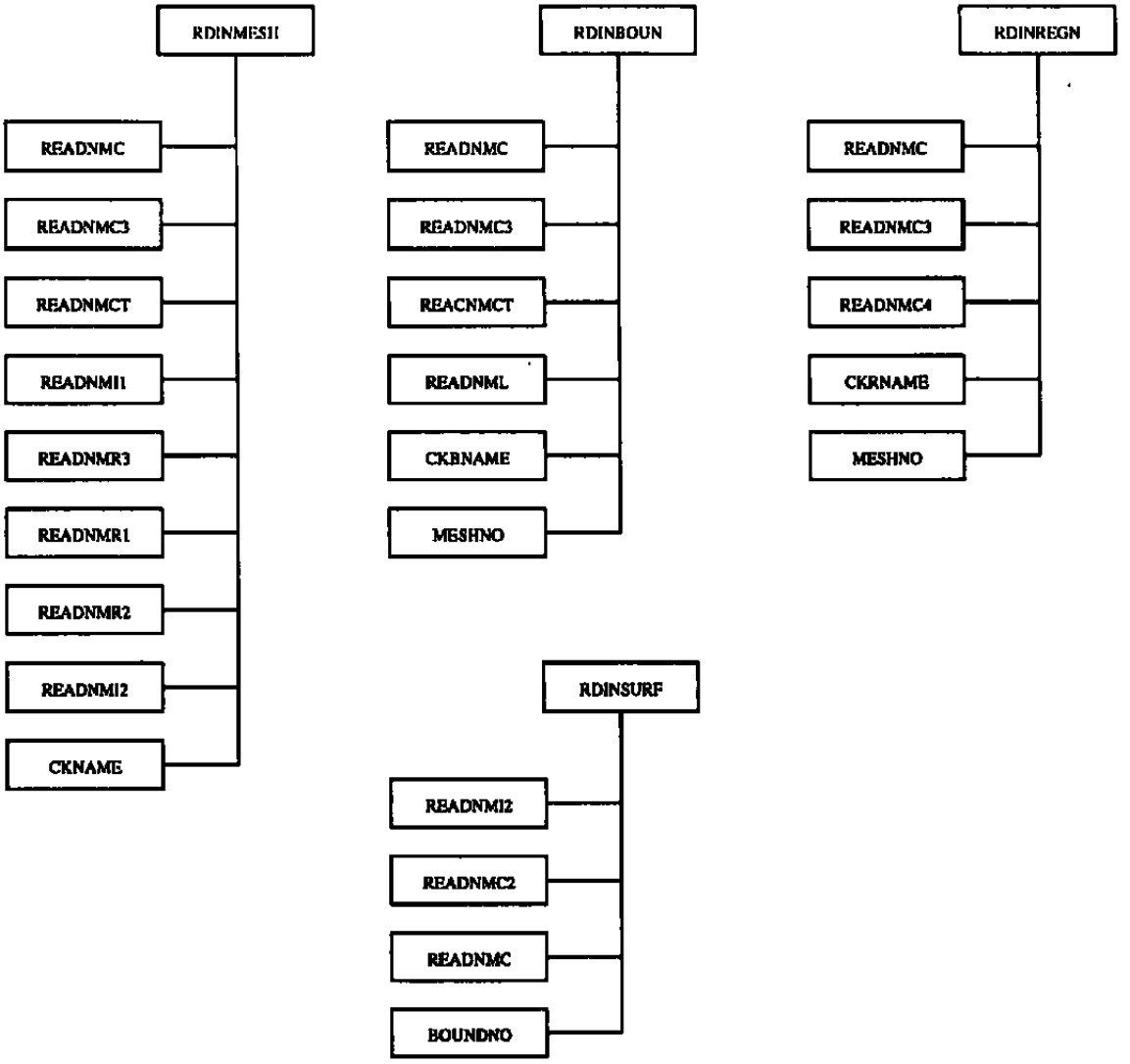
Figure C-1. Program structure.



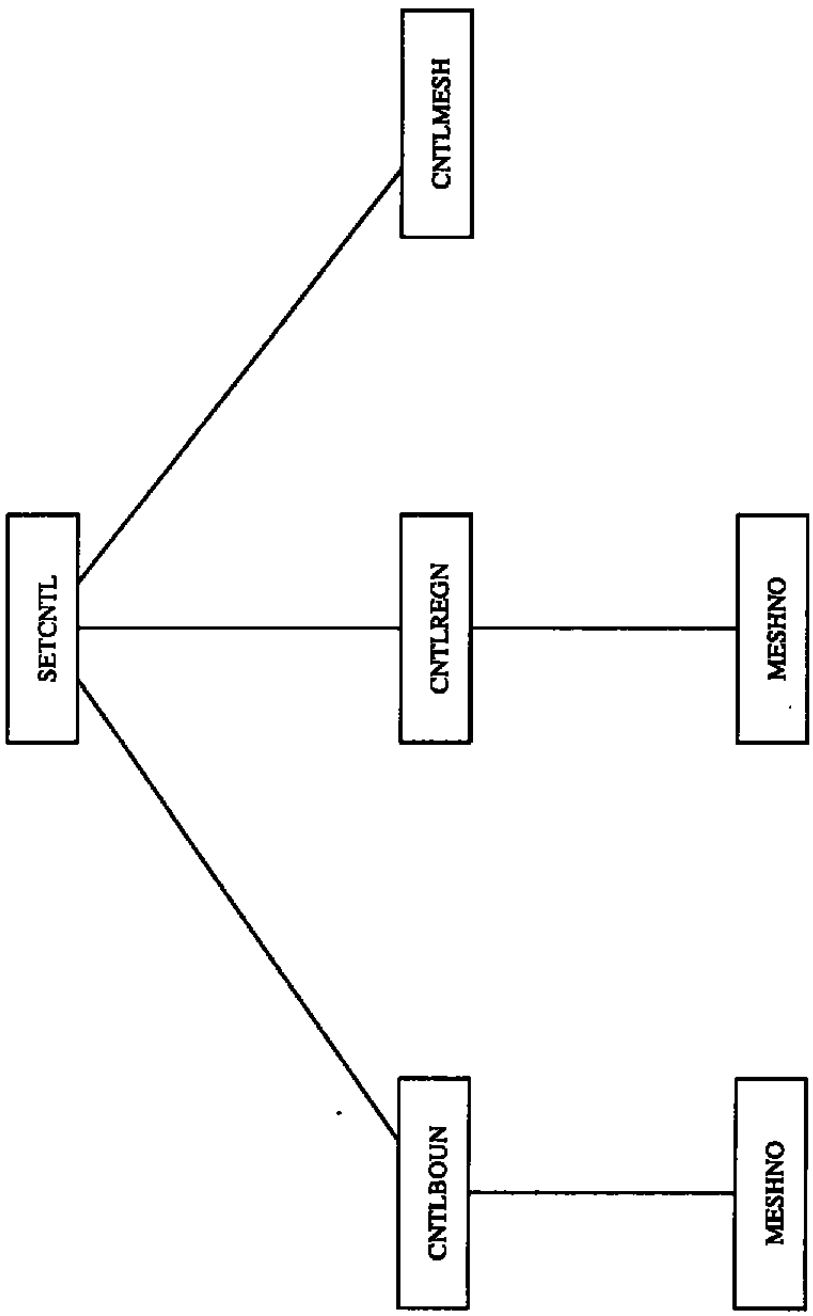
b. Program structure, page 2
Figure C-1. Continued.



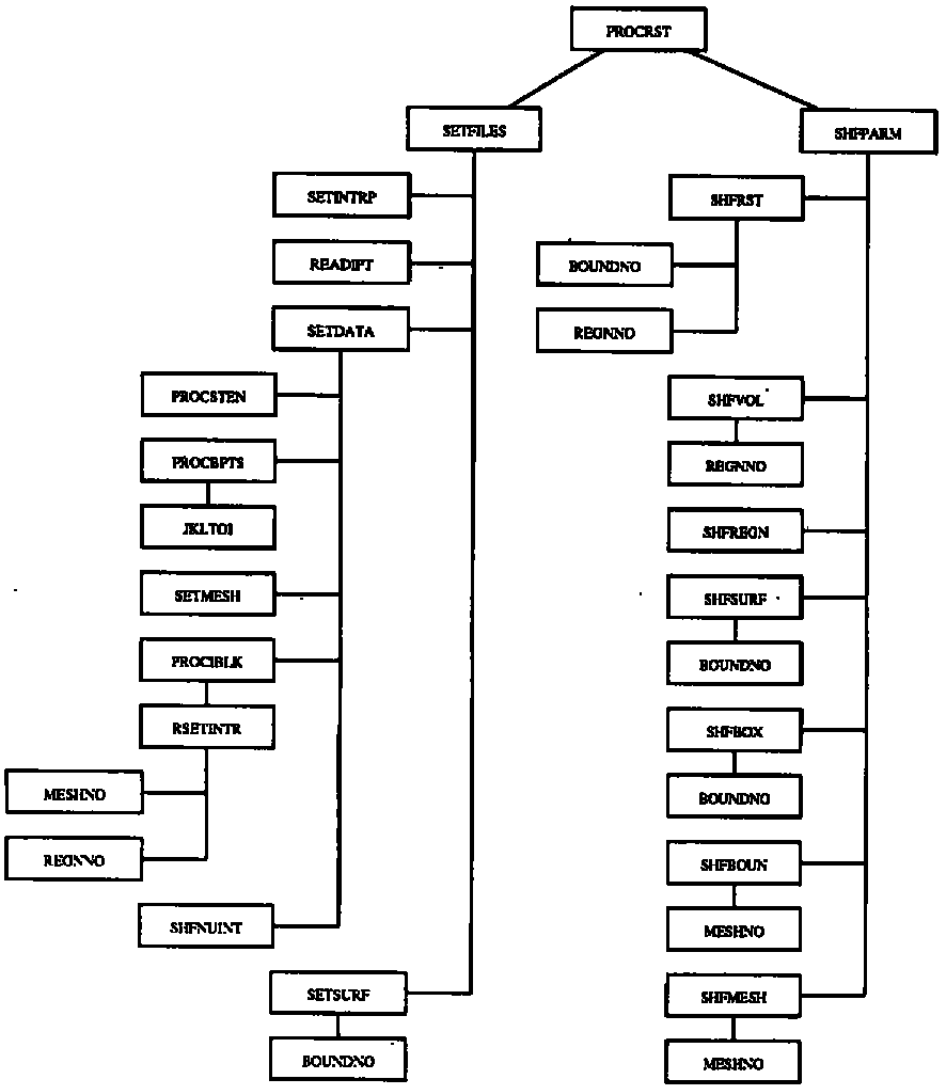
c. Program structure, page 3
Figure C-1. Continued.



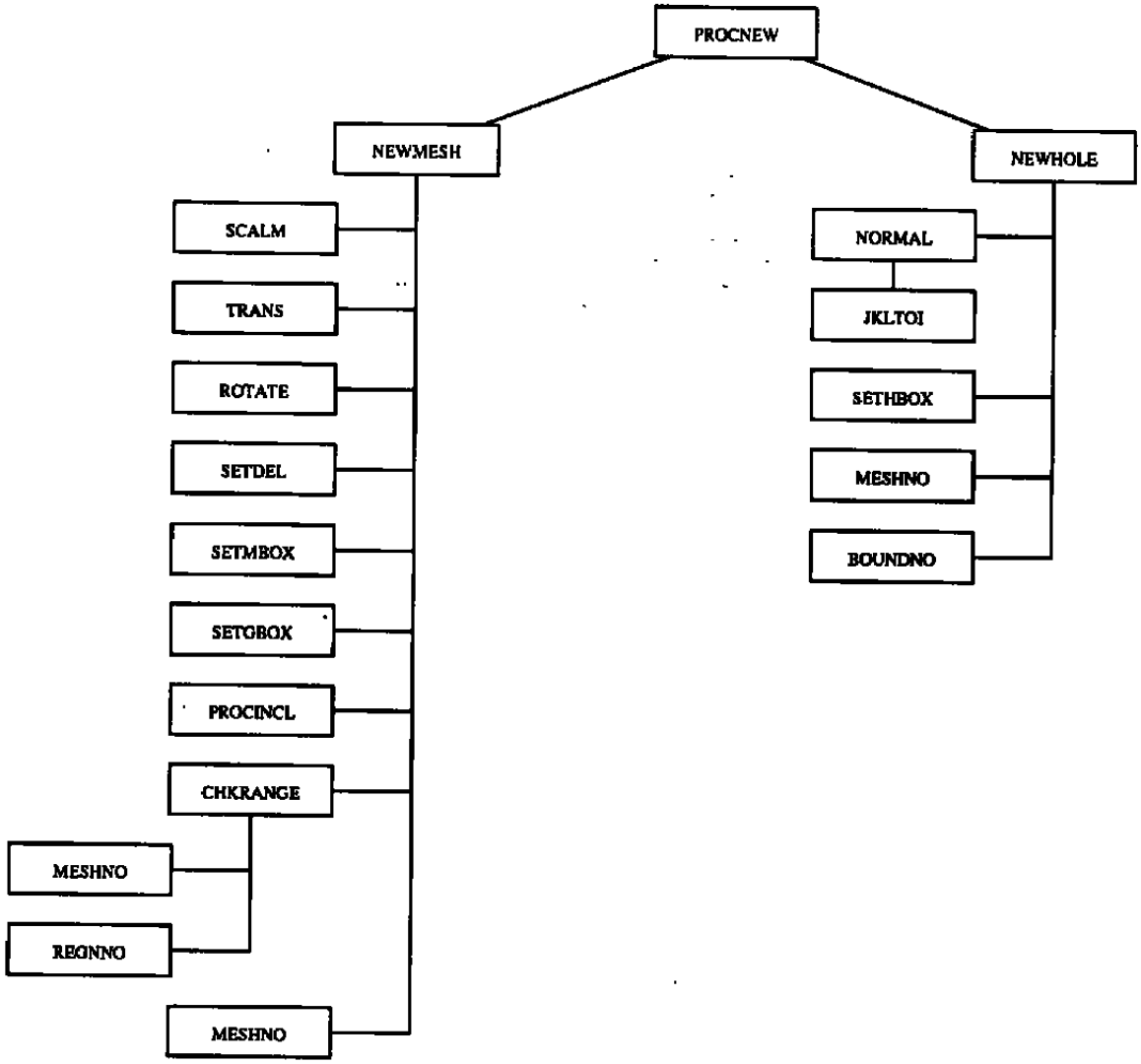
d. Program structure, page 4
Figure C-1. Continued.



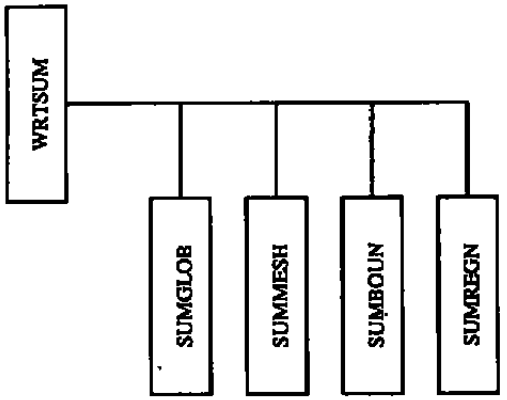
e. Program structure, page 5
Figure C-1. Continued.

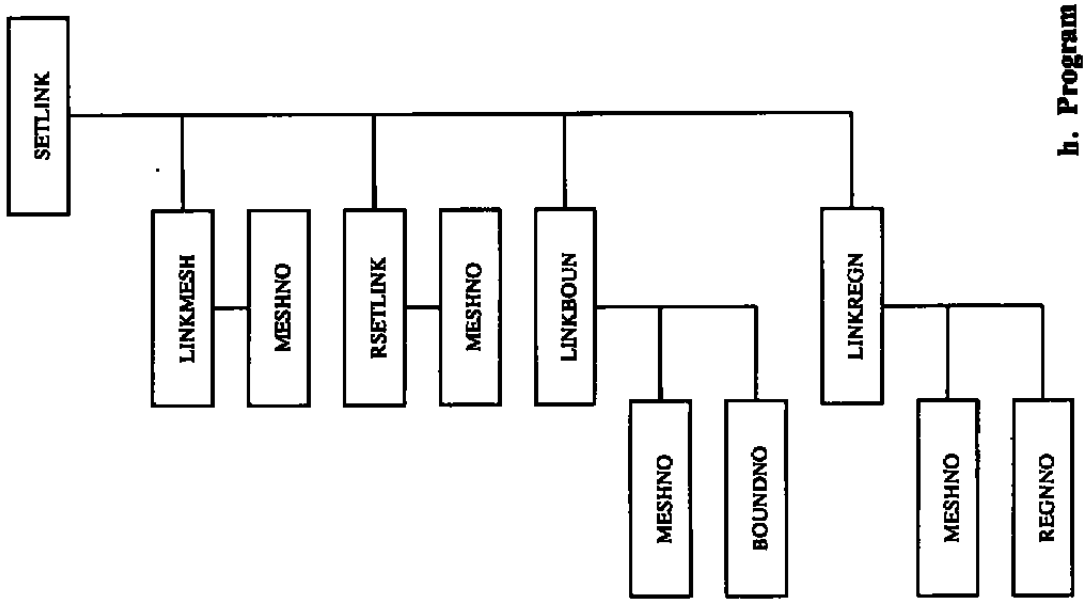


f. Program structure, page 6
Figure C-1. Continued.

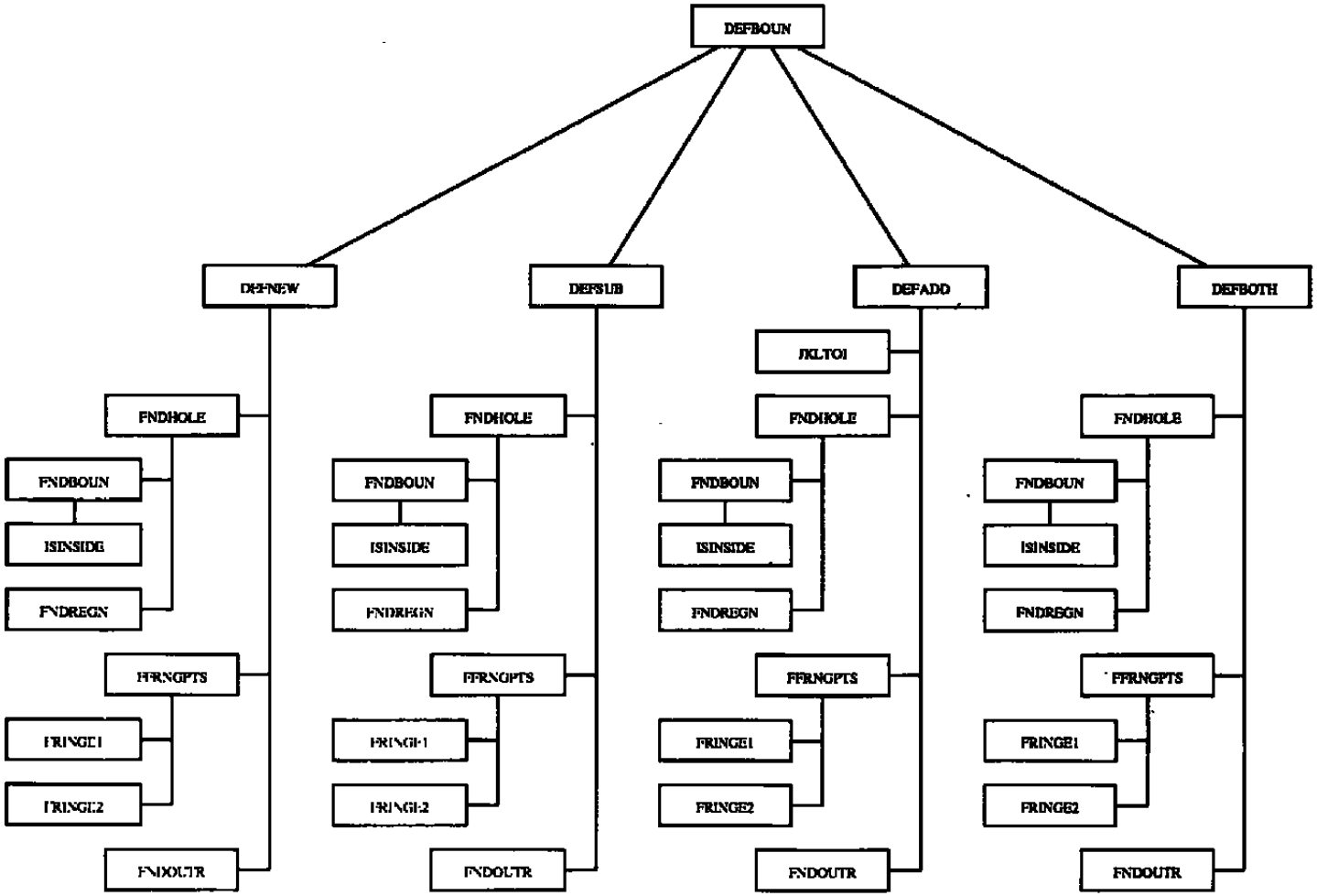


g. Program structure, page 7
Figure C-1. Continued.

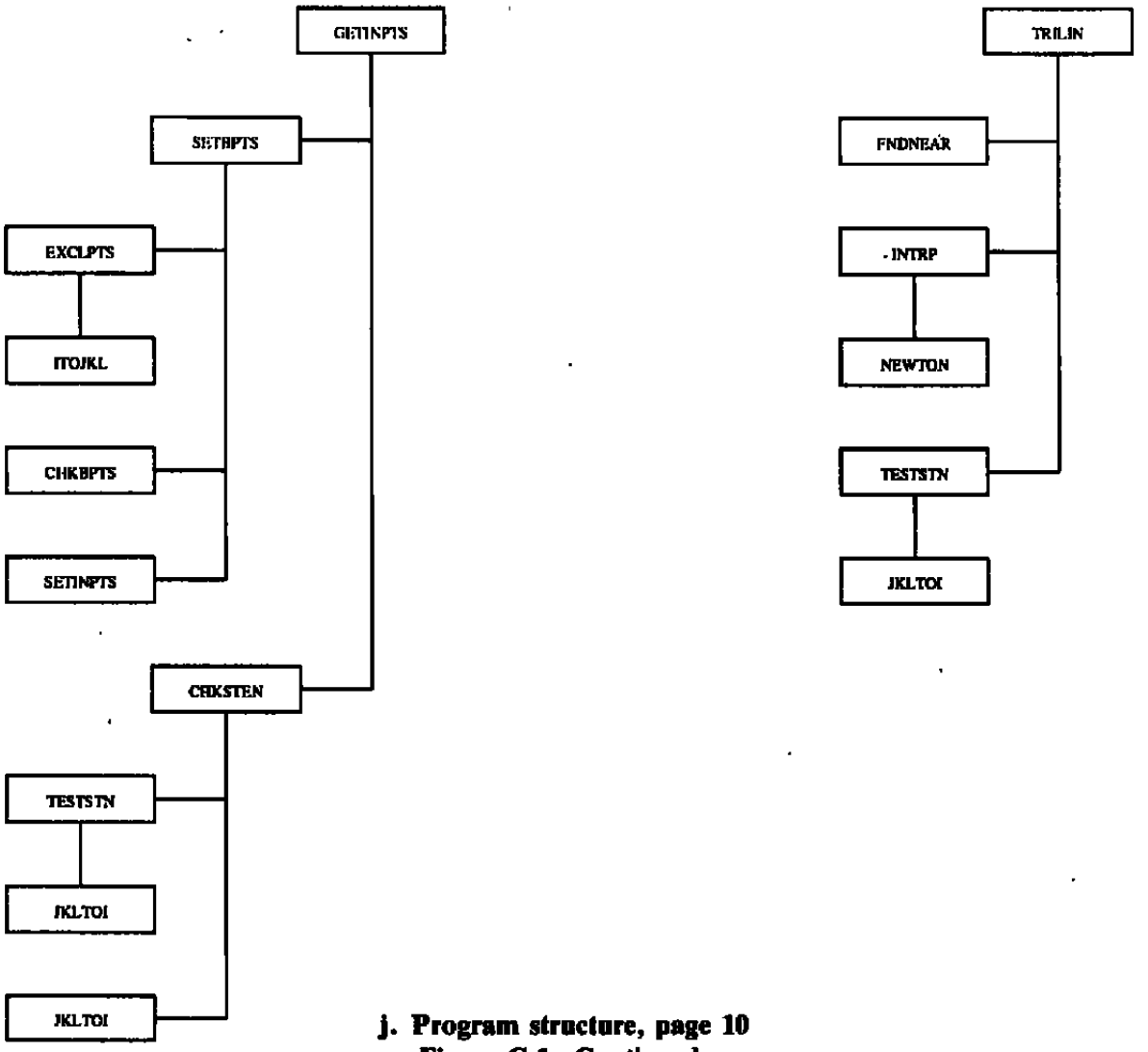




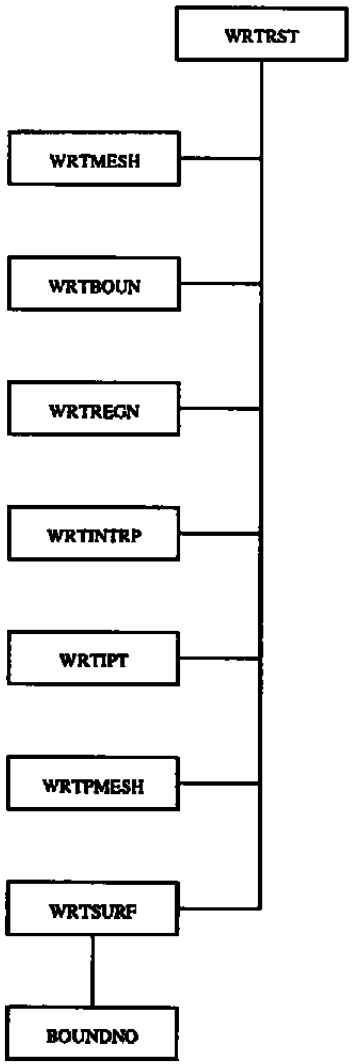
h. Program structure, page 8
Figure C-1. Continued.

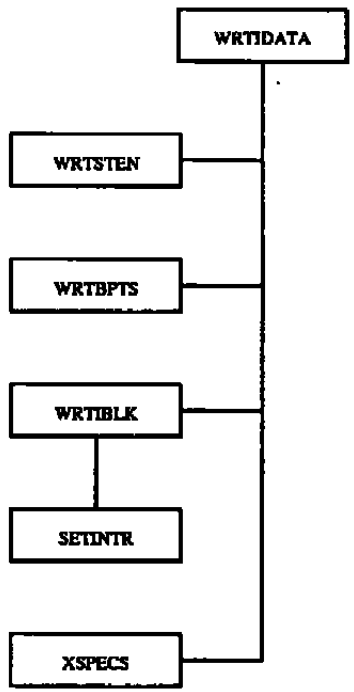


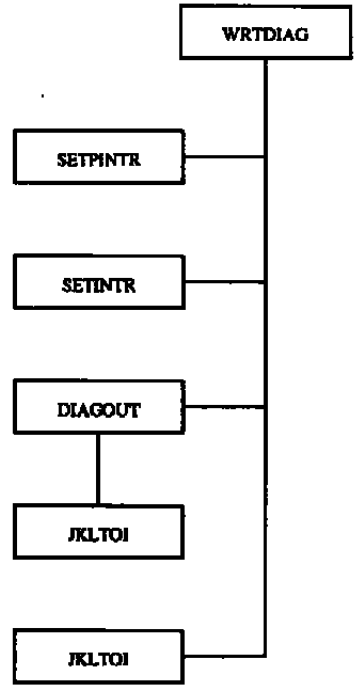
i. Program structure, page 9
Figure C-1. Continued.



j. Program structure, page 10
Figure C-1. Continued.







k. Program structure, page 11
Figure C-1. Concluded.

APPENDIX D
SUBROUTINE DESCRIPTION

ADDHOLES	Processes user input that specifies added hole boundaries.
ADDLINKS	Processes user input that specifies added links.
BGATHER	Performs gather-type function. Target and source arrays may have different dimensions.
BOUNDNO	Given a boundary name, returns the boundary number.
BOUNMODE	For a restart condition, processes the user input based on the boundary mode parameter.
CALCINT	From a list of points requiring interpolation, finds interpolation coefficients.
CHKBPTS	For a restart condition, checks previous boundary points to determine if they are still valid in current run.
CHKMBOX	Given a point and a mesh box (minimum/maximum region), determines if point is inside the mesh box.
CHKPARM	For restart condition, checks current parameters against previous parameters.
CHKRANGE	Checks the J, K, and L index ranges for surfaces and volumes to ensure that they are between 1 and the maximum values for their respective mesh.
CHKSTEN	For restart condition, checks all stencils to determine if previous interpolations are still valid. If not, they are removed from the list of points that have been interpolated and are added to the list of points that need interpolation.
CHKSURF	Checks surface inputs to determine if connected to existing mesh.
CKABOUN	Checks the key words for user input $BOUNDARY.

CKABOX	Checks the key words for user input $BOX.
CKAGLOB	Checks the key words for user input $GLOBAL.
CKAMESH	Checks the key words for user input $MESH.
CKAREGN	Checks the key words for user input $REGION.
CKASURF	Checks the key words for user input $SURFACE.
CKAVOL	Checks the key words for user input $VOLUME.
CKBNAME	Given a boundary name, checks all previous boundary names for duplication.
CKNAME	Given a mesh name, checks all previous mesh names for duplication.
CKRNAME	Given a region name, checks all previous region names for duplication.
CNTLBOUN	Sets control flags based on boundaries.
CNTLMESH	Sets control flags based on meshes.
CNTLREGN	Sets control flags based on regions.
DEFADD	Defines all interpolated boundary points for meshes with addition.
DEFBOTH	Defines all interpolation boundary points for meshes with both subtraction and addition.
DEFBOUN	Defines all interpolation boundary points.
DEFNEW	Defines all interpolation boundary points for new meshes.
DEFSUB	Defines all interpolation boundary points for meshes with subtraction.
DIAGOUT	Writes out the diagnostic maps for a given mesh.

EXCLPTS	Excludes specified interpolation boundary points based on user inputs XINCLUDE, YINCLUDE, ZINCLUDE, JINCLUDE, KINCLUDE, and LINCLUDE.
FFRNGPTS	Finds fringe points surrounding hole points.
FINDPTS	Finds all boundary points that require interpolation.
FNDBOUN	Finds the hole points defined by a boundary.
FNDHOLE	Finds all hole points within a mesh.
FNDNEAR	Given a point, finds the nearest point in the given mesh subset.
FNDOUTR	Sets IBLANK array for outer boundary points.
FNDREGN	Finds the hole points as defined by a region.
FRINGE1	Finds a single fringe surrounding all hole points.
FRINGE2	Finds a double fringe surrounding all hole points.
GETCOOR	From a list of indices, gets the X, Y, and Z coordinates.
GETINPTS	Determines which boundary points identified in DEFBOUN require interpolation. Boundary and stencil points are corrected for configuration changes.
INITIA	Initializes all COMMON blocks and opens I/O files.
INTRP	Computes interpolation coefficients for trilinear interpolation using Newton's method. Performs secondary function of "stencil-walking" during search for interpolation stencil.
ISINSIDE	Returns .TRUE. if a mesh point is inside a surface; .FALSE. otherwise.
ITOJKL	Calculates J, K, and L indices from a given absolute index I.
JKLTOI	Calculates an absolute index I from given J, K, and L indices.

LINKBOUN	Sets the boundary links based on user inputs.
LINKMESH	Sets the mesh links based on user inputs.
LINKREGN	Sets the region links based on user inputs.
MESHMODE	For a restart condition, processes the user inputs based on the mesh mode parameter.
MESHNO	Given a mesh name, returns a mesh number.
NEWHOLE	Sets up surface normals and the hole box for new holes.
NEWMESH	Reads and processes new meshes.
NEWTON	Performs trilinear interpolation using Newton's method.
NORMAL	Calculates normal vectors at all points on a surface. The surface is assumed to be defined in a right-handed coordinate system. A left-handed system will result in surface normals facing inward, which will cause holes to be generated incorrectly.
PEGSUS40	Main calling routine.
PROCBOUN	Processes the boundary-related values of the user inputs.
PROCBPTS	For a restart condition, reads and processes previous interpolation boundary points.
PROCGLOB	Processes global values of the user inputs.
PROCIBLK	For a restart condition, reads previous IBLANK array.
PROCIN	Reads and processes all user inputs, meshes and restart files.
PROCINCL	Sets the include values XINCLUDE, etc.
PROCMESH	Processes the mesh values of the user inputs.
PROCNEW	Processes new meshes and holes.

PROCOUT	Processes and writes results to appropriate files.
PROCREGN	Processes the region-related values of the user inputs.
PROCRST	Reads and processes the restart values.
PROCSTEN	Reads and processes the previous interpolation stencils and coefficients.
RDINBOUN	Reads the boundary user inputs.
RDINBOX	Reads the box user inputs.
RDINMESH	Reads the mesh user inputs.
RDINREGN	Reads the region user inputs.
RDINSURF	Reads the surface user inputs.
RDINVOL	Reads the volume user inputs.
READBOUN	Reads the boundary-related values from PEGSUS restart file.
READIN	Reads and processes all user inputs.
READIPT	For a restart condition, reads pointers for previous interpolation points.
READMESH	Reads the mesh values from PEGSUS restart file.
READNMC	NAMELIST-like read for a single-variable-name character array.
READNMC2	NAMELIST-like read for a two-character name.
READNMC3	NAMELIST-like read for a three-character name.
READNMC4	NAMELIST-like read for a four-character name.
READNMCM	NAMELIST-like read for multiple-variable-length-name character array (double dimension).

READNMCT	NAMELIST-like read for multiple-variable-length-name character array (triple dimension).
READNMI1	NAMELIST-like read for one integer value.
READNMI2	NAMELIST-like read for two integer values.
READNML	NAMELIST-like read for a logical value.
READNMR1	NAMELIST-like read for one real value.
READNMR2	NAMELIST-like read for two real values.
READNMR3	NAMELIST-like read for three real values.
READREGN	Reads region-related values from PEGSUS restart file.
READRST	Reads PEGSUS restart file.
REGNMODE	For a restart condition, processes the user inputs based on the region mode.
REGNNO	Given a region name, returns a region number.
RMSPCS	Removes the spaces from the NAMELIST-like input.
ROTATE	Rotates mesh about defined point in space.
RSETINTR	Resets the interior blanked points back to 1 before PEGSUS processing.
RSETLINK	Resets mesh links based on mesh boxes.
SCALM	Scales meshes.
SETBPTS	Gets the boundary points specified by the IBLANK array and determines which values need to be interpolated. If mesh is not new some boundary points may have already been interpolated.
SETCNTL	Sets control flags.

SETDATA	Reads and processes previous interpolation data.
SETDEL	Calculates smallest distance between points in a mesh. The values are used to define how close two points from different meshes must be in order to be considered coincident.
SETFILES	Sets up direct access files or work arrays.
SETGBOX	Calculates minimum/maximum box of the domain.
SETHBOX	Calculates minimum/maximum box of the hole creation boundary.
SETINPTS	Determines which boundary points require interpolation.
SETINTR	Sets the interior blanked region equal to 0.
SETINTRP	Reads and processes orphan points from previous PEGSUS run.
SETLINK	Sets the links based on user inputs.
SETMBOX	Calculates the minimum/maximum box of each mesh.
SETMESH	For a restart condition, reads and processes meshes from previous composite mesh file.
SETPINTR	Sets the interior blanked region equal to 0 for plotting.
SETSURF	Reads surfaces from PEGSUS restart file.
SHFBOUN	Shifts boundary values for subtracted boundaries.
SHFBOX	Shifts box values for subtracted boundaries.
SHFMESH	Shifts mesh values for subtracted meshes.
SHFNUINT	Shifts NUINTR array for subtracted meshes.
SHFPARM	Shifts all values for subtraction.
SHFREGN	Shifts region values for subtracted regions.

SHFRST	Shifts restart counters for subtracted meshes, boundaries, or regions.
SHFSURF	Shifts surface values for subtracted boundaries.
SHFVOL	Shifts volume values for subtracted regions.
STOREIN	Reads user inputs and stores them in character arrays for later processing.
STRIP	Removes comments from NAMELIST-like input.
SUBMESH	Develops two different subsets of a mesh for near-point search.
SUMBOUN	Writes a summary of boundary-related values.
SUMGLOB	Writes a summary of global values.
SUMMESH	Writes a summary of mesh values.
SUMQUAL	Writes a summary of the quality of the interpolations.
SUMREGN	Writes a summary of region-related values.
TESTSTN	Tests interpolation stencil to determine if it is valid.
TRANS	Translates mesh.
TRILIN	Finds an initial stencil reference point to begin search for valid interpolation stencil.
WRTBOUN	Writes the boundary-related values to PEGSUS restart file.
WRTBPTS	Writes the boundary points to the interpolation file.
WRTDIAG	Sets up array for diagnostic maps and writes IBLANK array for plotting.
WRTIBLK	Writes the IBLANK array to the interpolation file.
WRTIDATA	Writes the interpolation file.

WRTINTRP	Writes orphan points to the PEGSUS restart file.
WRTIPT	Writes the NUINTR array to the PEGSUS restart file.
WRTMESH	Writes the mesh values to the PEGSUS restart file.
WRTPMESH	Writes the phantom meshes to the PEGSUS restart file.
WRTREGN	Writes the region-related values to the PEGSUS restart file.
WRTRST	Writes the PEGSUS restart file.
WRTSTEN	Writes the interpolation stencils and coefficients to the interpolation file.
WRTSUM	Writes the summary of user inputs and meshes.
WRTSURF	Writes the surface points and normals to the PEGSUS restart file.
XSPECS	Calculates and writes parameter specification for flow solver.

APPENDIX E
DIRECT ACCESS FILES

The following are stored as direct access files during the execution of PEGSUS 4.0 with INCORE = 0.

Mesh Coordinates

Unit	12
Record Format	X(MLEN), Y(MLEN), Z(MLEN)
Index	Mesh ID number

Surface Definitions

Unit	13
Record Format	XS(MSLEN), YS(MSLEN), ZS(MSLEN), VNX(MSLEN), VNY(MSLEN), VNZ(MSLEN)
Index	Surface ID number

Blanked Points

Unit	14
Record Format	IBLANK(MLEN)
Index	Mesh ID number

Boundary Points

Unit	15
Record Format	JB(MSLEN), KB(MSLEN), LB(MSLEN)
Index	Mesh ID number

Interpolation Stencil Reference Points

Unit	16
Record Format	JI(MSLEN), KI(MSLEN), LI(MSLEN)
Index	Mesh ID number

Interpolation Coefficients

Unit	17
Record Format	DXINT(MSLEN), DYINT(MSLEN), DZINT(MSLEN)
Index	Mesh ID number

Points Requiring Interpolation

Unit	19
Record Format	INLIST(MILEN)
Index	Mesh ID number

APPENDIX F
PEGSUS OUTPUT FILE STRUCTURE

Five output files are defined: (1) composite mesh file, (2) interpolation data file, (3) PEGSUS restart file, (4) IBLANK array for plotting file, and (5) orphan list file.

Composite Mesh File

This file is written to file COMPOUT. Each mesh NM is written separately. The order in which the meshes are written is the order in which the meshes are specified in the user input by $MESH.

Record Number	Format
1	NM, MJMAX(NM), MKMAX(NM), MLMAX(NM), MNAME
2	(X(I),I=I1,I2), (Y(I),I=I1,I2), (Z(I),I=I1,I2)

where I1 = 1 and I2 = MJMAX(NM)*MKMAX(NM)*MLMAX(NM)

Interpolation Data File

This file is written to file INTOUT. The data for each mesh NM are written separately.

Record Number	Format
1	IBPNTS(NM),IIPNTS(NM),IIEPTR(NM), IISPTR(NM),MJMAX(NM),MKMAX(NM), MLMAX(NM)
2	(JI(I),KI(I),LI(I),DXINT(I),DYINT(I),DZINT(I), I=1,IIPNTS(NM))
3	(JB(I),KB(I),LB(I),IBC(I),I=1,IBPNTS(NM))
4	(IBLANK(I),I=I1,I2)

where I1 = 1 and I2 = MJMAX(NM)*MKMAX(NM)*MLMAX(NM)

PEGSUS Restart File

This file is written to file RSTOUT. The PEGSUS restart file is divided into eight parts: (1) global values, (2) mesh values, (3) boundary-related values, (4) region-related values, (5) points requiring interpolation (orphan points), (6) interpolation-list pointer, (7) phantom meshes, and (8) hole creation surfaces.

(1) Global Values

Record Number	Format
1	MDIM, LNDIM, ICHAR, NBDIM, MHDIM, NSDIM, MSLEN,NXDIM, NRDIM, NVDIM, MLEN, MLEMAX, MILEN
2	IFRNGGL, (QUALGL(N),N = 1,3), EPSGL, (XINCGL(N), YINCGL(N), ZINCGL(N), N = 1,2), INCLDGL, XMAXG, XMING, YMAXG, YMING, ZMAXG, ZMING

(2) Mesh Values

Record Number	Format
1	NMESH

The following record is written for each mesh NM from 1 to NMESH.

2	(NAME(IC,NM),IC = 1,ICHAR), IFRINGE(NM), (QUALITY(N,NM),N = 1,3), EPS(NM), (XINCLD(N,NM), YINCLD(N,NM), ZINCLD(N,NM), JINCLD(N,NM), KINCLD(N,NM), LINCLD(N,NM),N = 1,2), INCLD(NM), NLINK(NM), LINKALL(NM), PHANTOM(NM), ((LINK(IC,NM,NL),IC = 1,ICHAR),NL = 1, NLINK(NM)), XMAXM(NM), XMINM(NM), YMAXM(NM), YMINM(NM), ZMAXM(NM), ZMINM(NM),

MJMAX(NM), MKMAX(NM),
MLMAX(NM), X0(NM),
Y0(NM), Z0(NM),
XR(NM), YR(NM),
ZR(NM), ALPHA(NM),
BETA(NM), GAM(NM),
SCALE(NM), DELTA(NM)

(3) Boundary-Related Values

Record Number	Format
1	NBOUN

The following record is written for each boundary NB from 1 to NBOUN.

2	(BNAME(IC,NB),IC = 1,ICHAR), (IPARTMB(IC,NB),IC = 1,ICHAR), BTYPE(NB), NHOLEIN(NB), ((MHOLEIN(IC,NB,NH),IC = 1,ICHAR), NH = 1,NHOLEIN(NB)), XMAXH(NB), YMAXH(NB), ZMAXH(NB), XMINH(NB), YMINH(NB), ZMINH(NB), CLOSED(NB)
3	NSURF

The following record is written for each surface NS from 1 to NSURF

4	(JRANGE(I,NS), KRANGE(I,NS), LRANGE(I,NS), I = 1,2), NVOUT(NS), (IPARTBS(IC,NS),IC = 1,ICHAR), (ISMESH(IC,NS),IC = 1,ICHAR)
5	NBOX

The following record is written for each box NX from 1 to NBOX.

6	(XRANGE(I,NX), YRANGE(I,NX), ZRANGE(I,NX), I = 1,2), (IPARTBX(IC,NX),IC = 1,ICHAR)

(4) Region-Related Values

Record Number	Format
1	NREGN

The following record is written for each region NR from 1 to NREGN.

2	(RNAME(IC,NR),IC = 1,ICHAR), (IPARTMR(IC,NR),IC = 1,ICHAR), RTYPE(NR)
3	NVOL

The following record is written for each volume NV from 1 to NVOL.

4	(JRANGEV(I,NV), KRANGEV(I,NV), LRANGEV(I,NV), I = 1,2), (IPARTR(IC,NV),IC = 1,ICHAR)

(5) Points Requiring Interpolation

Record Number	Format

Records 1 and 2 are written for each mesh NM from 1 to NMESH.

1	NULIST(NM)

The following record is written only if NULIST(NM) is greater than 0.

2	(INLIST(I),I = I1,I2)

where I1 = 1 and I2 = NULIST(NM).

(6) Interpolation-List Pointer

Record Number	Format

The following record is written for each recipient mesh NMR (outer loop) from 1 to NMESH and each donor mesh NMD (inner loop) from 1 to NMESH.

1	NUINTR(NMR,NMD)

(7) Phantom Meshes

Record Number	Format

The following records are written for each mesh NM from 1 to NMESH, if PHANTOM(NM) is .TRUE.

1	NM, MJMAX(NM), MKMAX(NM), MLMAX(NM), MNAME
2	(X(I),I=I1,I2), (Y(I),I=I1,I2), (Z(I),I=I1,I2)

where I1 = 1 and I2 = MJMAX(NM)*MKMAX(NM)*MLMAX(NM).

(8) Hole Creation Surfaces

Record Number	Format

The following records are written for each surface NSURF that points to a hole creation boundary (BTYPE='HOLE').

1	NSPTS(NS)
2	(XS(I), YS(I), ZS(I), VNX(I), VNY(I), VNZ(I), I=I1,I2)

where I1 = 1 and I2 = NSPTS(NS).

IBLANK Array for Plotting File

This file is written to file IBPLOT.

Record Number	Format

This record is written for each mesh NM (except phantom meshes) from 1 to NMESH.

1	(IBLANK(I),I=1,MJKL)

where MJKL = MJMAX(NM)*MKMAX(NM)*MLMAX(NM).

Orphan List File

This file is written to file ORPHAN.

Record Number	Format
This record is written for each orphan point.	
1	MESH, J, K, L

APPENDIX G
EXAMPLE OF COMPLEX CONFIGURATION

This example is taken from actual computations performed on the Cray X-MP/12 at AEDC. PEGSUS 4.0 and the XMERA flow solver (Ref. 5) have been successfully executed for each part of this example. The meshes used were generated with AEDC codes and sized for the available memory of the AEDC Cray. An improved understanding of PEGSUS 4.0 should be accomplished by careful study of this example, just as was the case in Sec. 4.0 of the main text.

The example is presented in three steps to demonstrate the ability of PEGSUS 4.0 to handle complex aerodynamic configurations as well as to illustrate many of the advanced features of PEGSUS 4.0. The three steps are shown by the three configurations in Fig. G-1. Figure G-1a depicts a rectangular cavity mounted below a flat plate. The flow is assumed to be above the plate and within the cavity only. Figure G-1b shows the same cavity configuration with the addition of a store mounted on a bent sting. The third configuration (Fig. G-1c) is also the same cavity and store but with the sting removed. A description of each mesh is given as well as the user input required to execute PEGSUS 4.0. To demonstrate the addition and subtraction capabilities of PEGSUS 4.0, the configuration with the store/sting (Fig. G-1b) will be obtained by the addition of the appropriate meshes to the configuration of the empty cavity (Fig. G-1a). Similarly, the store in a cavity without the sting (Fig. G-1c) will be obtained by subtraction and addition to the store/sting in the cavity (Fig. G-1b). These cases will also demonstrate the use of the double fringe for all interpolation boundaries.

G.1 Empty Cavity

To begin, the domain of the configuration will be split by a plane of symmetry, requiring only half of the domain to be gridded. The empty cavity configuration is divided into five computational meshes, two meshes defining the exterior region above the flat plate, and three meshes defining the cavity and an overlap region above the cavity mouth. The names of the five meshes are: 'EXTERIOR A', 'EXTERIOR B', 'CAVITY A', 'CAVITY B', and 'CAVITY C'. Shown in Figs. G-2 through G-6 are (a) the basic perspective, (b) a side view, (c) a top view, and (d) a front view of each mesh. The cavity meshes are shown to have regions that are omitted from the domain by using interior blanking. This technique will be discussed later in conjunction with Fig. G-9. Table G-1 gives the dimensions of each mesh.

The user input required to connect these meshes (see Sec. 3.2) is given in Fig. G-7. The $GLOBAL user inputs (see Fig. G-7a) for this example specify that a double fringe (FRINGE = 2) will be placed around all holes. The QUALITY user input specifies that the first attempt at interpolation of a point will be done with a QUALITY of 1.0 and the second

attempt will be done with a QUALITY of 0.5 (see Sec. 2.2.3 for the discussion on QUALITY). Also set is the value EPS equal to 0.0005. The user inputs XINCLUDE, YINCLUDE, and ZINCLUDE are set to reduce the number of orphan points that can occur when a hole boundary is near the extreme boundary of the domain (see Sec. 2.3.2). Also shown in Fig. G-7a are the user inputs required for each mesh. For this example, only the LINK values are required.

The remainder of the user input specifies the interpolation boundary points. First, the 'EXTERIOR A' mesh will interpolate flow-field values from 'EXTERIOR B' as well as from the cavity meshes. To obtain interpolated values from 'EXTERIOR B', an outer boundary is specified for constant J planes 66 and 67 (see Fig. G-7b). In order to avoid interpolation from the middle of an expected high gradient region (the region about the L = 1 plane of the 'EXTERIOR A' mesh), a hole creation region is specified within the 'EXTERIOR A' mesh. The user inputs of $REGION and $VOLUME for this hole creation region (see Sec. 2.1.2.2) are shown in Fig. G-7b. The resulting hole that is made in the 'EXTERIOR A' mesh is depicted in Fig. G-8. The interpolation boundary specification for the 'EXTERIOR B' mesh is similar to that for the 'EXTERIOR A' mesh, with an outer boundary and hole creation region being specified (see Fig. G-7c). The interpolation boundary points within the cavity meshes are all defined with outer boundary specifications (see Figs. G-7d through G-7f). For the 'CAVITY A' and 'CAVITY C' meshes five surfaces each are required. 'CAVITY B' requires six surfaces to define its outer boundary. Each of these outer boundary surfaces is composed of two constant planes in order to provide a double interpolated boundary (double fringe).

Another PEGSUS 4.0 feature used for this example is interior blanking (see Sec. 2.3.1). Interior blanking was used to keep the corner of the cavity within a single mesh that is not stretched or skewed (i.e., this region is not split with one mesh defining the top wall and another mesh defining the side wall). Depicted in Fig. G-9 is a portion of the front corner of the 'CAVITY A' mesh that has been specified by using the $REGION and $VOLUME user inputs with TYPE = 'INTR' (see Fig. G-7d). This specification sets IBLANK = 0 within the interior region. Solid wall boundary conditions are set on the surfaces of this region within the flow solver. Corner regions occur for the front, back, and side walls of the cavity. This requires the 'CAVITY A' and 'CAVITY C' meshes to have two $VOLUME inputs each and 'CAVITY B' to have one $VOLUME input (see Figs. G-7d through G-7f).

A sampling of the diagnostic maps for the empty cavity PEGSUS execution is shown in Fig. G-10. Shown in Fig. G-10a is the legend for the maps giving a key to the letters and symbols used. The maps are a representation of the computational domain for each L plane of each mesh. Only selected L plane maps are shown here. In Figs. G-10b and G-10c are shown L = 10 maps for the 'EXTERIOR A' and 'EXTERIOR B' meshes, respectively, with

the hole and its double fringe. This fringe interpolates from the 'CAVITY A', 'CAVITY B', and 'CAVITY C' meshes. The outer boundary specified for the 'EXTERIOR A' mesh at J lines 66 and 67 interpolates from 'EXTERIOR B' and 'CAVITY B'. For the 'EXTERIOR B' mesh, the outer boundary specified for J lines 1 and 2 interpolates from 'EXTERIOR A' and 'CAVITY B'. In Figs. G-10d through G-10f, the L=24 diagnostic maps for the 'CAVITY A', 'CAVITY B', and 'CAVITY C' meshes, respectively, are shown. The interior blanked region ('#' signs with no fringe points) and the double interpolation boundaries are clearly evident.

G.2 Store/Sting in a Cavity

Using the PEGSUS results from the empty cavity case, the store/sting can be added. The store and sting are each represented with separate meshes which are shown in Figs. G-11 ('STORE') and G-12 ('STING'), respectively. Another mesh ('INTERFACE') was also required to obtain satisfactory overlap between the sting and exterior meshes. Without the 'INTERFACE' mesh, the hole created by the 'STING' mesh in the 'EXTERIOR B' mesh did not have sufficient overlap between the hole and outer boundaries and thus created numerous orphan points. The 'INTERFACE' mesh is shown in Fig. G-13. The dimensions for all three meshes are given in Table G-1. In Fig. G-14, the user input for specifying this example is given. All three new meshes are added by specifying MODE='ADD' (see Fig. G-14a). To adjust the LINK lists of the meshes from the previous PEGSUS execution, the ADDLINK input is specified. For example, the 'STORE' mesh has ADDLINK names of 'EXTERIOR A', 'EXTERIOR B', 'CAVITY A', 'CAVITY B', and 'CAVITY C'. From this input, the mesh name 'STORE' will be added to the top of the LINK list of the meshes specified by ADDLINK. Since no hole creation boundaries existed in the previous execution, the ADDHOLE input was not required.

The interpolation boundary definitions for each added mesh are performed in the same manner as for an initial execution. The 'STORE', 'STING', and 'INTERFACE' meshes each make holes in the meshes from the initial execution (example G-1) as well as having outer boundaries. The user input for the hole creation and outer boundary specifications is given in Figs. G-14b through G-14d.

After creating the mesh for the sting, it was determined that part of the 'STING' mesh was inside the solid back wall of the cavity. Instead of creating a new mesh, a hole creation boundary was added to create a hole in the 'STING' mesh. Shown in Fig. G-14e is the user input specifying this hole. Surfaces from the 'CAVITY C' mesh were used to create the hole. This hole creation boundary was not closed (see Sec. 2.1.2.1) since enough of the surfaces was available to create the proper hole. The exterior, cavity, and interface meshes, with the

hole created by the 'STORE' and 'STING' meshes, are shown in Fig. G-15 for the plane of symmetry.

At this point, a detail of the use of interconnecting meshes with PEGSUS needs to be discussed. In the current example several meshes (e.g., 'EXTERIOR A' and 'EXTERIOR B' meshes) are overlapped at their boundaries. When another mesh (e.g., the 'STORE' mesh) crosses this boundary, the overlap region of the earlier two meshes must be sufficient to avoid difficulties. For this example, which uses double interpolation boundaries, the 'EXTERIOR A' and 'EXTERIOR B' meshes must have a five-point overlap, i.e., five points (or planes) that have identical values. If a four-point overlap were to be used (the minimum required for a double fringe), difficulties can arise as shown in Fig. G-16. First, in Fig. G-16a a portion of a four-point overlap for the 'EXTERIOR A' and 'EXTERIOR B' meshes is shown as well as a portion of the outer boundary of the 'STORE' mesh. In Fig. G-16b, the double interpolation boundary points for the 'EXTERIOR A' mesh are shown along with interpolation points that have valid or invalid interpolation stencils (as far as the 'EXTERIOR A' mesh is concerned) for the 'STORE' mesh. Points P1 through P3 are interpolation points with valid stencils while the stencils for P4 through P9 are invalid. Shown in Fig. G-16c are the double interpolation boundary of 'EXTERIOR B', with valid stencils for points P6 through P9 and invalid stencils for points P1 through P5 of the 'STORE' mesh outer boundary. For a four-point overlap, then, points P4 and P5 cannot find valid interpolation stencils in either mesh. If a five-point overlap is used as depicted in Fig. G-17, points P4 and P5 have valid stencils in the 'EXTERIOR B' mesh (see Fig. G-17c). For a five-point overlap, then, all points are found to have valid stencils in one mesh or the other (see Fig G-17b and G-17c). Of course, QUALITY could be used to reduce the restrictions for valid interpolation, but is not advised when the simple solution of increasing the overlap is available.

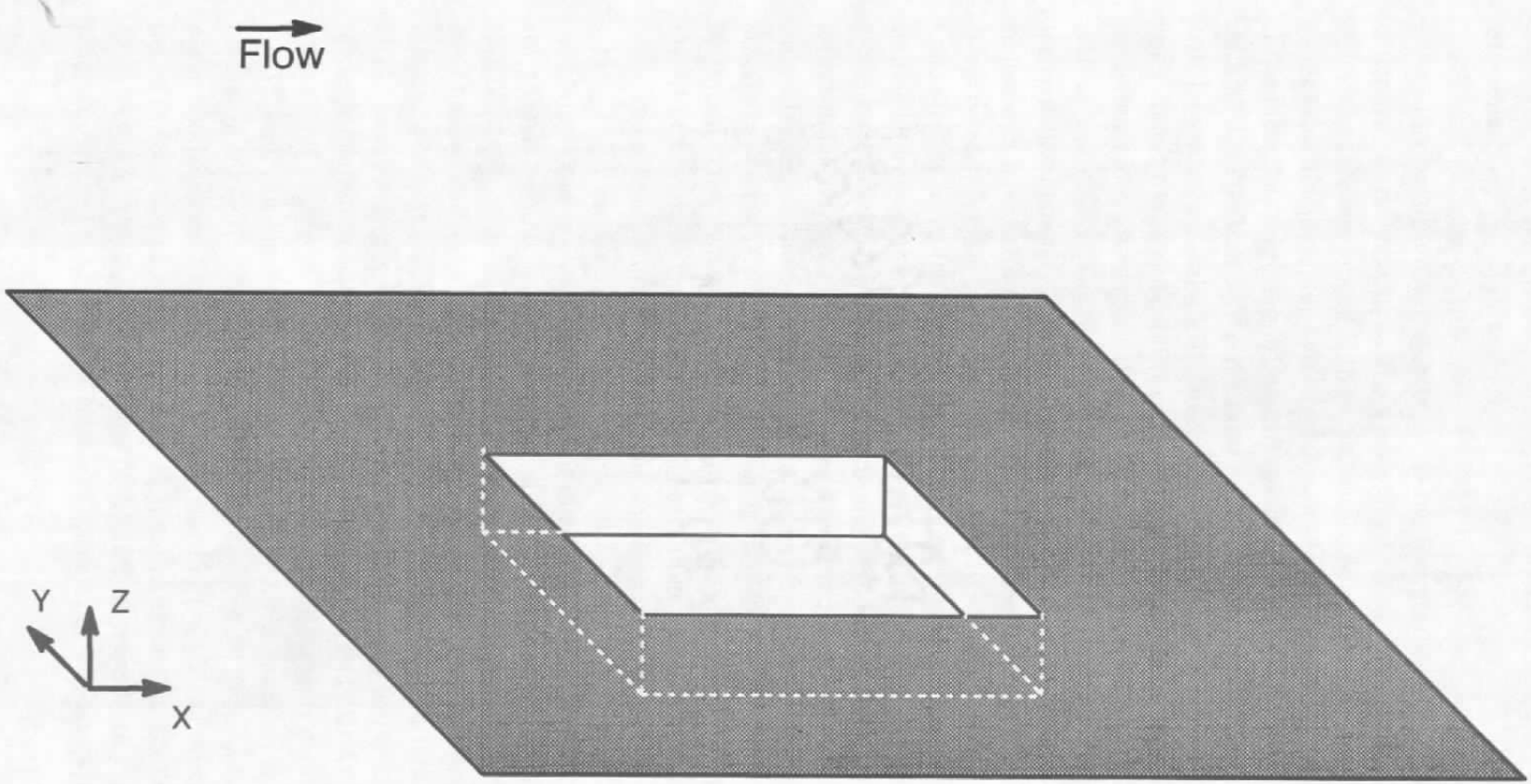
Figure G-18 presents a sampling of diagnostic maps for this case. In Fig. G-18a, a key for the letters and symbols is given. In Fig. G-18b, the L = 26 plane is shown for the 'EXTERIOR A' mesh. Depicted in this figure are the outer boundary interpolation points for J lines 66 and 67 and the interpolation stencils for the 'EXTERIOR B' and 'INTERFACE' meshes. The hole made by the 'STING' mesh in the 'EXTERIOR B' mesh is shown in Fig. G-18c. Also shown are the outer boundary connections to the 'EXTERIOR A' mesh and the interpolation stencils for the 'EXTERIOR A' and 'INTERFACE' meshes. The hole made by the 'INTERFACE' mesh in the 'EXTERIOR B' mesh is depicted in Fig. G-18d for the L = 30 plane. In Figs. G-18e through G-18h, the hole made by the 'STORE' and 'STING' meshes in the 'CAVITY A', 'CAVITY B', and 'CAVITY C' meshes is shown. The L = 29 outer plane of the 'STORE' mesh is depicted in Fig. G-18i. As can be seen in this figure, the 'STORE' mesh interpolates flow-field values from the 'EXTERIOR A', 'EXTERIOR B', 'CAVITY A', 'CAVITY B', and 'CAVITY C' meshes. In Fig. G-18j, the L = 29 outer plane for the 'STING' mesh is shown. The main item to note from this figure is the hole

that appears at J = 18 and J = 19. This hole was made by the hole creation boundary 'AFT CAVITY HOLE BOUNDARY'. Finally, in Figs. G-18k through G-18m, three L planes of the 'INTERFACE' mesh are given to show the hole created by the 'STING' mesh and the interactions of the 'INTERFACE' mesh with the 'EXTERIOR A' and 'EXTERIOR B' meshes.

G.3 Store Alone in a Cavity

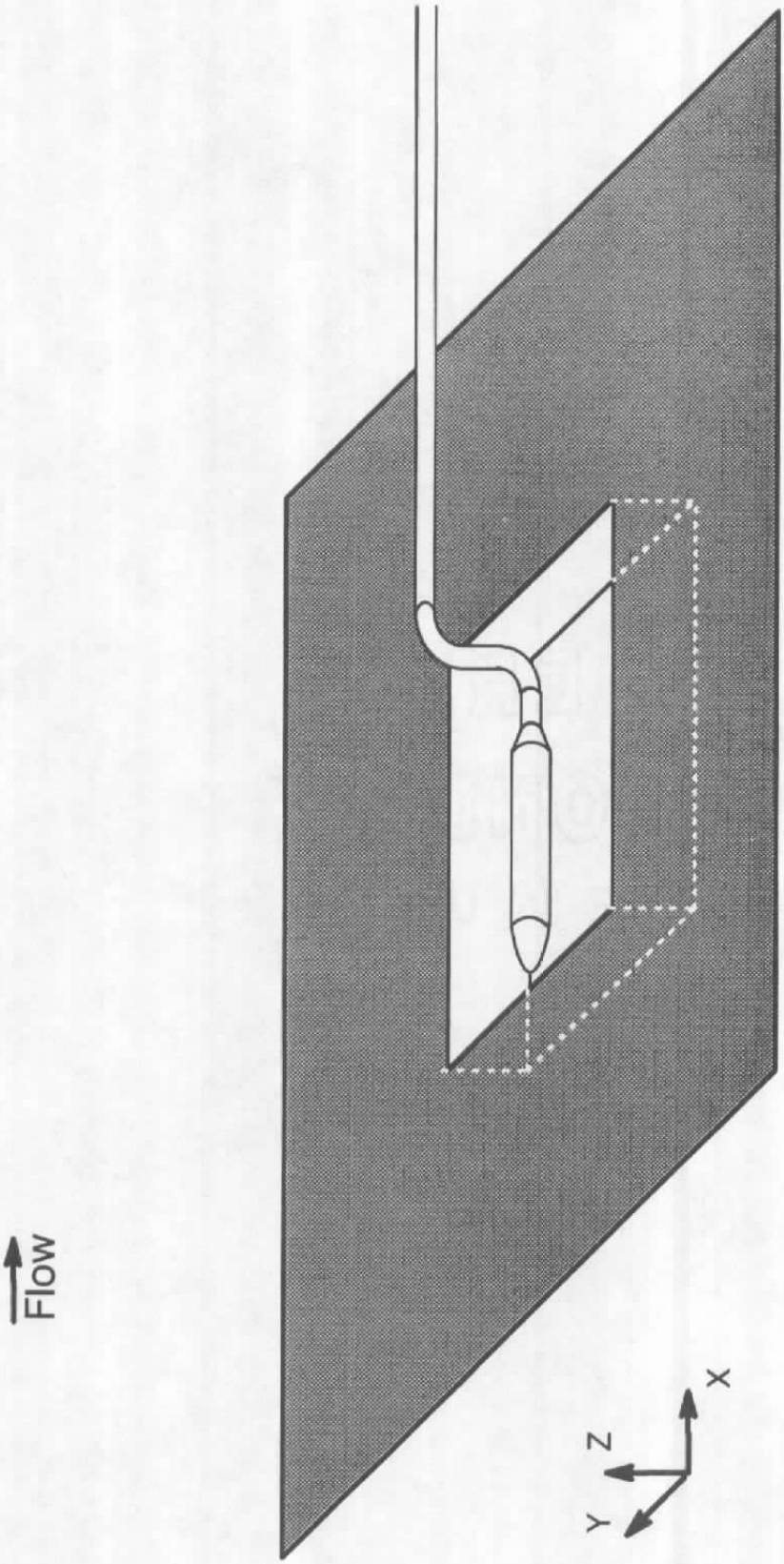
The final example demonstrates the capability of PEGSUS 4.0 to perform subtraction and addition during the same execution. The 'STING' and 'INTERFACE' meshes will be subtracted while the 'STORE CAP' mesh will be added. A description of the 'STORE CAP' mesh is shown in Fig. G-19. Table G-1 contains the dimensions of this mesh. The user input for this case is given in Fig. G-20. The mesh definitions given in Fig. G-20a show subtraction of the 'STING' and 'INTERFACE' meshes by using the input MODE = 'SUB'. All boundaries associated with these two meshes are subtracted. The 'STORE CAP' mesh is added to the configuration with the associated LINK and ADDLINK inputs, as shown in Fig. G-20a. In Fig. G-20b, the specification of the outer boundary for the 'STORE CAP' mesh and interior blanked region is given. The interior blanked region for the 'STORE CAP' is used to obtain the required overlap for the double interpolation boundary used by the 'STORE CAP' and 'STORE' meshes. An additional outer boundary is added to the 'STORE' mesh in Fig. G-20c to correctly specify the portion of the L = 1 surface of the 'STORE' mesh that was previously part of the sting surface. Finally, the 'AFT CAVITY HOLE BOUNDARY' of the 'CAVITY C' mesh is subtracted in Fig. G-20c since the 'STING' mesh is no longer present. The hole that now exists in the plane of symmetry for the exterior and cavity meshes is shown in Fig. G-21.

Figure G-22 is a sampling of diagnostic maps for this case. Figure G-22a gives a key to the letters and symbols used. Most of the diagnostic maps have no changes from those seen for example G-2, with the exception of the subtraction of the 'STING' and 'INTERFACE' meshes. Figure G-22b shows the hole made by the 'STORE' mesh in the L = 56 plane of the 'CAVITY C' mesh. In Fig. G-22c, the L = 27 plane for the 'STORE CAP' mesh is shown to highlight the interior blanked region.



a. Cavity

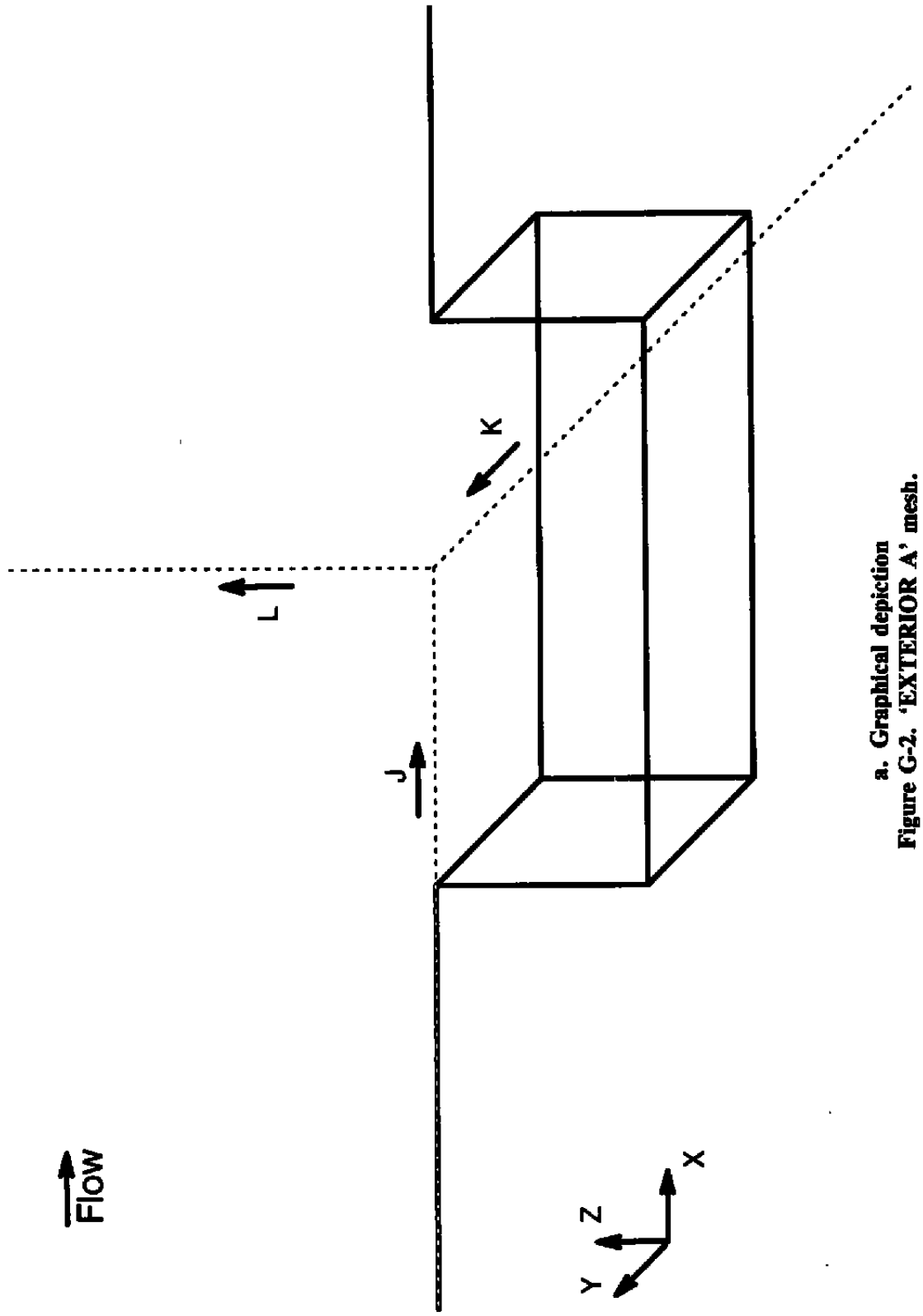
Figure G-1. Graphical depiction of complex configuration.



b. Cavity/store/sting
Figure G-1. Continued.

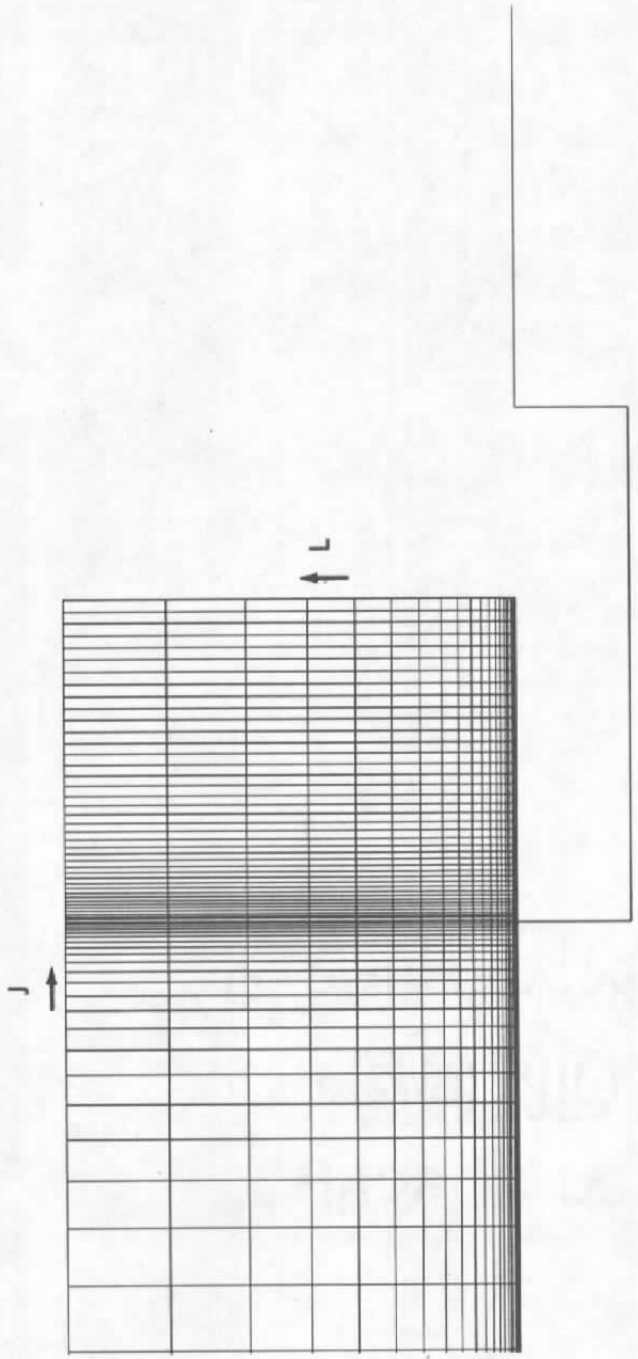
c. Cavity/store

Figure G-1. Concluded.

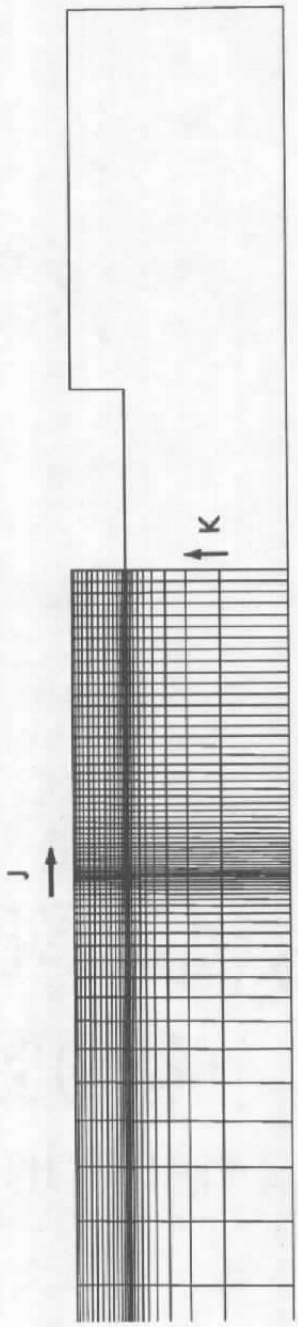


a. Graphical depiction

Figure G-2. 'EXTERIOR A' mesh.

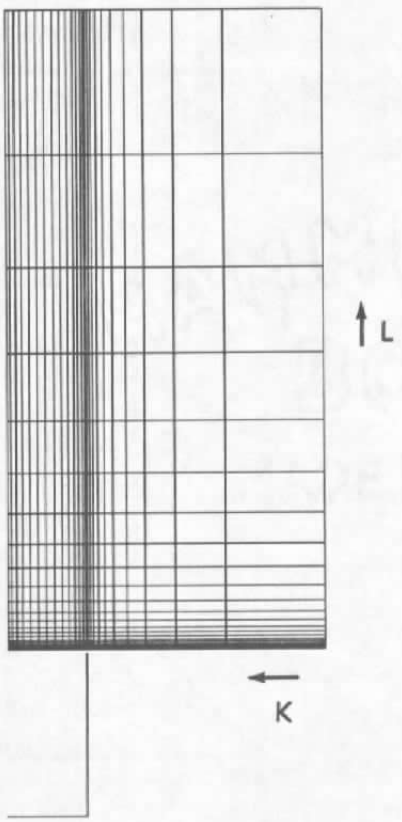


b. Side view
Figure G-2. Continued.

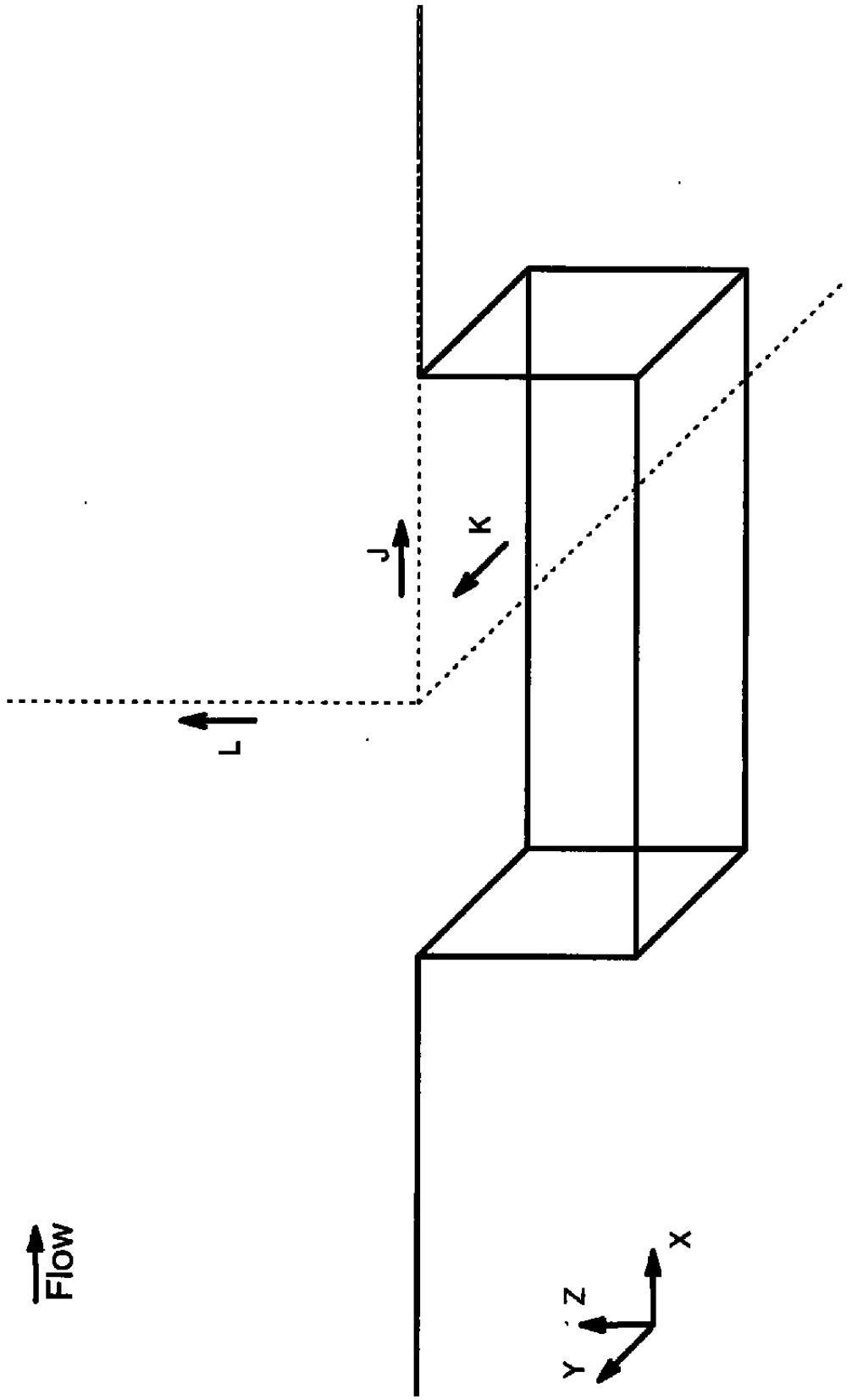


c. Top view

Figure G-2. Continued.

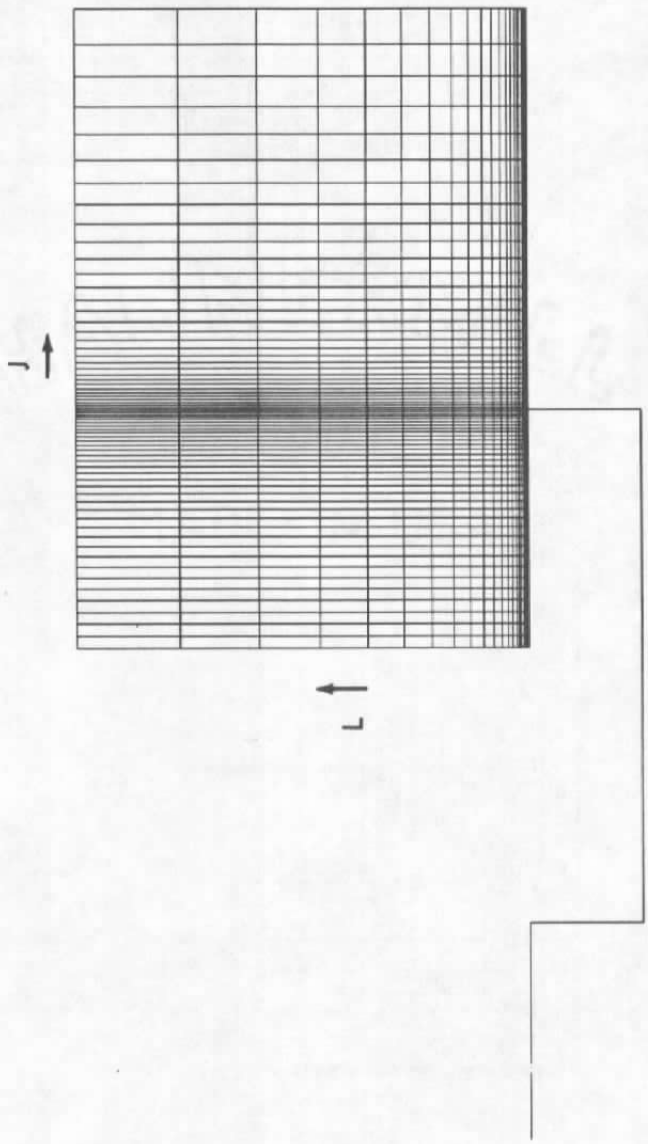


d. Front view
Figure G-2. Concluded.



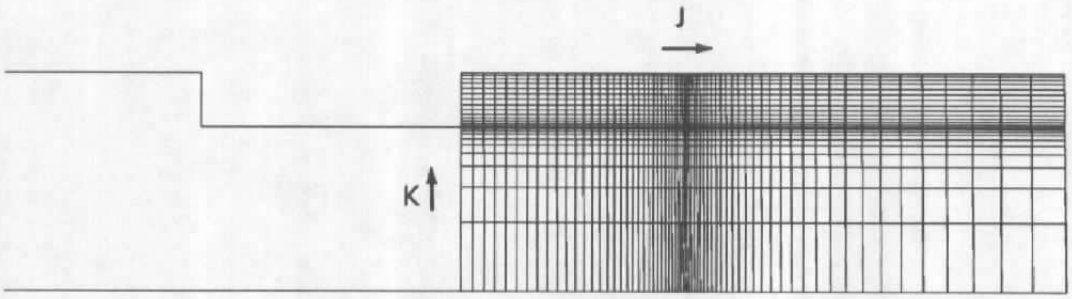
a. Graphical depiction

Figure G-3. 'EXTERIOR B' mesh.

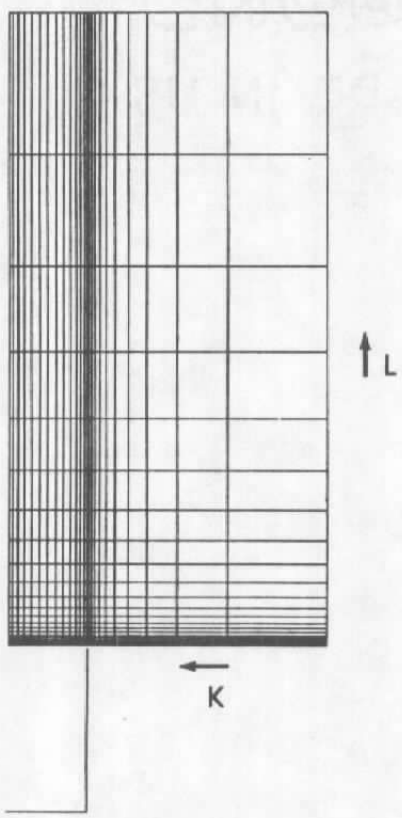


b. Side view

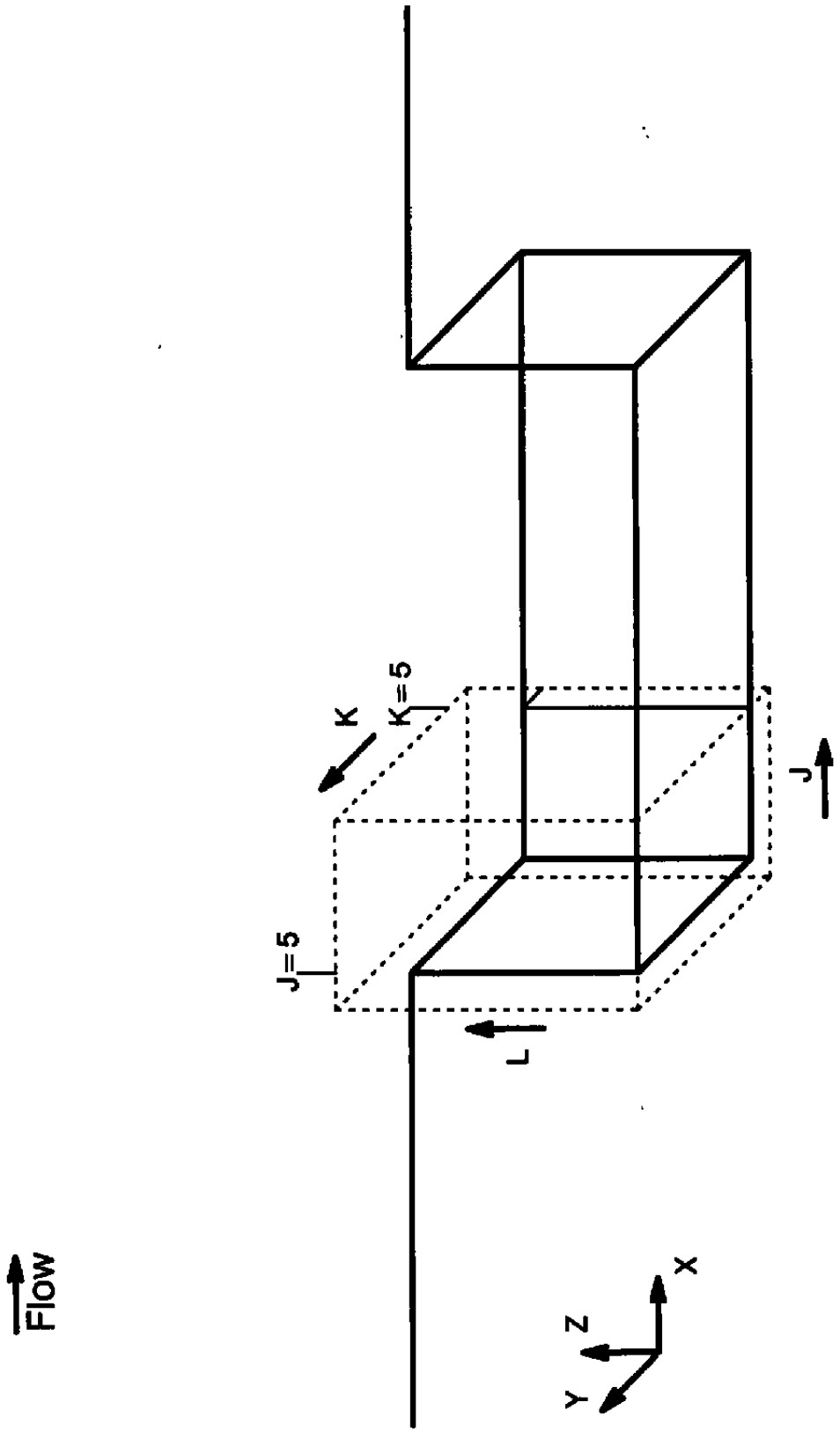
Figure G-3. Continued.



c. Top view
Figure G-3. Continued.

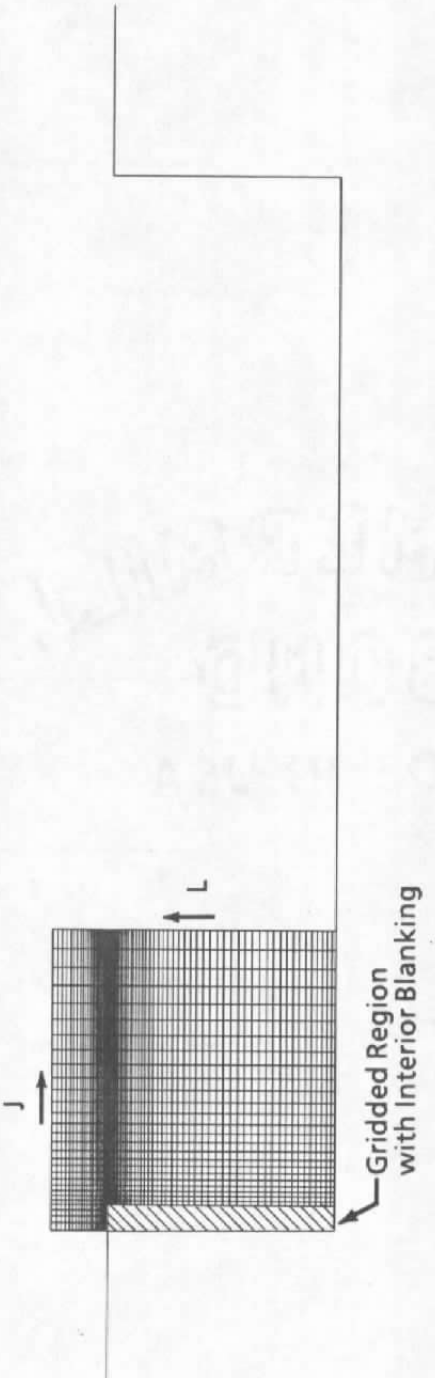


d. Front view
Figure G-3. Concluded.



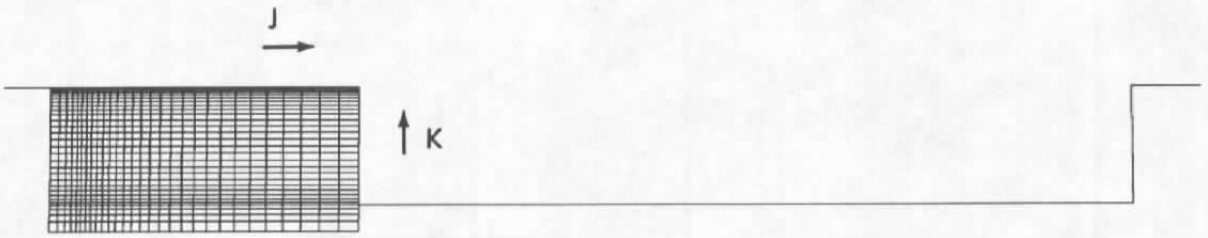
a. Graphical depiction

Figure G-4. 'CAVITY A' mesh.

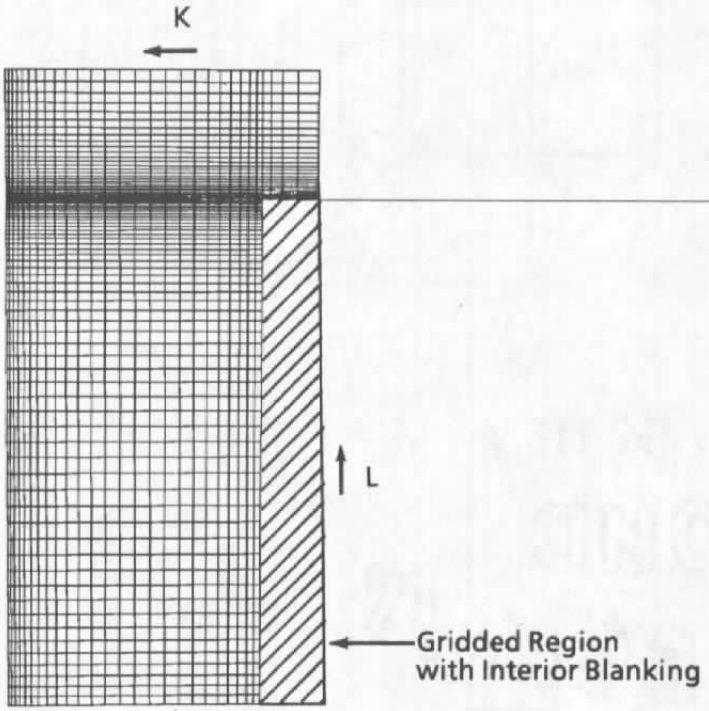


b. Side view

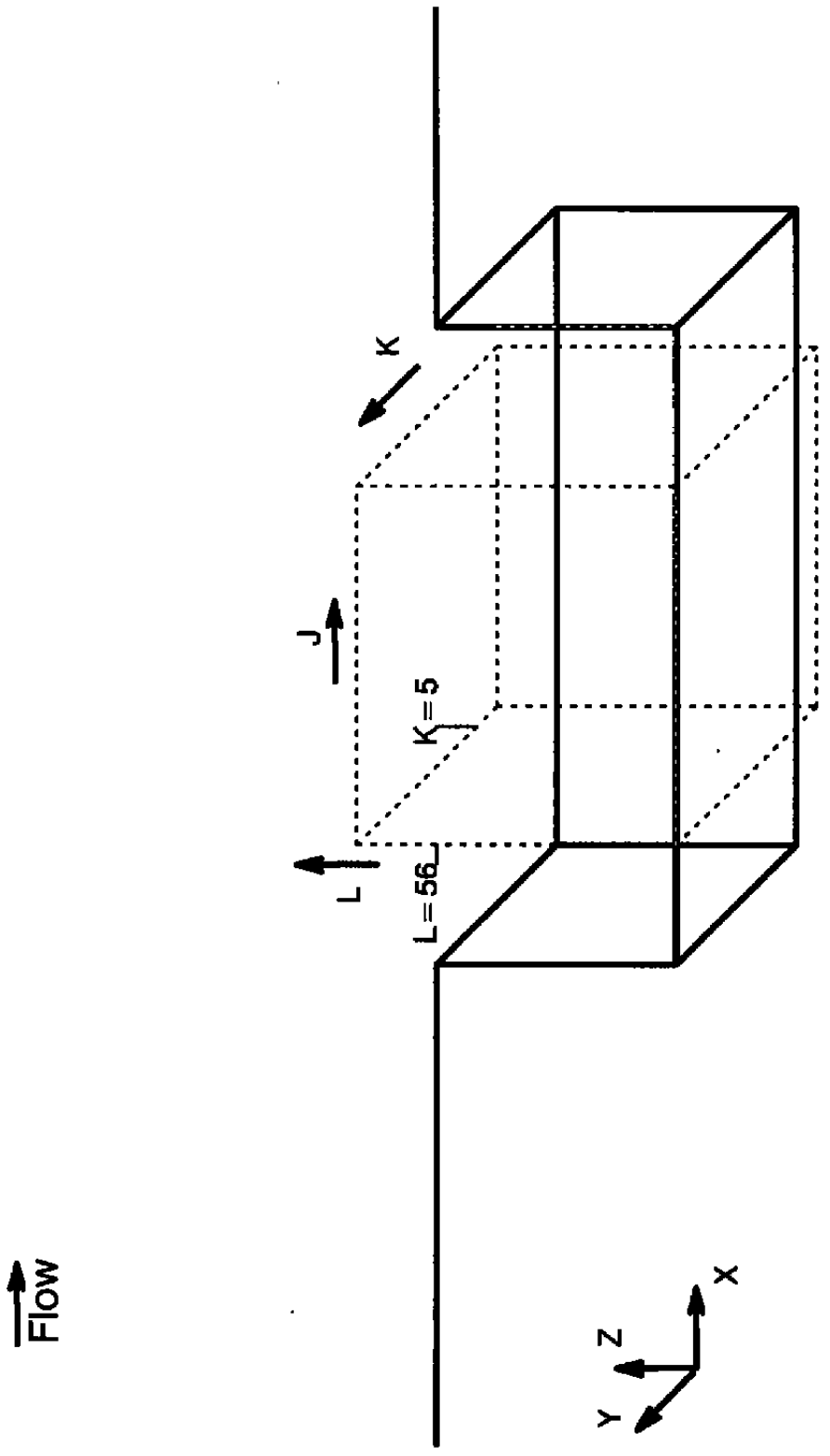
Figure G-4. Continued.



c. Top view
Figure G-4. Continued.

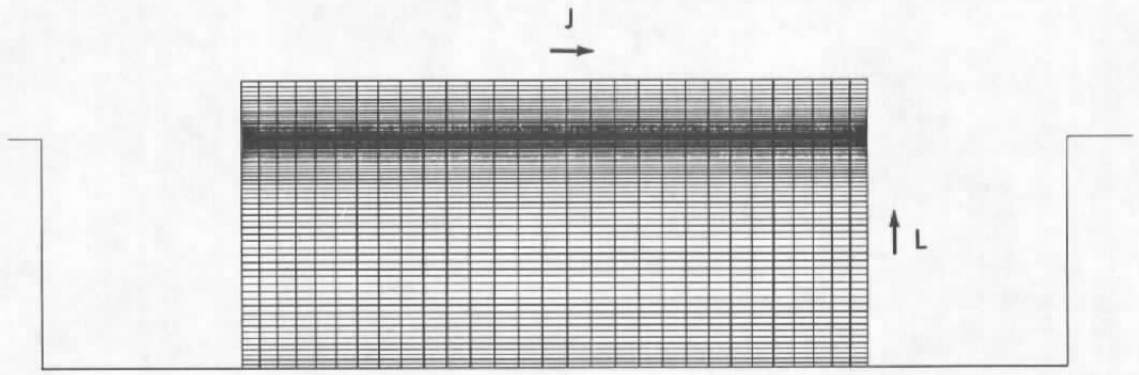


d. Front view
Figure G-4. Concluded.

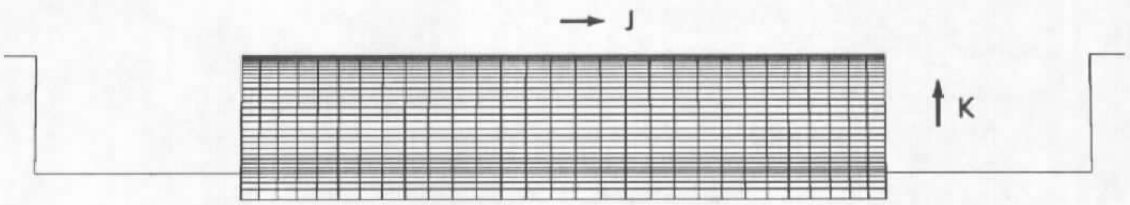


a. Graphical depiction

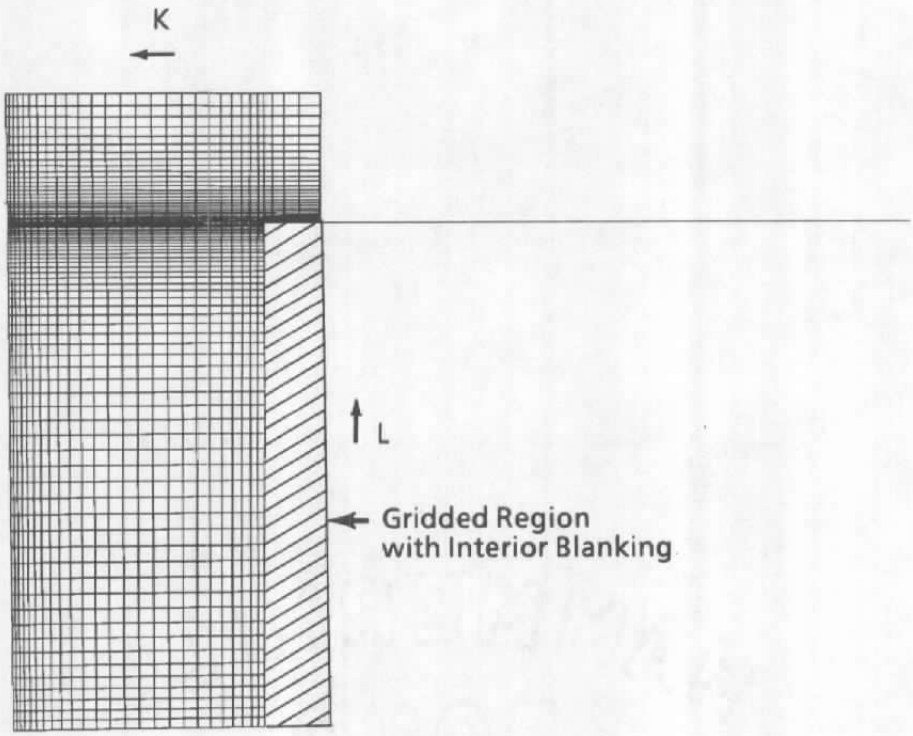
Figure G-5. 'CAVITY B' mesh.



b. Side view
Figure G-5. Continued.

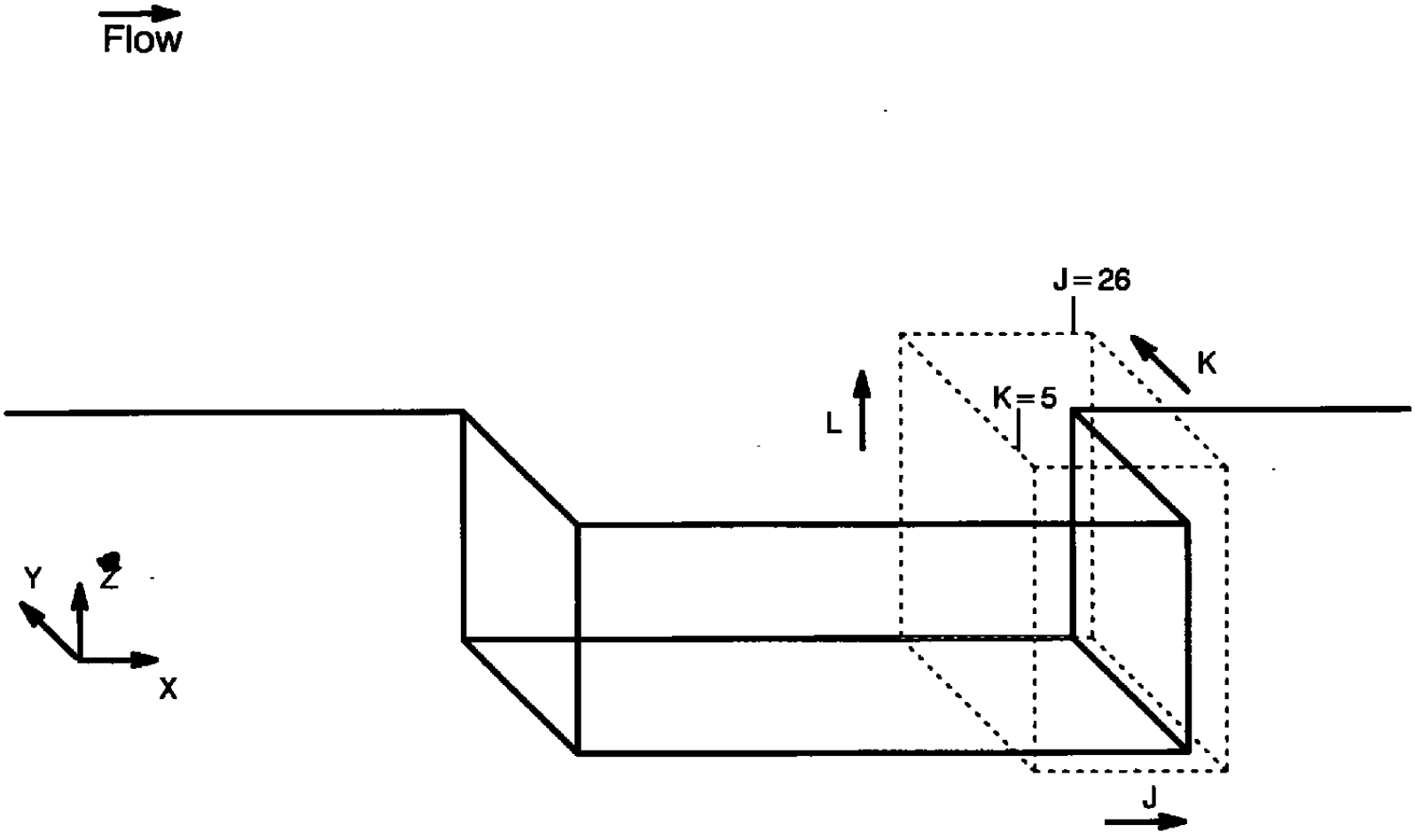


c. Top view
Figure G-5. Continued.



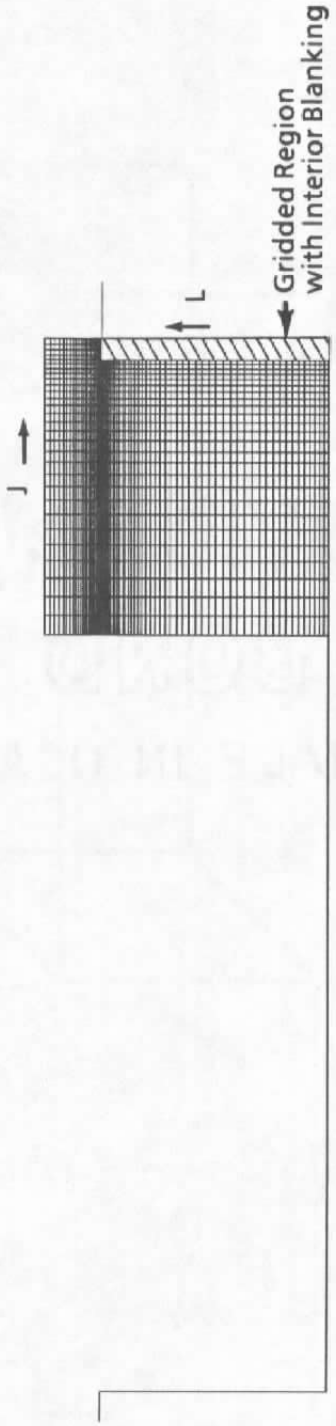
d. Front view

Figure G-5. Concluded.



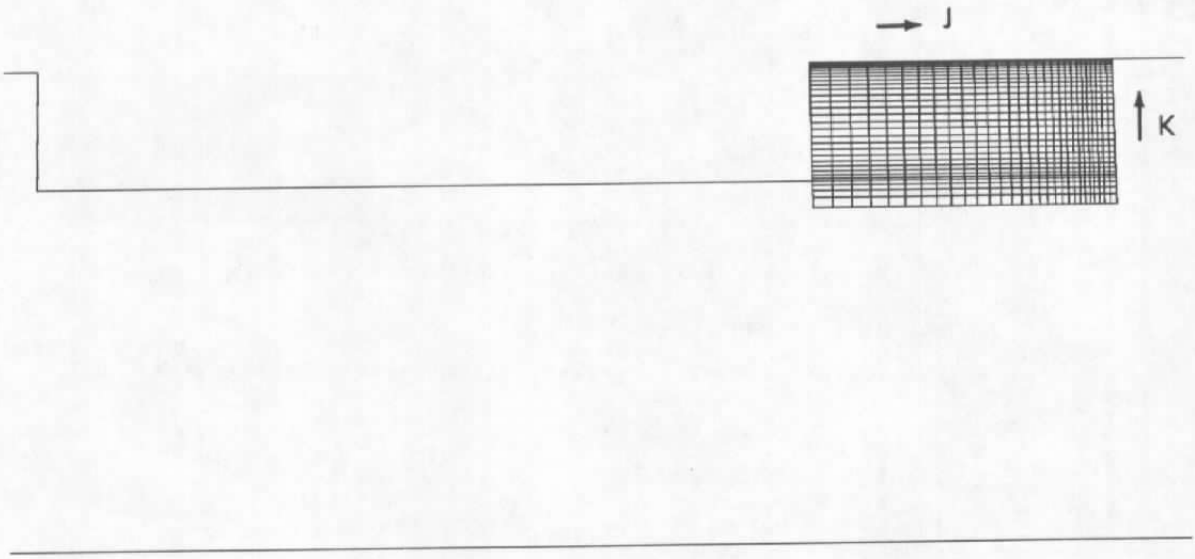
a. Graphical depiction

Figure G-6. 'CAVITY C' mesh.

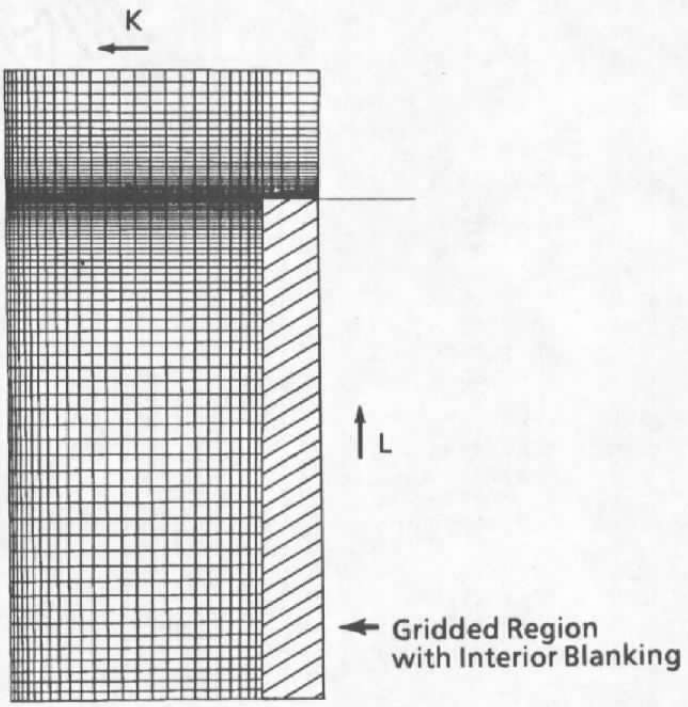


b. Side view

Figure G-6. Continued.



c. Top view
Figure G-6. Continued.



d. Front view
Figure G-6. Concluded.

```
C*************************************************************
C
C                    PEGSUS 4.0 INPUT
C
C               EXAMPLE G-1   EMPTY CAVITY
C                     5 MESH CASE
C
C*************************************************************

C*************************************************************
C
C  GLOBAL DEFINITIONS
C
C*************************************************************

 $GLOBAL
      FRINGE = 2,
      QUALITY = 1.0,0.5,-0.5,
      EPS = 0.0005,
      XINCLUDE = -30.0,46.999,
      YINCLUDE = -30.0,0.0,
      ZINCLUDE = -10.0,15.999,
                                    $END

C*************************************************************
C
C  MESH DEFINITIONS
C
C*************************************************************

 $MESH NAME = 'EXTERIOR A',
       LINK = 'CAVITY C',
              'CAVITY B',
              'CAVITY A',
              'EXTERIOR B',
                                    $END
 $MESH NAME = 'EXTERIOR B',
       LINK = 'CAVITY A',
              'CAVITY B',
              'CAVITY C',
              'EXTERIOR A',
                                    $END
 $MESH NAME = 'CAVITY A',
       LINK = 'CAVITY B',
              'EXTERIOR A',
              'EXTERIOR B',
                                    $END
 $MESH NAME = 'CAVITY B',
       LINK = 'CAVITY A',
              'CAVITY C',
              'EXTERIOR A',
              'EXTERIOR B',
                                    $END
 $MESH NAME = 'CAVITY C',
       LINK = 'CAVITY B',
              'EXTERIOR B',
              'EXTERIOR A',
                                    $END
```

a. Page 1

Figure G-7. User input for empty cavity.

```
C*****************************************************************
C
C  INTERPOLATION BOUNDARY DEFINITIONS
C
C*****************************************************************

C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  EXTERIOR A
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C-----------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C-----------------------------------------------------------------

 $BOUNDARY  NAME = 'EXTERIOR A OUTER BOUNDARY',
            ISPARTOF = 'EXTERIOR A',
                                    $END

 $SURFACE ISPARTOF = 'EXTERIOR A OUTER BOUNDARY',
           JRANGE = 66,67,
           KRANGE = 2,25,
           LRANGE = 2,35,
                                    $END
C-----------------------------------------------------------------
C  HOLE REGION DEFINITION
C-----------------------------------------------------------------

 $REGION NAME = 'EXTERIOR A HOLE REGION',
           TYPE = 'HOLE',
           ISPARTOF = 'EXTERIOR A',
                                    $END

 $VOLUME ISPARTOF = 'EXTERIOR A HOLE REGION',
           JRANGE = 27,67,
           KRANGE = 12,26,
           LRANGE = 1,20,
                                    $END
```

b. Page 2

Figure G-7. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  EXTERIOR B
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C-------------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C-------------------------------------------------------------------

 $BOUNDARY  NAME = 'EXTERIOR B OUTER BOUNDARY',
            ISPARTOF = 'EXTERIOR B',
                                  $END

 $SURFACE ISPARTOF = 'EXTERIOR B OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 2,25,
           LRANGE = 2,35,
                                  $END
C-------------------------------------------------------------------
C  HOLE REGION DEFINITION
C-------------------------------------------------------------------

 $REGION NAME = 'EXTERIOR B HOLE REGION',
           TYPE = 'HOLE',
           ISPARTOF = 'EXTERIOR B',
                                  $END
 $VOLUME ISPARTOF = 'EXTERIOR B HOLE REGION',
           JRANGE = 1,34,
           KRANGE = 12,26,
           LRANGE = 1,20,
                                  $END
```

c. Page 3

Figure G-7. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CAVITY A
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C---------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C---------------------------------------------------------------

 $BOUNDARY  NAME = 'CAVITY A OUTER BOUNDARY',
            ISPARTOF = 'CAVITY A',
                                $END

 $SURFACE ISPARTOF = 'CAVITY A OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 1,30,
           LRANGE = 57,85,
                                $END

 $SURFACE ISPARTOF = 'CAVITY A OUTER BOUNDARY',
           JRANGE = 29,30,
           KRANGE = 5,30,
           LRANGE = 2,56,
                                $END

 $SURFACE ISPARTOF = 'CAVITY A OUTER BOUNDARY',
           JRANGE = 29,30,
           KRANGE = 1,30,
           LRANGE = 57,85,
                                $END

 $SURFACE ISPARTOF = 'CAVITY A OUTER BOUNDARY',
           JRANGE = 3,28,
           KRANGE = 1,2,
           LRANGE = 57,85,
                                $END

 $SURFACE ISPARTOF = 'CAVITY A OUTER BOUNDARY',
           JRANGE = 3,28,
           KRANGE = 3,30,
           LRANGE = 84,85,
                                $END

C---------------------------------------------------------------
C  INTERIOR REGION DEFINITION
C---------------------------------------------------------------

 $REGION NAME = 'CAVITY A INTR REGION',
           TYPE = 'INTR',
           ISPARTOF = 'CAVITY A',
                                $END
 $VOLUME ISPARTOF = 'CAVITY A INTR REGION',
           JRANGE = 1,5,
           KRANGE = 1,31,
           LRANGE = 1,56,
                                $END

 $VOLUME ISPARTOF = 'CAVITY A INTR REGION',
           JRANGE = 1,30,
           KRANGE = 1,5,
           LRANGE = 1,56,
                                $END
```

d. Page 4

Figure G-7. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CAVITY B
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C--------------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C--------------------------------------------------------------------

 $BOUNDARY  NAME = 'CAVITY B OUTER BOUNDARY',
            ISPARTOF = 'CAVITY B',
                                  $END


 $SURFACE ISPARTOF = 'CAVITY B OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 5,30,
           LRANGE = 2,56,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY B OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 1,30,
           LRANGE = 57,85,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY B OUTER BOUNDARY',
           JRANGE = 29,30,
           KRANGE = 5,30,
           LRANGE = 2,56,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY B OUTER BOUNDARY',
           JRANGE = 29,30,
           KRANGE = 1,30,
           LRANGE = 57,85,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY B OUTER BOUNDARY',
           JRANGE = 3,28,
           KRANGE = 1,2,
           LRANGE = 57,85,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY B OUTER BOUNDARY',
           JRANGE = 3,28,
           KRANGE = 3,30,
           LRANGE = 84,85,
                                  $END

C--------------------------------------------------------------------
C  INTERIOR REGION DEFINITION
C--------------------------------------------------------------------

 $REGION NAME = 'CAVITY B INTR REGION',
           TYPE = 'INTR',
           ISPARTOF = 'CAVITY B',
                                  $END
 $VOLUME ISPARTOF = 'CAVITY B INTR REGION',
           JRANGE = 1,30,
           KRANGE = 1,5,
           LRANGE = 1,56,
                                  $END
```

e. Page 5

Figure G-7. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CAVITY C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C---------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C---------------------------------------------------------------


 $BOUNDARY  NAME = 'CAVITY C OUTER BOUNDARY',
            ISPARTOF = 'CAVITY C',
                                  $END

 $SURFACE ISPARTOF = 'CAVITY C OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 5,30,
           LRANGE = 2,56,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY C OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 1,30,
           LRANGE = 57,85,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY C OUTER BOUNDARY',
           JRANGE = 29,30,
           KRANGE = 1,30,
           LRANGE = 57,85,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY C OUTER BOUNDARY',
           JRANGE = 3,28,
           KRANGE = 1,2,
           LRANGE = 57,85,
                                  $END

 $SURFACE ISPARTOF = 'CAVITY C OUTER BOUNDARY',
           JRANGE = 3,28,
           KRANGE = 3,30,
           LRANGE = 84,85,
                                  $END

C---------------------------------------------------------------
C  INTERIOR REGION DEFINITION
C---------------------------------------------------------------

 $REGION NAME = 'CAVITY C INTR REGION',
           TYPE = 'INTR',
           ISPARTOF = 'CAVITY C',
                                  $END

 $VOLUME ISPARTOF = 'CAVITY C INTR REGION',
           JRANGE = 26,30,
           KRANGE = 1,31,
           LRANGE = 1,56,
                                  $END

 $VOLUME ISPARTOF = 'CAVITY C INTR REGION',
           JRANGE = 1,30,
           KRANGE = 1,5,
           LRANGE = 1,56,
                                  $END

C  END OF EMPTY CAVITY CASE
```

f. Page 6

Figure G-7. Concluded.

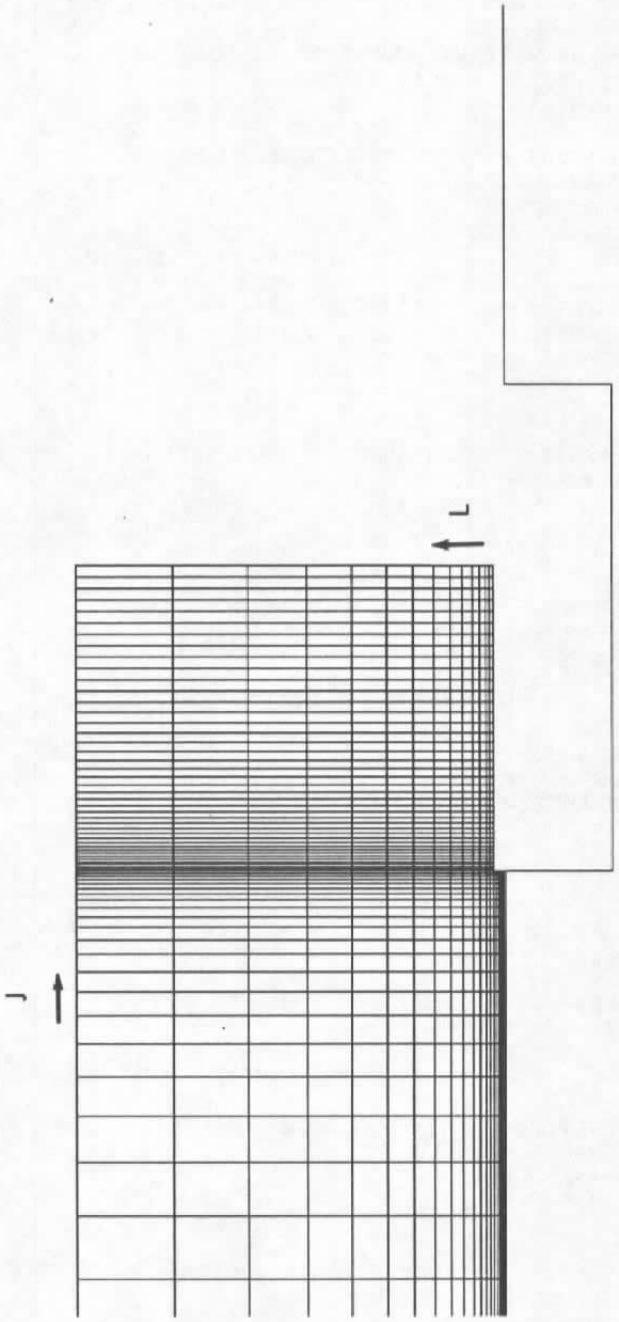


Figure G-8. Hole creation region in 'EXTERIOR A' mesh.

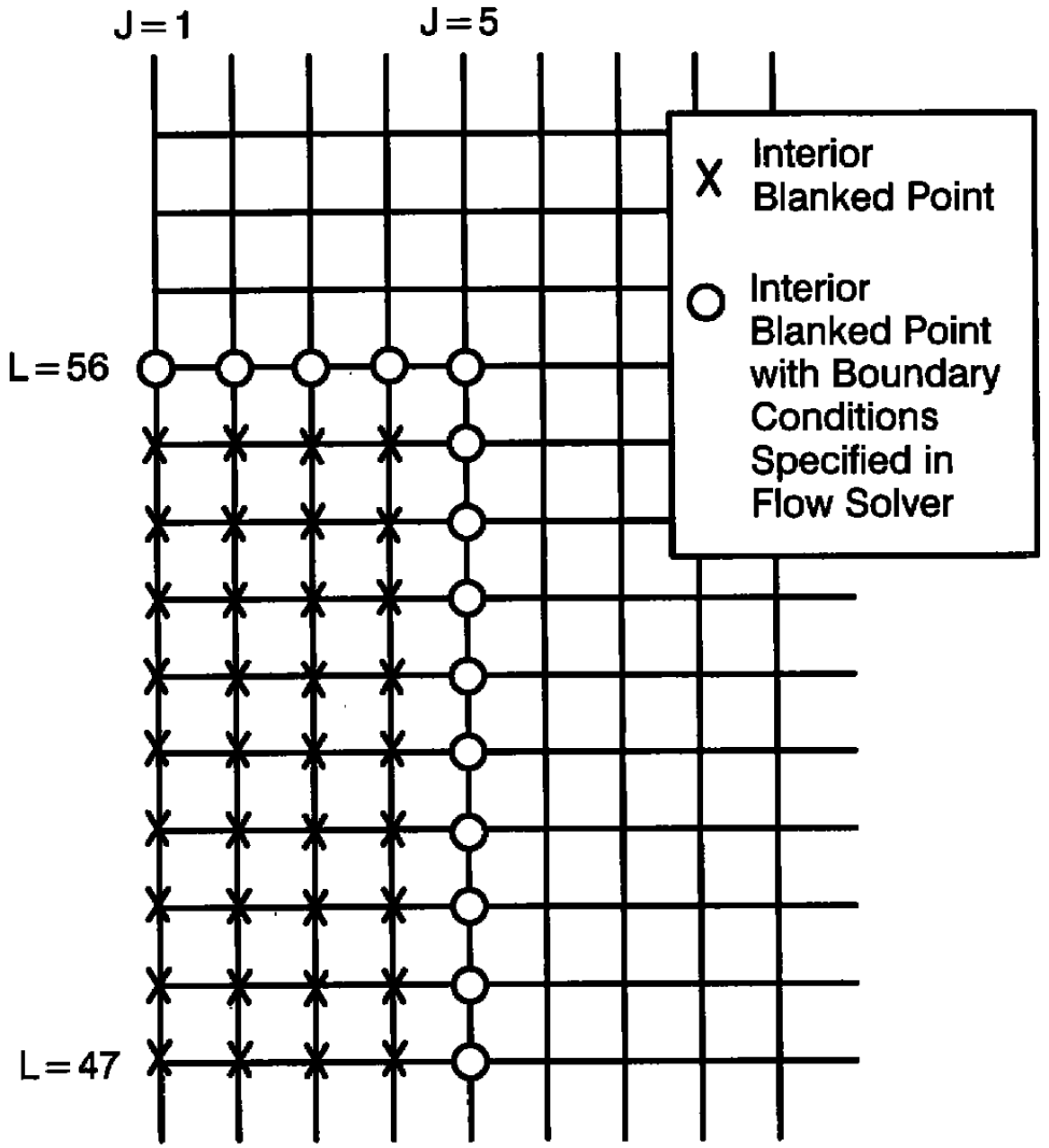


Figure G-9. Interior blanking of corner.

```
                    DIAGNOSTIC MAPS
            LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  EXTERIOR A
 B  -  EXTERIOR B
 C  -  CAVITY A
 D  -  CAVITY B
 E  -  CAVITY C


 INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  EXTERIOR A
 b  -  EXTERIOR B
 c  -  CAVITY A
 d  -  CAVITY B
 e  -  CAVITY C


 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT
```

a. Page 1

Figure G-10. Sample diagnostic maps for empty cavity.

```
MAP FOR EXTERIOR A

        PLANE  L =  10

  J         1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
 K
 1 ..........................................................................
 2 ..........................................................................bb.BB
 3 ..........................................................................bb.BB
 4 ..........................................................................bb.BB
 5 ..........................................................................bb.BB
 6 .....................*....................................................bb.BB
 7 .....................cccc.*.*.*.*.*.*.*.*.*.******.*.*.*.*.*.*d**.BB
 8 .....................*cccc.*.*.*.*.*.*.*.*.*.******.*.*.*.*.*.*d**.BB
 9 .....................*.....................................................DD
10 .....................*....cccccccccccccccccccccccDDDDDDDDDDDDDDDDDDDDD
11 .....................*...ccccccccccccccccccccccccDDDDDDDDDDDDDDDDDDDD
12 .....................*..cc##############################################
13 .....................*..cc##############################################
14 .....................*..cc##############################################
15 .....................*..cc##############################################
16 .....................*..cc##############################################
17 .....................*..cc##############################################
18 .....................*..cc##############################################
19 .....................*..cc##############################################
20 .....................*..cc##############################################
21 .....................*..cc##############################################
22 .....................*..cc##############################################
23 .....................*..cc##############################################
24 .....................*..cc##############################################
25 ......................cc###############################################
26 ......................################################################
```

b. Page 2

Figure G-10. Continued.

```
MAP FOR EXTERIOR B

        PLANE  L =  10

  J        1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ..........................................................................
  2 AA.aa.....................................................................
  3 AA.aa.....................................................................
  4 AA.aa.....................................................................
  5 AA.aa.....................................................................
  6 AA.aa.....................................................................
  7 AAd**dddd****eeeeeeeeeeeeeeeeeeee.*.*.....................................
  8 AAd**dddd****eeeeeeeeeeeeeeeeeeee.*.*.....................................
  9 DD.........................................*.....................................
 10 DDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEE...*.....................................
 11 DDDDDDDDDDDDEEEEEEEEEEEEEEEEEEEEE..*.....................................
 12 ########################################EE.*.....................................
 13 ########################################EE.*.....................................
 14 ########################################EE.*.....................................
 15 ########################################EE.*.....................................
 16 ########################################EE.*.....................................
 17 ########################################EE.*.....................................
 18 ########################################EE.*.....................................
 19 ########################################EE.*.....................................
 20 ########################################EE.*.....................................
 21 ########################################EE.*.....................................
 22 ########################################EE.*.....................................
 23 ########################################EE.*.....................................
 24 ########################################EE.*.....................................
 25 ########################################EE.......................................
 26 ########################################.........................................
```

c. Page 3

Figure G-10. Continued.

```
MAP FOR CAVITY A

          PLANE  L =  24

   J         1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ################################
  2 ################################
  3 ################################
  4 ################################
  5 ############################dd###
  6 #####.....................dd.DD
  7 #####.....................dd.DD
  8 #####.....................dd.DD
  9 #####.....................dd.DD
 10 #####.....................dd.DD
 11 #####.....................dd.DD
 12 #####.....................dd.DD
 13 #####.....................dd.DD
 14 #####.....................dd.DD
 15 #####.....................dd.DD
 16 #####.....................dd.DD
 17 #####.....................dd.DD
 18 #####.....................dd.DD
 19 #####.....................dd.DD
 20 #####.....................dd.DD
 21 #####.....................dd.DD
 22 #####.....................dd.DD
 23 #####.....................dd.DD
 24 #####.....................dd.DD
 25 #####.....................dd.DD
 26 #####.....................dd.DD
 27 #####.....................dd.DD
 28 #####.....................dd.DD
 29 #####.....................dd.DD
 30 #####.....................dd.DD
 31 #####.........................
```

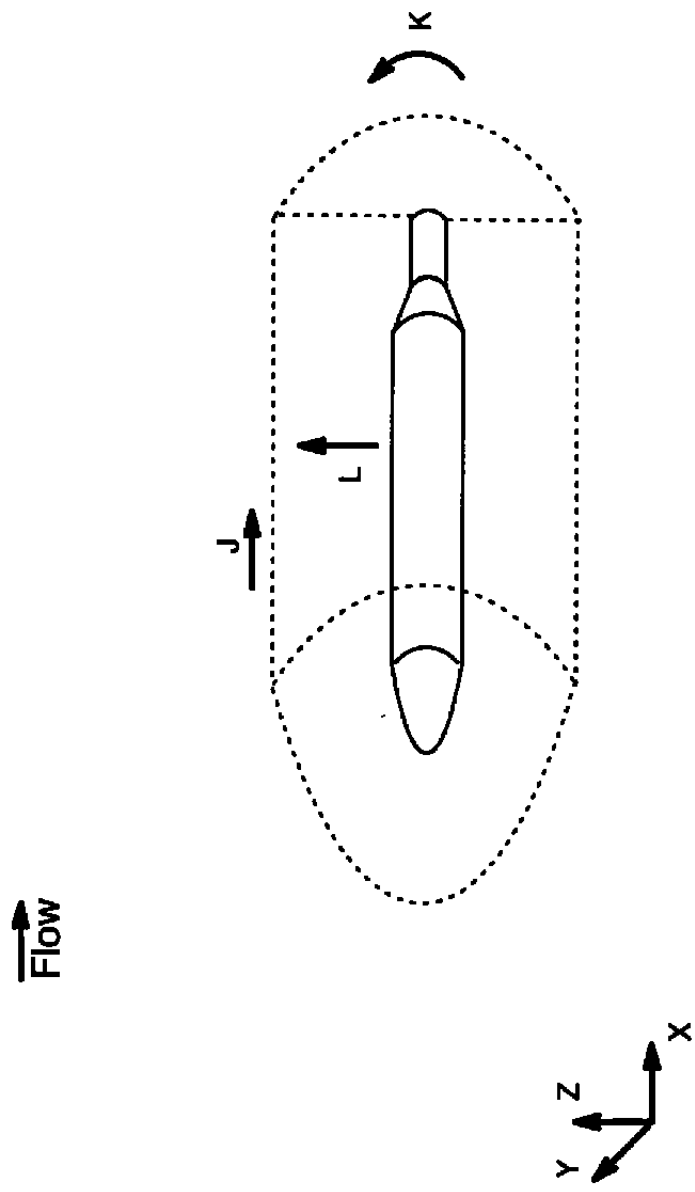
d. Page 4
Figure G-10. Continued.

```
MAP FOR CAVITY B

          PLANE  L =  24

    J         1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ################################
  2 ################################
  3 ################################
  4 ################################
  5 ###cc########################ee###
  6 CC.cc.....................ee.EE
  7 CC.cc.....................ee.EE
  8 CC.cc.....................ee.EE
  9 CC.cc.....................ee.EE
 10 CC.cc.....................ee.EE
 11 CC.cc.....................ee.EE
 12 CC.cc.....................ee.EE
 13 CC.cc.....................ee.EE
 14 CC.cc.....................ee.EE
 15 CC.cc.....................ee.EE
 16 CC.cc.....................ee.EE
 17 CC.cc.....................ee.EE
 18 CC.cc.....................ee.EE
 19 CC.cc.....................ee.EE
 20 CC.cc.....................ee.EE
 21 CC.cc.....................ee.EE
 22 CC.cc.....................ee.EE
 23 CC.cc.....................ee.EE
 24 CC.cc.....................ee.EE
 25 CC.cc.....................ee.EE
 26 CC.cc.....................ee.EE
 27 CC.cc.....................ee.EE
 28 CC.cc.....................ee.EE
 29 CC.cc.....................ee.EE
 30 CC.cc.....................ee.EE
 31 ............................
```

e. Page 5
Figure G-10. Continued.

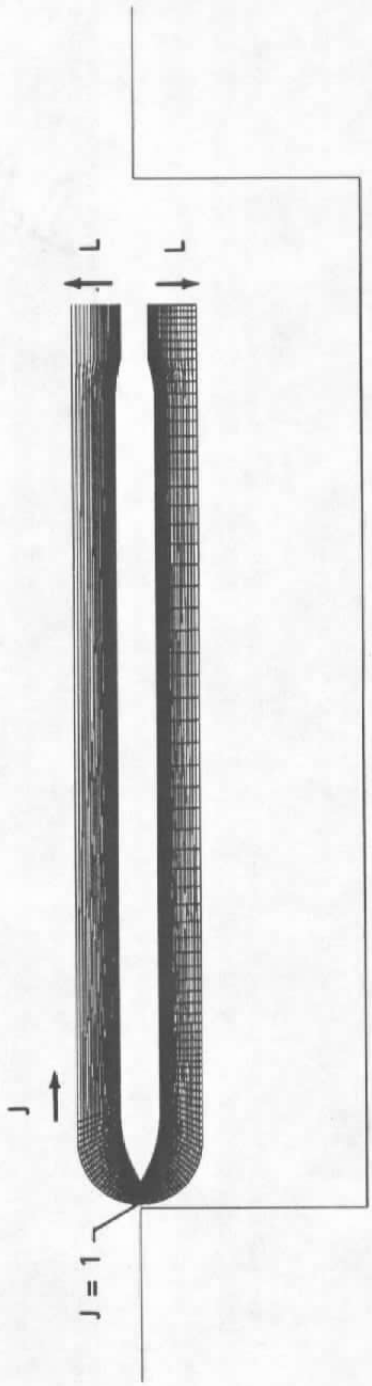
```
MAP FOR CAVITY C

          PLANE  L =  24

    J         1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ##############################
  2 ##############################
  3 ##############################
  4 ##############################
  5 ###dd##########################
  6 DD.dd..................######
  7 DD.dd..................######
  8 DD.dd..................######
  9 DD.dd..................######
 10 DD.dd..................######
 11 DD.dd..................######
 12 DD.dd..................######
 13 DD.dd..................######
 14 DD.dd..................######
 15 DD.dd..................######
 16 DD.dd..................######
 17 DD.dd..................######
 18 DD.dd..................######
 19 DD.dd..................######
 20 DD.dd..................######
 21 DD.dd..................######
 22 DD.dd..................######
 23 DD.dd..................######
 24 DD.dd..................######
 25 DD.dd..................######
 26 DD.dd..................######
 27 DD.dd..................######
 28 DD.dd..................######
 29 DD.dd..................######
 30 DD.dd..................######
 31 ........................######
```

f. Page 6
Figure G-10. Concluded.

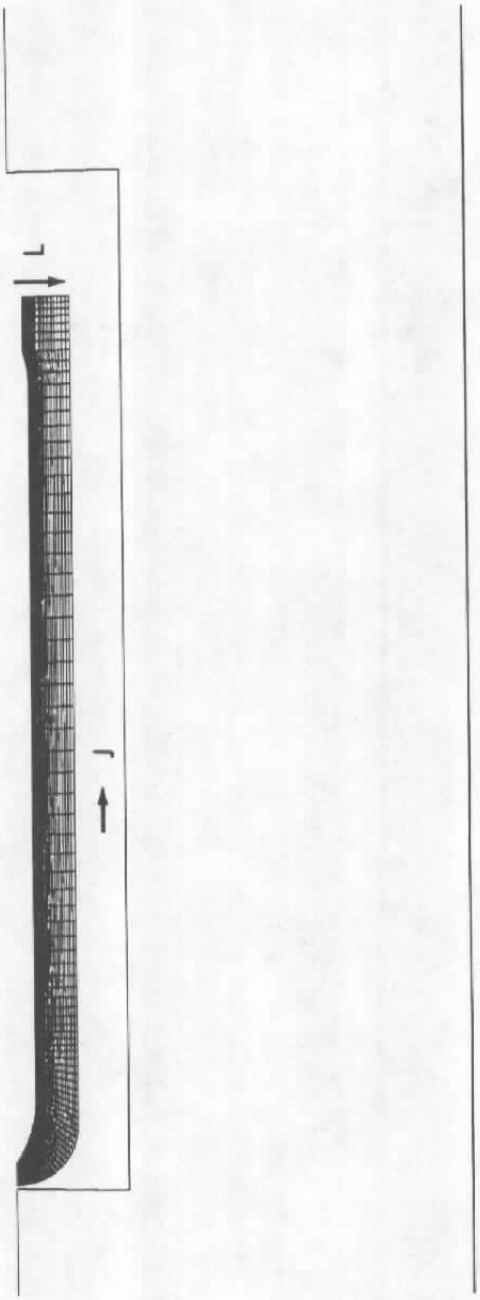


a. Graphical depiction

Figure G-11. 'STORE' mesh.

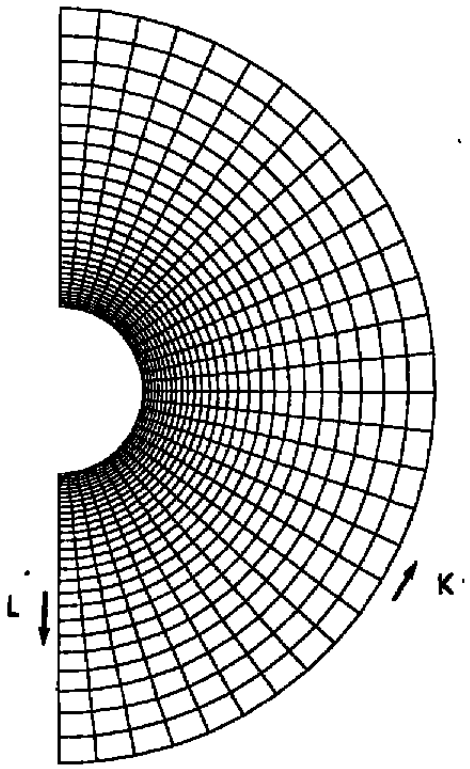


b. Side view
Figure G-11. Continued.

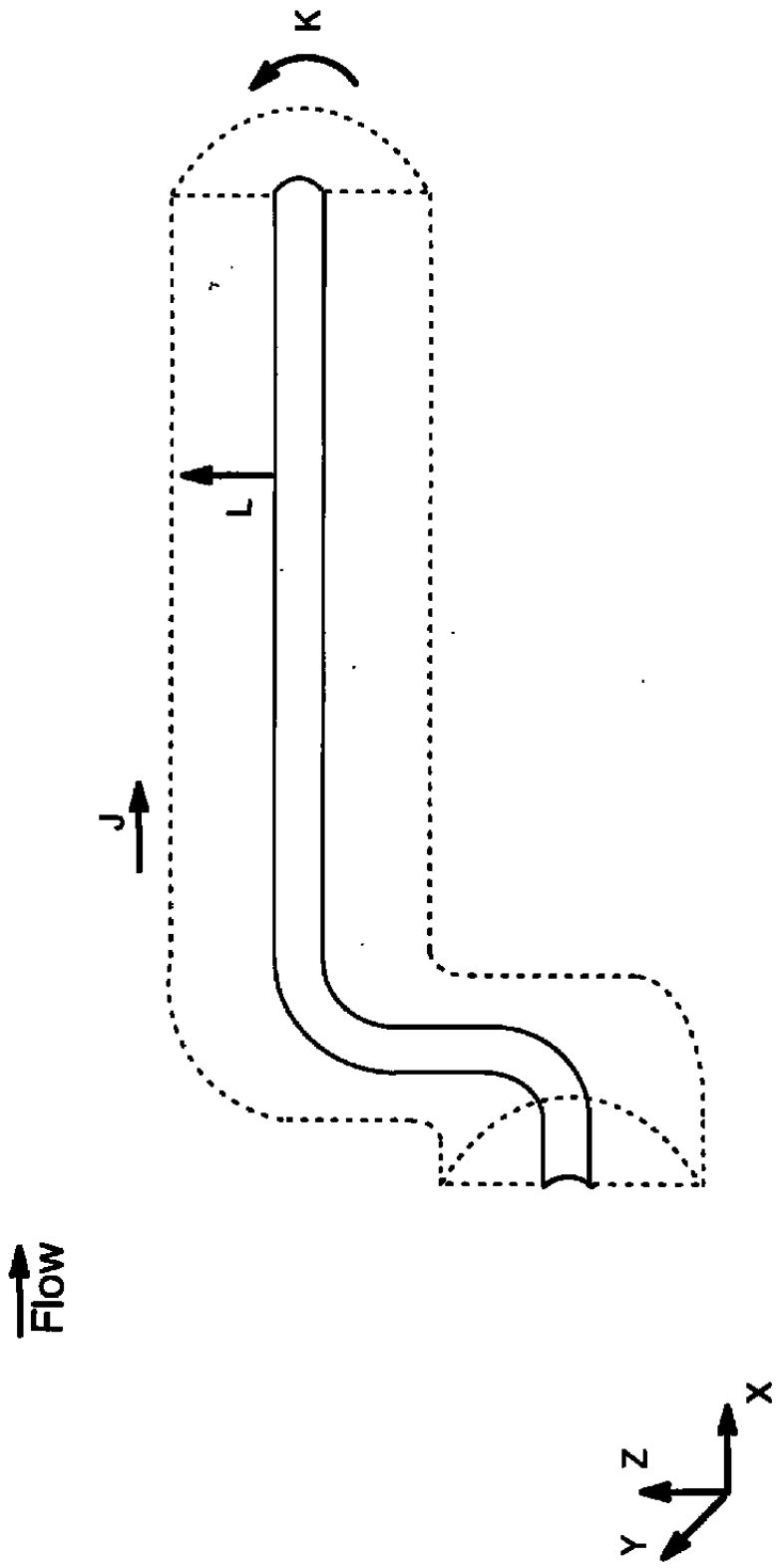


c. Top view

Figure G-11. Continued.

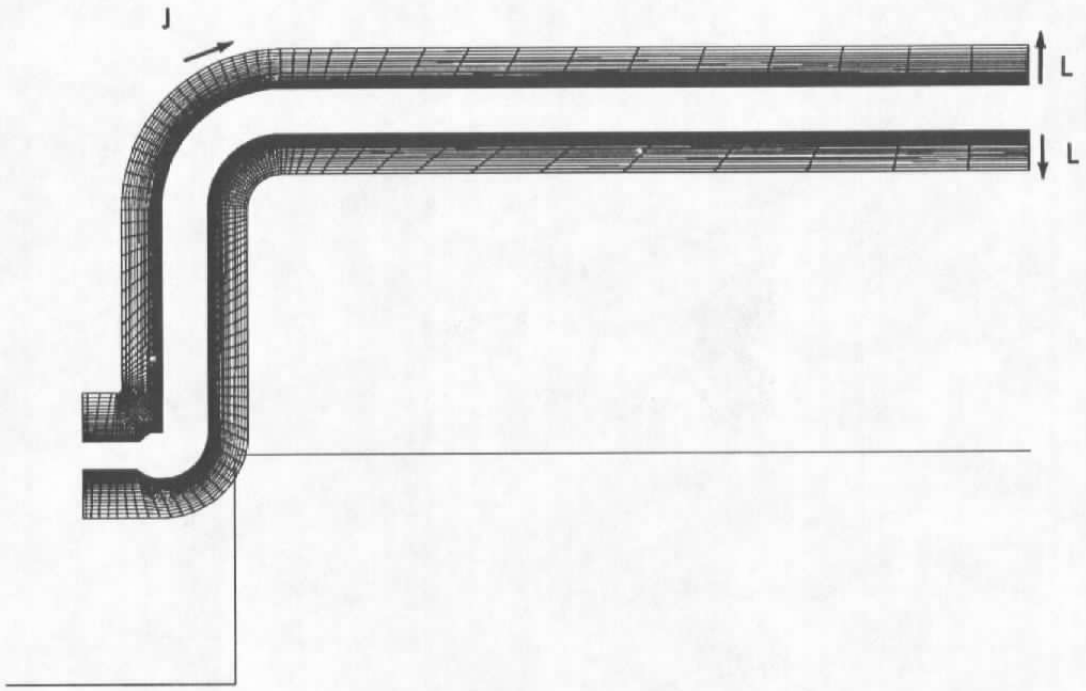


d. Front view
Figure G-11. Concluded.

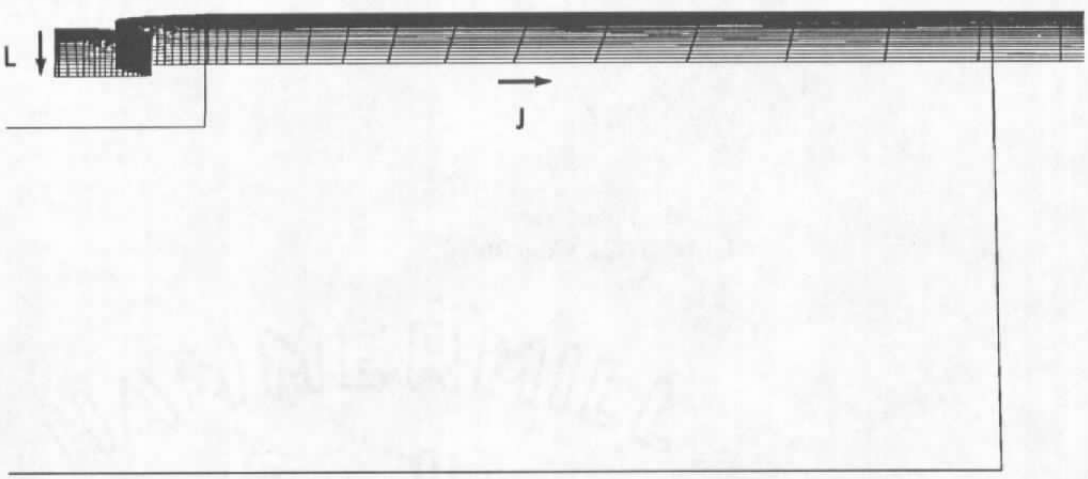


a. Graphical depiction

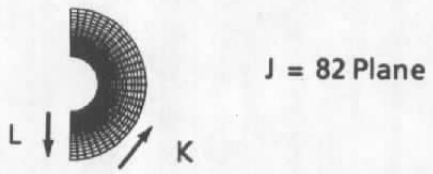
Figure G-12. 'STING' mesh.

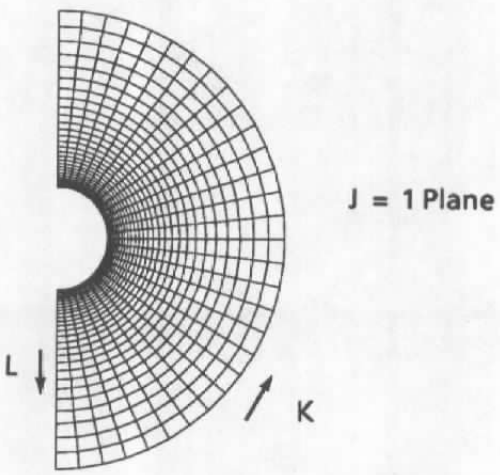


b. Side view
Figure G-12. Continued.

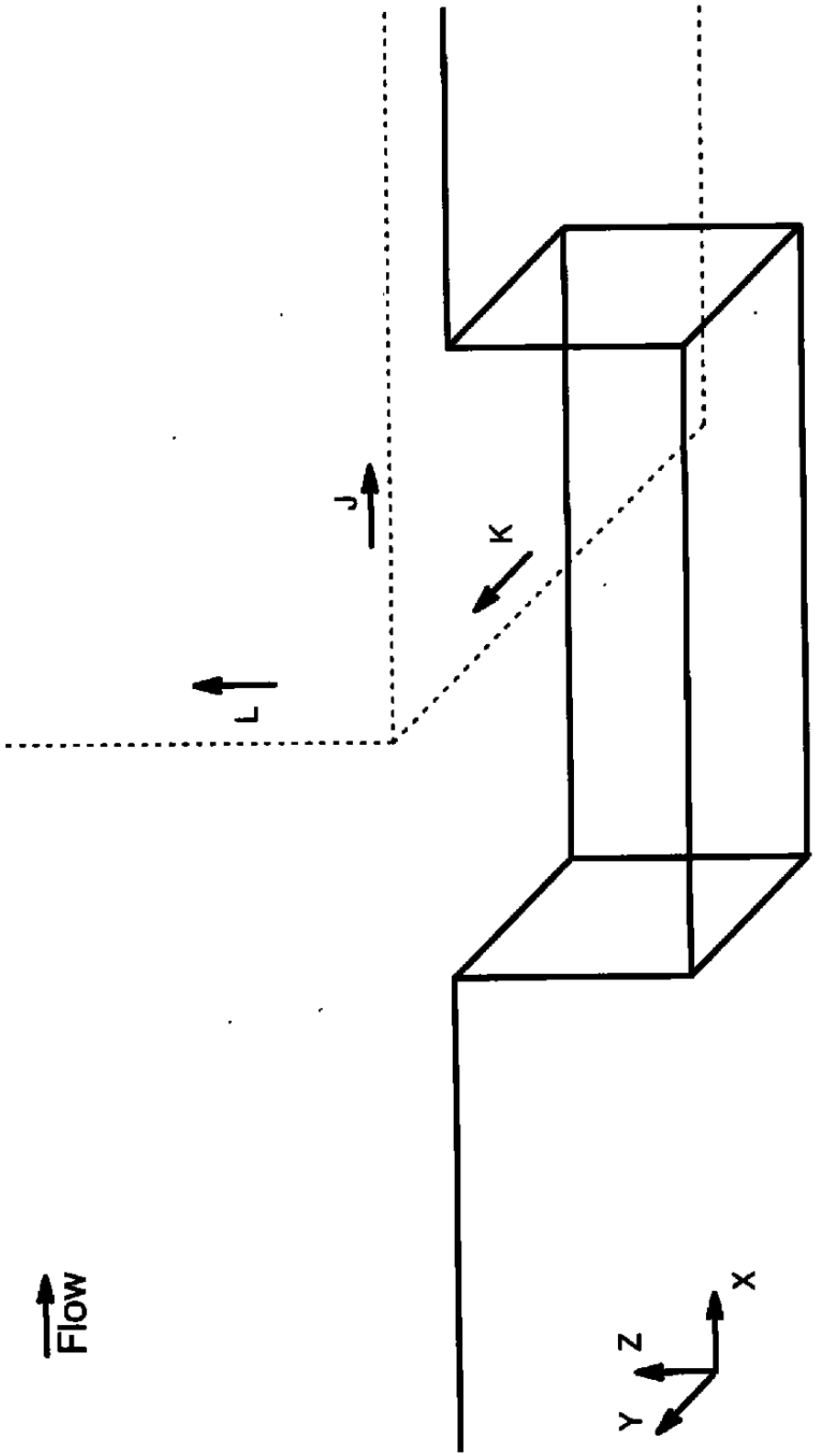


c. Top view
Figure G-12. Continued.



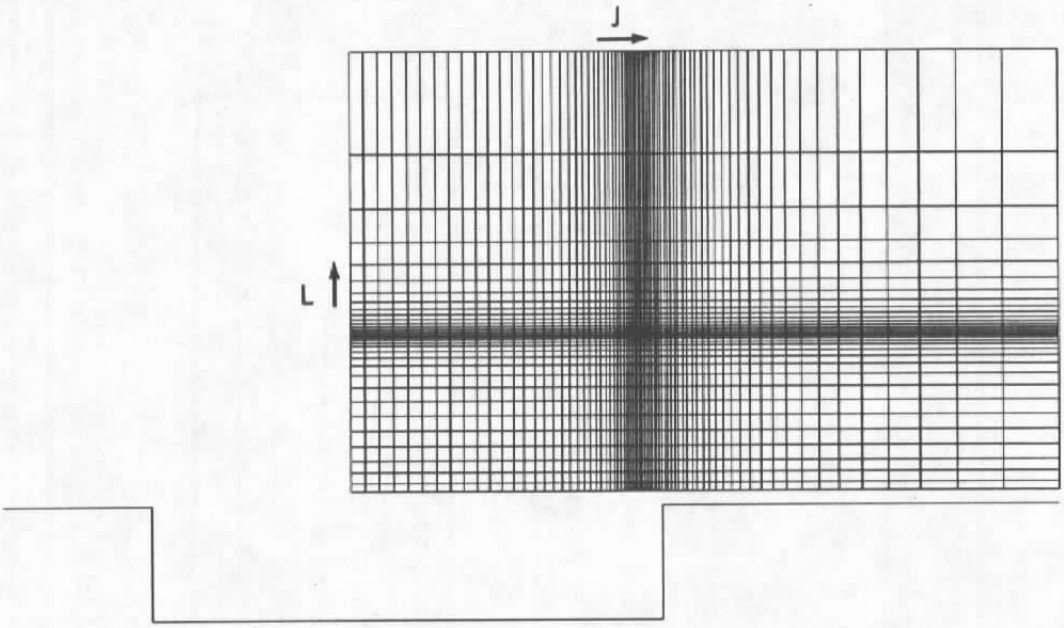


d. Front view
Figure G-12. Concluded.

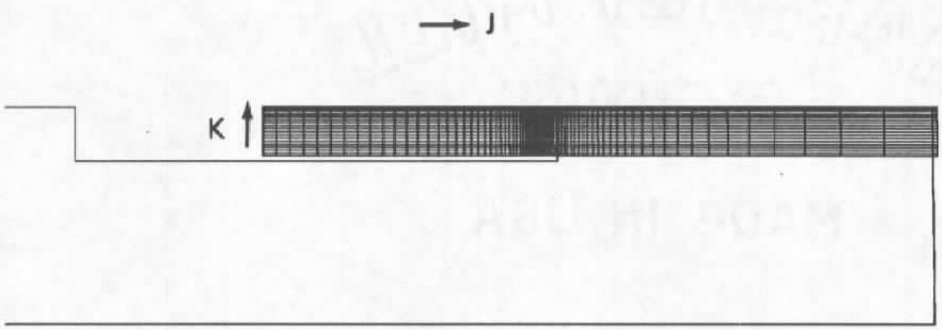


a. Graphical depiction

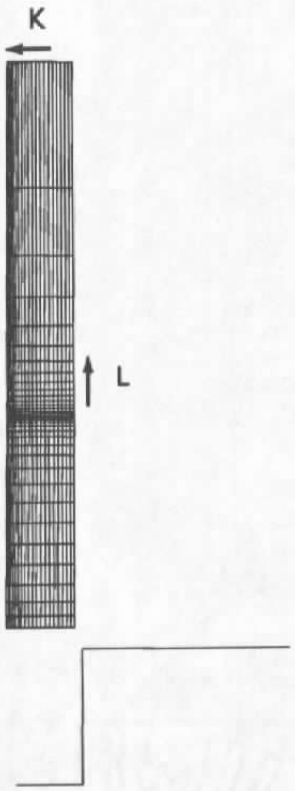
Figure G-13. 'INTERFACE' mesh.



b. Side view
Figure G-13. Continued.



c. Top view
Figure G-13. Continued.



d. Front view
Figure G-13. Concluded.

```
*************************************************************
C
C                    PEGSUS 4.0 INPUT
C
C       EXAMPLE G-2    EMPTY CAVITY WITH STORE/STING
C                      8 MESH CASE
C       STORE, STING, AND INTERFACE MESHES ADDED TO 5 MESH
C       EMPTY CAVITY
C
C*************************************************************

C*************************************************************
C
C  MESH DEFINITIONS
C
C*************************************************************

 $MESH NAME = 'STORE',
         MODE = 'ADD',
         LINK = 'CAVITY A',
                'STING',
                'CAVITY B',
                'CAVITY C',
                'EXTERIOR A',
                'EXTERIOR B',
                'INTERFACE',
         ADDLINK = 'EXTERIOR A',
                   'EXTERIOR B',
                   'CAVITY A',
                   'CAVITY B',
                   'CAVITY C',
       X0 = 15.468, Y0 = 0.0, Z0 = 0.0,
                                  $END

 $MESH NAME = 'STING',
       MODE = 'ADD',
       LINK = 'CAVITY C',
              'EXTERIOR B',
              'INTERFACE',
              'STORE',
              'CAVITY B',
              'EXTERIOR A',
       ADDLINK = 'EXTERIOR A',
                 'EXTERIOR B',
                 'CAVITY B',
                 'CAVITY C',
       YINCLUDE = -30.0,30.0,
       KINCLUDE = 2,32,
       X0 = 15.468, Y0 = 0.0, Z0 = 0.0,
                                  $END
 $MESH NAME = 'INTERFACE',
       MODE = 'ADD',
         LINK = 'EXTERIOR B',
                'STING',
                'CAVITY C',
                'EXTERIOR A',
                'STORE',
                'CAVITY B',
       ADDLINK = 'EXTERIOR A',
                 'EXTERIOR B',
                 'CAVITY B',
                 'CAVITY C',
                                  $END
```

a. Page 1

Figure G-14. User input for store/sting in a cavity.

```
C*************************************************************
C
C  INTERPOLATION BOUNDARY DEFINITIONS
C
C*************************************************************


C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  STORE
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C-------------------------------------------------------------
C  HOLE CREATION BOUNDARY DEFINITION
C-------------------------------------------------------------

 $BOUNDARY  NAME = 'STORE HOLE BOUNDARY',
            MHOLEIN = 'CAVITY A',
                      'CAVITY B',
                      'CAVITY C',
                      'EXTERIOR A',
                      'EXTERIOR B',
            ISPARTOF = 'STORE',
                                  $END

 $SURFACE ISPARTOF = 'STORE HOLE BOUNDARY',
            JRANGE = 1,82,
            KRANGE = 2,32,
            LRANGE = 5,5,
            NVOUT  = '+L',
                                  $END
 $SURFACE ISPARTOF = 'STORE HOLE BOUNDARY',
            JRANGE = 82,82,
            KRANGE = 2,32,
            LRANGE = 1,5,
            NVOUT  = '+J',
                                  $END

C-------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C-------------------------------------------------------------

 $BOUNDARY  NAME = 'STORE OUTER BOUNDARY',
            ISPARTOF = 'STORE',
                                  $END

 $SURFACE ISPARTOF = 'STORE OUTER BOUNDARY',
            JRANGE = 1,82,
            KRANGE = 2,32,
            LRANGE = 28,29,
                                  $END
 $SURFACE ISPARTOF = 'STORE OUTER BOUNDARY',
            JRANGE = 81,82,
            KRANGE = 2,32,
            LRANGE = 1,28,
                                  $END
```

b. Page 2

Figure G-14. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  STING
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C--------------------------------------------------------------------
C  HOLE CREATION BOUNDARY DEFINITION
C--------------------------------------------------------------------

 $BOUNDARY  NAME = 'STING HOLE BOUNDARY',
            MHOLEIN = 'EXTERIOR A',
                      'EXTERIOR B',
                      'CAVITY B',
                      'CAVITY C',
                      'INTERFACE',
            ISPARTOF = 'STING',
                                    $END

 $SURFACE ISPARTOF = 'STING HOLE BOUNDARY',
            JRANGE = 1,82,
            KRANGE = 2,32,
            LRANGE = 5,5,
            NVOUT  = '+L',
                                    $END

 $SURFACE ISPARTOF = 'STING HOLE BOUNDARY',
            JRANGE = 1,1,
            KRANGE = 2,32,
            LRANGE = 1,5,
            NVOUT  = '-J',
                                    $END

 $SURFACE ISPARTOF = 'STING HOLE BOUNDARY',
            JRANGE = 82,82,
            KRANGE = 2,32,
            LRANGE = 1,5,
            NVOUT  = '+J',
                                    $END

C--------------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C--------------------------------------------------------------------

 $BOUNDARY  NAME = 'STING OUTER BOUNDARY',
            ISPARTOF = 'STING',
                                    $END

 $SURFACE ISPARTOF = 'STING OUTER BOUNDARY',
            JRANGE = 1,81,
            KRANGE = 2,32,
            LRANGE = 28,29,
                                    $END
 $SURFACE ISPARTOF = 'STING OUTER BOUNDARY',
            JRANGE = 1,2,
            KRANGE = 2,32,
            LRANGE = 1,28,
                                    $END
```

c. Page 3

Figure G-14. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  INTERFACE
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C-----------------------------------------------------------------
C  HOLE CREATION BOUNDARY DEFINITION
C-----------------------------------------------------------------

 $BOUNDARY  NAME = 'INTERFACE HOLE BOUNDARY',
            MHOLEIN = 'EXTERIOR A',
                      'EXTERIOR B',
            ISPARTOF = 'INTERFACE',
                                  $END

 $SURFACE ISPARTOF = 'INTERFACE HOLE BOUNDARY',
           JRANGE = 10,10,
           KRANGE = 7,21,
           LRANGE = 6,35,
           NVOUT  = '-J',
                                  $END

 $SURFACE ISPARTOF = 'INTERFACE HOLE BOUNDARY',
           JRANGE = 10,77,
           KRANGE = 7,7,
           LRANGE = 6,35,
           NVOUT  = '-K',
                                  $END

 $SURFACE ISPARTOF = 'INTERFACE HOLE BOUNDARY',
           JRANGE = 10,77,
           KRANGE = 7,21,
           LRANGE = 6,6,
           NVOUT  = '-L',
                                  $END

C-----------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C-----------------------------------------------------------------

 $BOUNDARY  NAME = 'INTERFACE OUTER BOUNDARY',
            ISPARTOF = 'INTERFACE',
                                  $END

 $SURFACE ISPARTOF = 'INTERFACE OUTER BOUNDARY',
           JRANGE = 1,2,
           KRANGE = 1,20,
           LRANGE = 1,35,
                                  $END

 $SURFACE ISPARTOF = 'INTERFACE OUTER BOUNDARY',
           JRANGE = 1,76,
           KRANGE = 1,20,
           LRANGE = 1,2,
                                  $END

 $SURFACE ISPARTOF = 'INTERFACE OUTER BOUNDARY',
           JRANGE = 1,76,
           KRANGE = 1,2,
           LRANGE = 1,35,
                                  $END
```

d. Page 4
Figure G-14. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CAVITY C - ADDED HOLE
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C--------------------------------------------------------------------
C  HOLE CREATION BOUNDARY DEFINITION
C--------------------------------------------------------------------


 $BOUNDARY  NAME = 'AFT CAVITY HOLE BOUNDARY',
            MODE = 'ADD',
            MHOLEIN = 'STING',
            ISPARTOF = 'CAVITY C',
                                 $END

 $SURFACE ISPARTOF = 'AFT CAVITY HOLE BOUNDARY',
           JRANGE = 26,26,
           KRANGE = 5,30,
           LRANGE = 1,56,
           NVOUT  = '-J',
                                 $END

 $SURFACE ISPARTOF = 'AFT CAVITY HOLE BOUNDARY',
           JRANGE = 26,30,
           KRANGE = 5,30,
           LRANGE = 56,56,
           NVOUT  = '+L',
                                 $END


C  END OF CAVITY WITH STORE/STING CASE
```

e. Page 5

Figure G-14. Concluded.

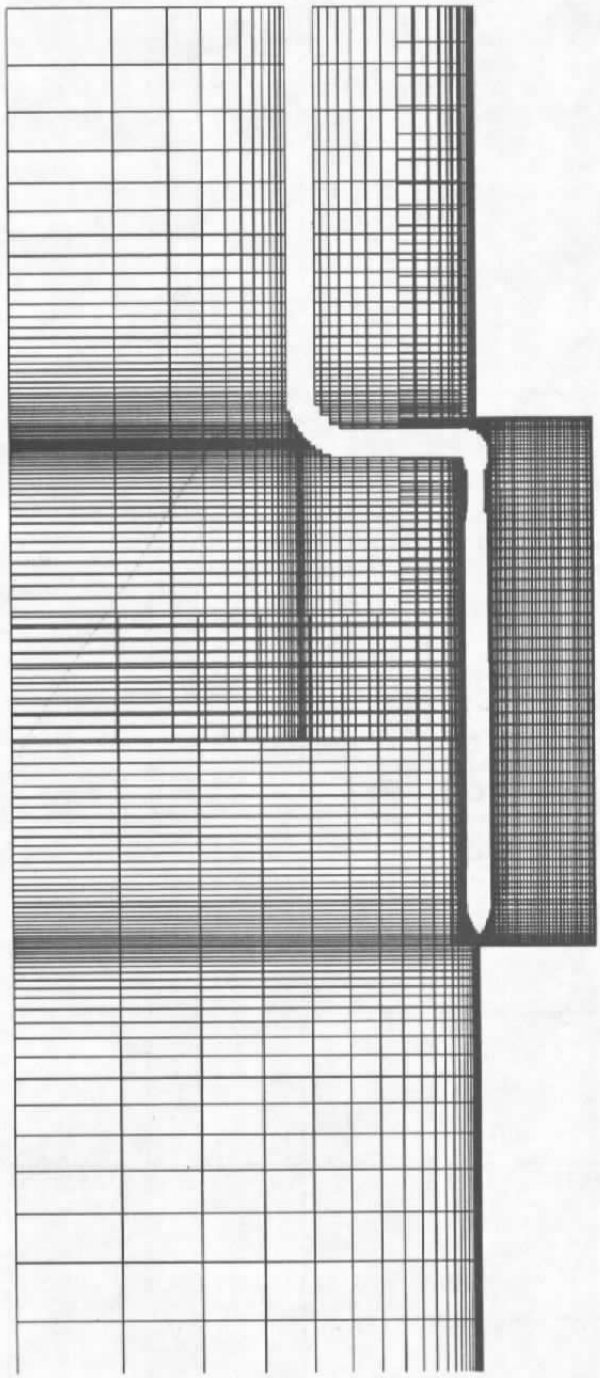
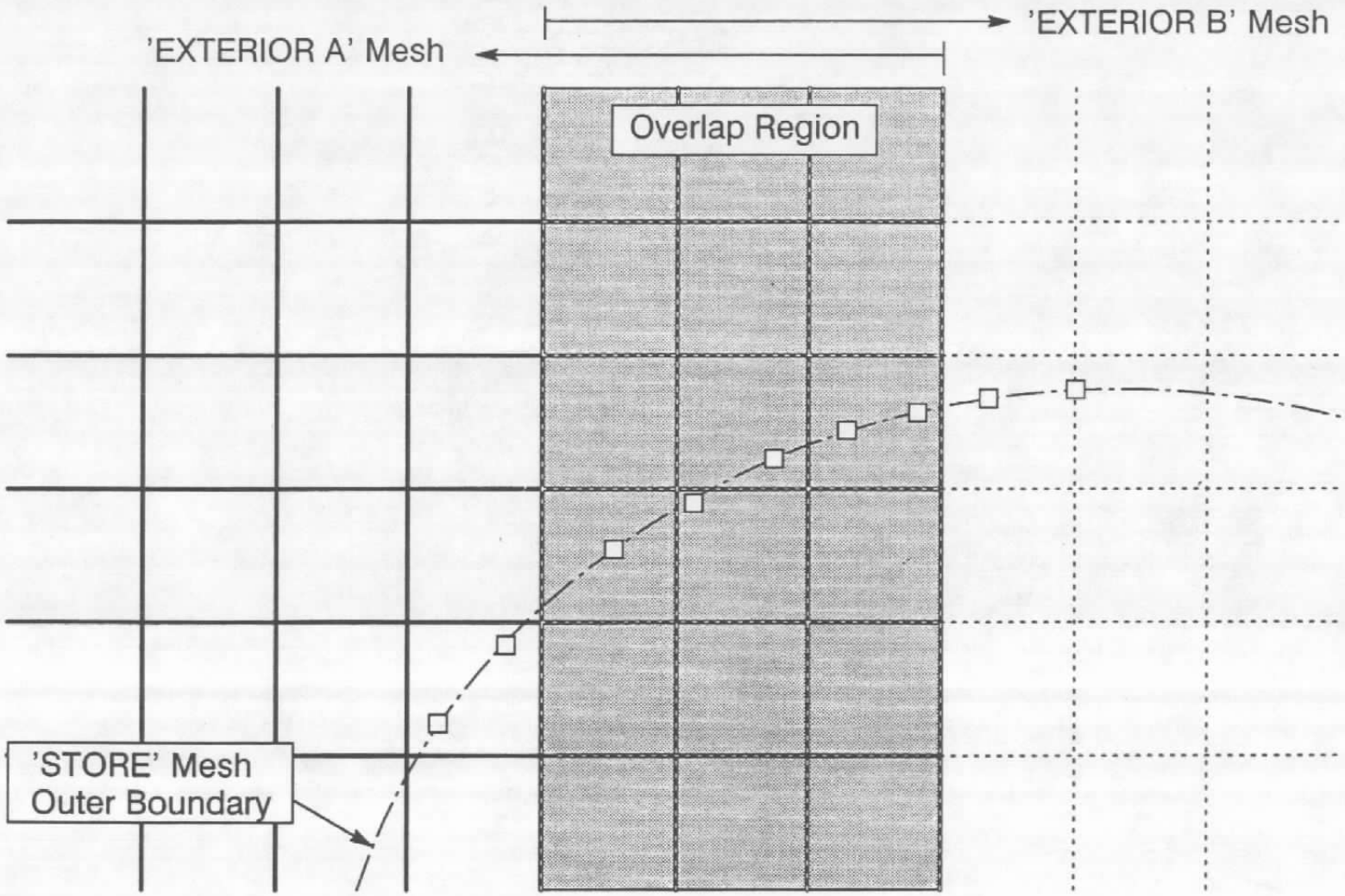
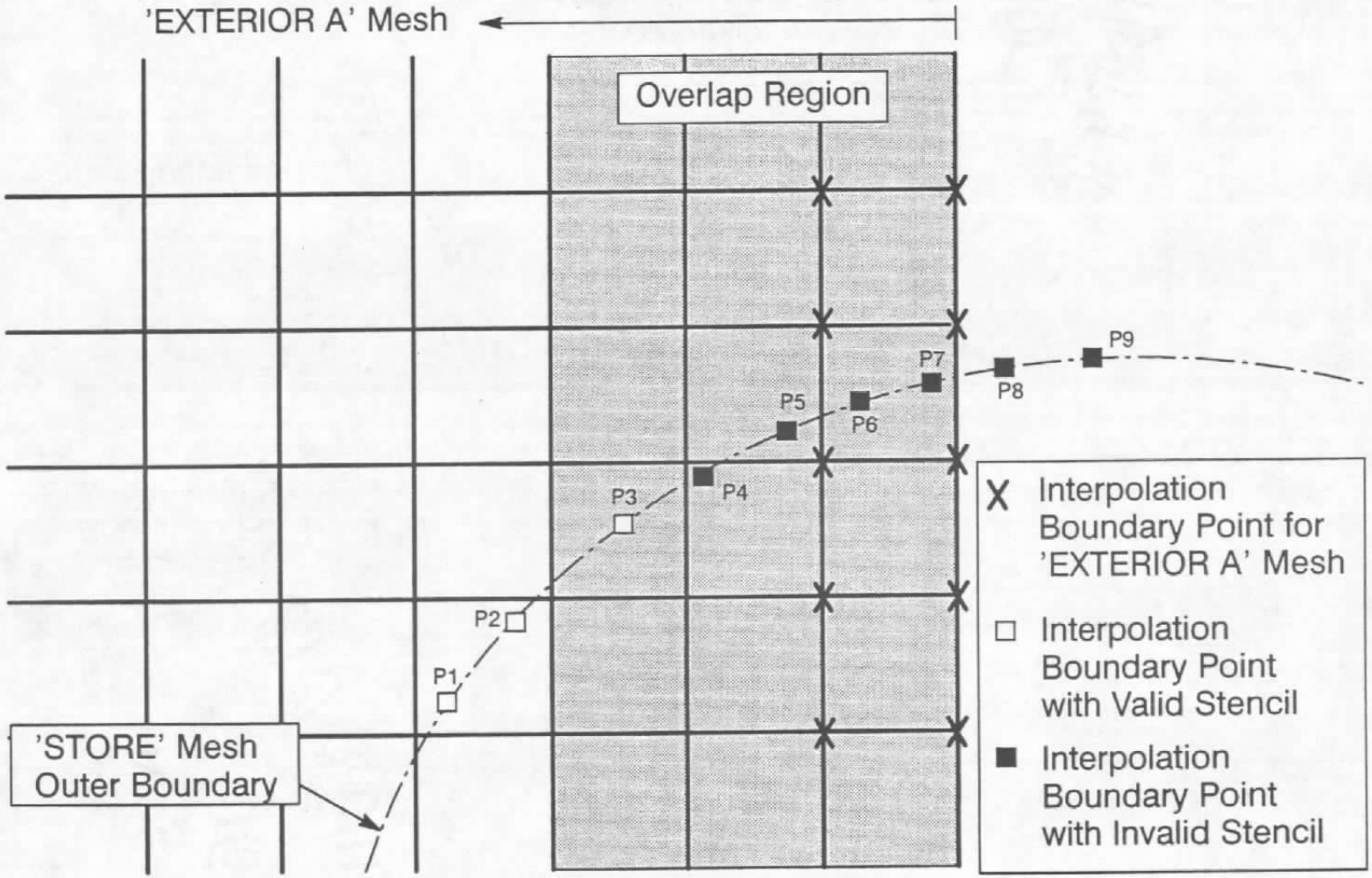


Figure G-15. Hole made by store/sting.

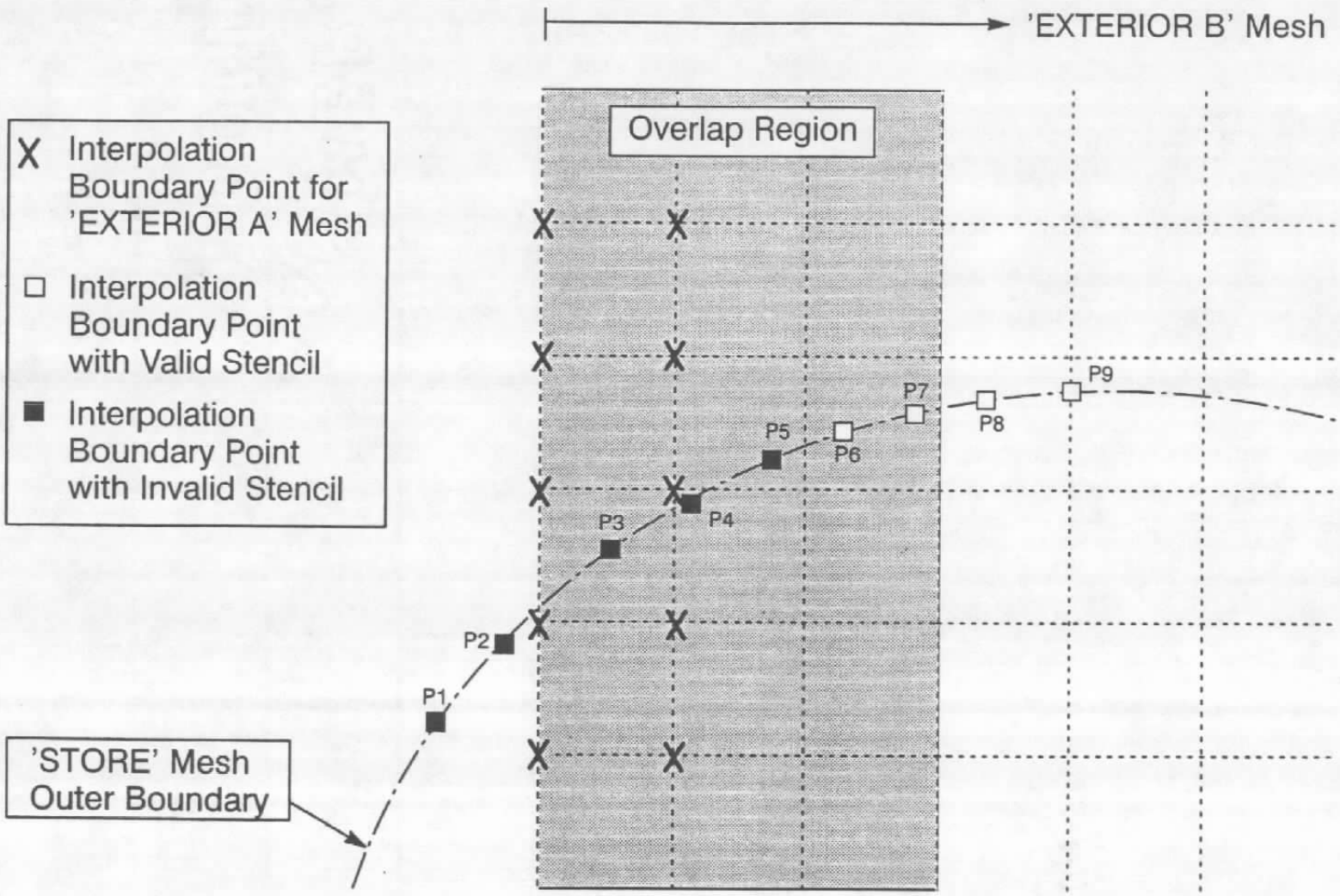


a. Overlap region

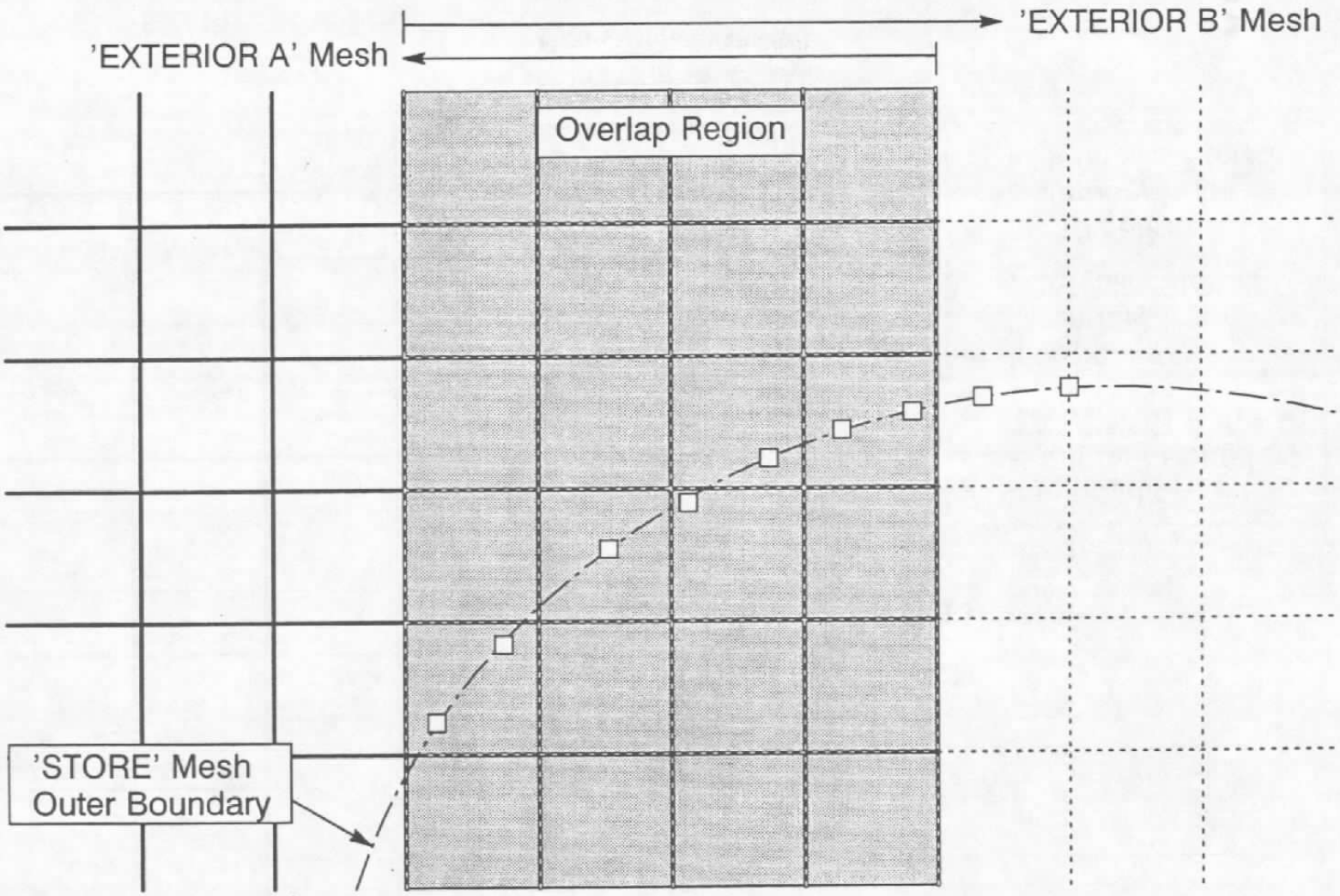
Figure G-16. Four-point overlap.



b. 'EXTERIOR A' mesh
Figure G-16. Continued.

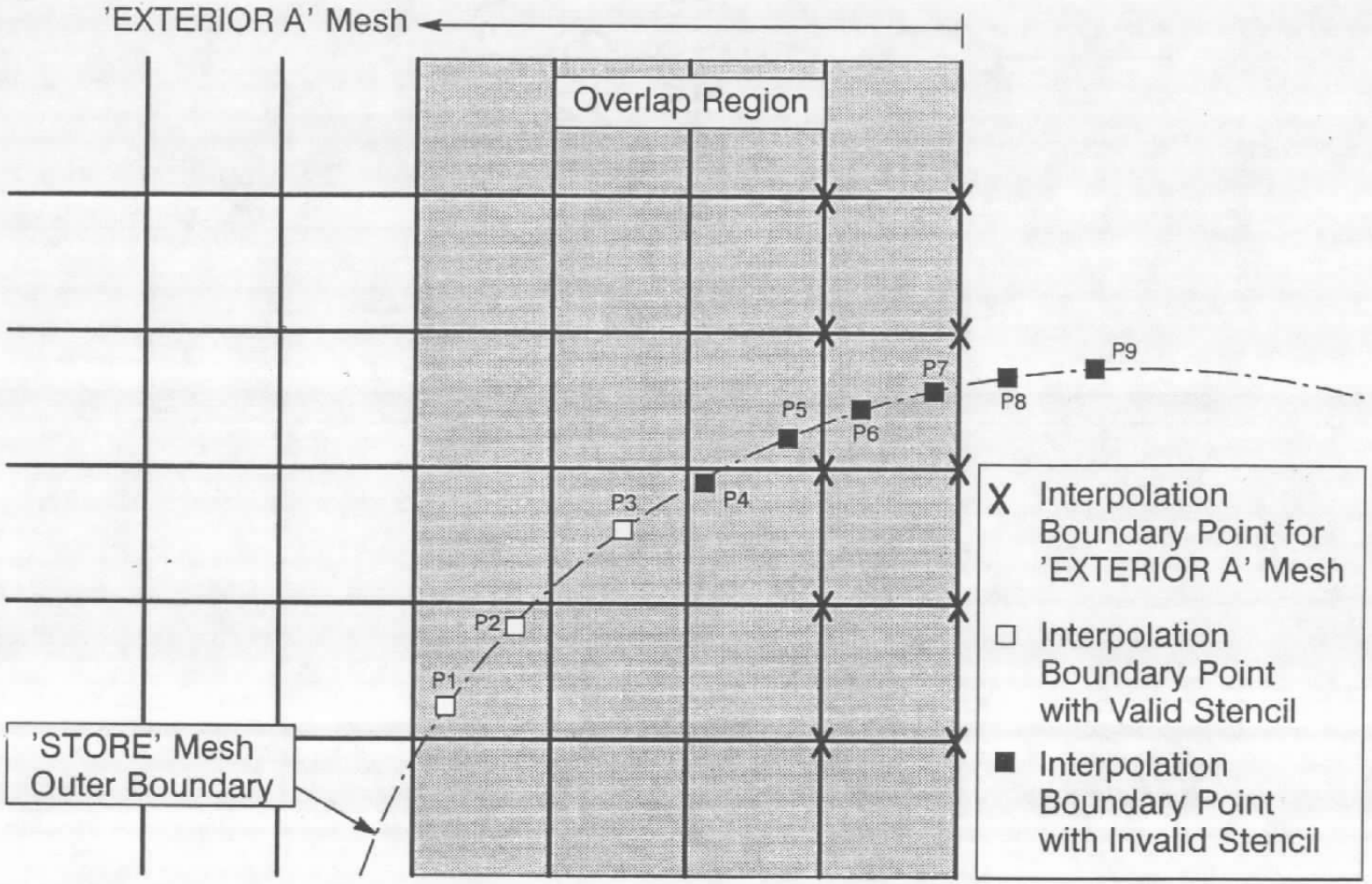


c. 'EXTERIOR B' mesh
Figure G-16. Concluded.

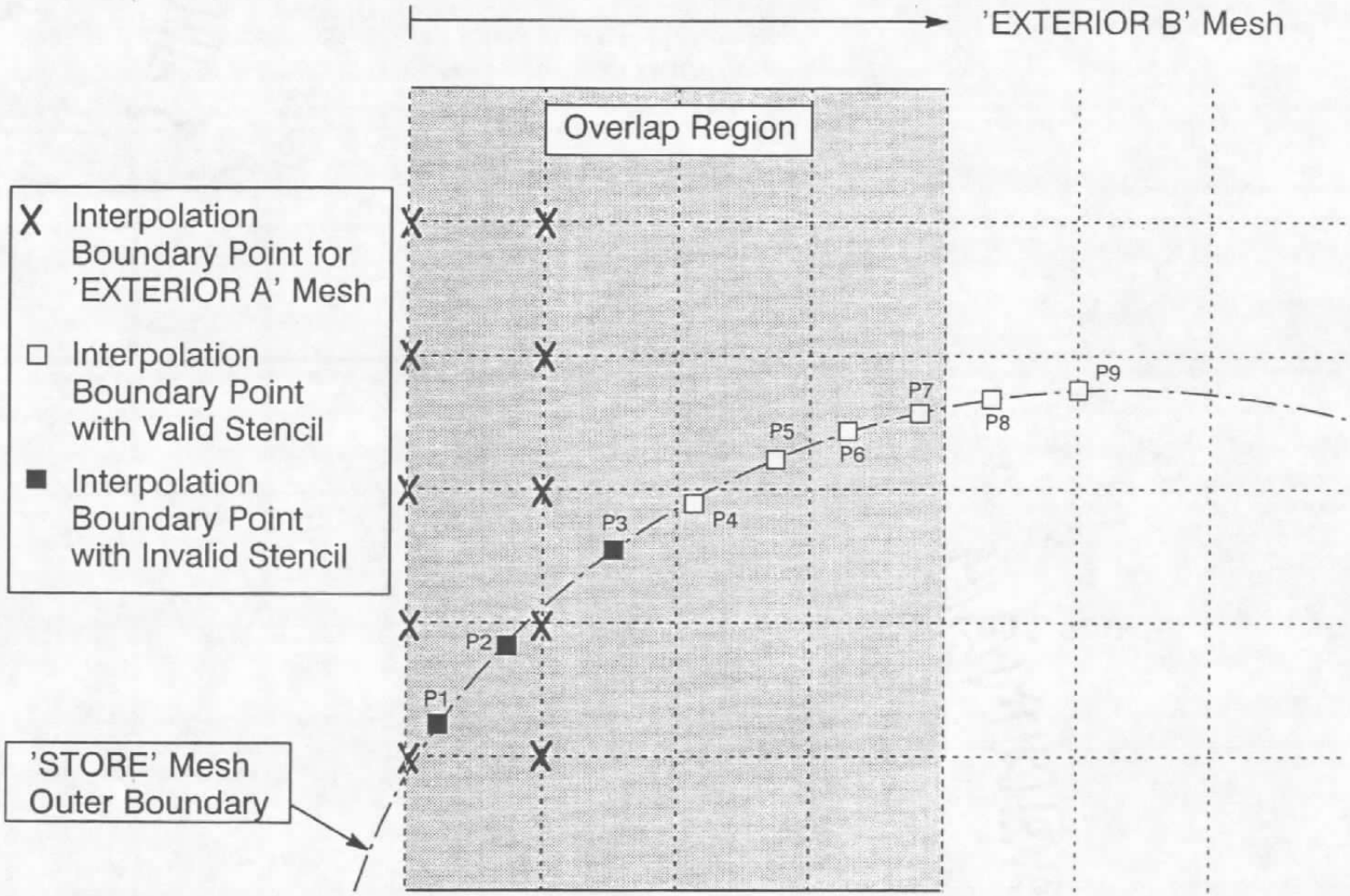


a. Overlap region

Figure G-17. Five-point overlap.



b. 'EXTERIOR A' mesh
Figure G-17. Continued.



c. 'EXTERIOR B' mesh
Figure G-17. Concluded.

```
                    DIAGNOSTIC MAPS
              LEGEND

BOUNDARY POINTS UPDATED BY :
 A  -  EXTERIOR A
 B  -  EXTERIOR B
 C  -  CAVITY A
 D  -  CAVITY B
 E  -  CAVITY C
 F  -  STORE
 G  -  STING
 H  -  INTERFACE


INTERPOLATION STENCILS UPDATING POINTS IN :
 a  -  EXTERIOR A
 b  -  EXTERIOR B
 c  -  CAVITY A
 d  -  CAVITY B
 e  -  CAVITY C
 f  -  STORE
 g  -  STING
 h  -  INTERFACE


 .  -  FIELD POINT
 #  -  HOLE POINT
 ?  -  ORPHANED BOUNDARY POINT
 *  -  INTERPOLATION STENCIL UPDATING MORE THAN ONE
       INTERPOLATION BOUNDARY POINT
```

a. Page 1

Figure G-18. Sample diagnostic maps for store/sting in a cavity.

```
MAP FOR EXTERIOR A

          PLANE  L =  26

    J         1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ..........................................................................
  2 ..........................................................bb.BB
  3 ..........................................................bb.BB
  4 ..........................................................bb.BB
  5 ..........................................................bb.BB
  6 ..........................................................bb.BB
  7 ..........................................................bb.BB
  8 ..........................................................bb.BB
  9 ..........................................................bb.BB
 10 ..........................................................bb.BB
 11 ..........................................................bb.BB
 12 ..........................................................bb.BB
 13 ..........................................hh.hhhh*b.BB
 14 ..........................................hh.hhhh*b.BB
 15 ..........................................hh.....bb.BB
 16 ..........................................**.....bb.BB
 17 ..........................................hh.....bb.BB
 18 ..........................................**.....bb.BB
 19 ..........................................**.....bb.BB
 20 ..........................................**.....bb.BB
 21 ..........................................**.....bb.BB
 22 ..........................................**.....bb.BB
 23 ..........................................**.....bb.BB
 24 ..........................................hh.....bb.BB
 25 ..........................................hh.....bb.BB
 26 ..........................................................................
```

b. Page 2

Figure G-18. Continued.

```
MAP FOR EXTERIOR B

          PLANE  L =  26

   J        1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ..........................................................................
  2 AA.aa.....................................................................
  3 AA.aa.....................................................................
  4 AA.aa.....................................................................
  5 AA.aa.....................................................................
  6 AA.aa.....................................................................
  7 AA.aa.....................................................................
  8 AA.aa.....................................................................
  9 AA.aa.....................................................................
 10 AA.aa.....................................................................
 11 AA.aa.....................................................................
 12 AA.aa.....................................................................
 13 AAh**hhhhhhhhhhhhhh*h*******h*hhhh.hhhhhhhhhhhhhhhh.hhhhhhh.hhh.h..
 14 AAh**hhhhhhhhhhhhhh*h*******h*hhhh.hhhhhhhhhhhhhhhh.hhhhhhh.hhh.h..
 15 AA.aa.....................................................................
 16 AA.aa.....................................................................
 17 AA.aa.....................................................................
 18 AA.aa.....................................................................
 19 AA.aa.....................................................................
 20 AA.aa.....................HHH.............................................
 21 AA.aa.....................GGGGG...........................................
 22 AA.aa.....................GG###GG.........................................
 23 AA.aa.....................GG#####GG.......................................
 24 AA.aa.....................HG#######GG.....................................
 25 AA.aa.....................HG#######GG.....................................
 26 ..........................................................................
```

c. Page 3

Figure G-18. Continued.

```
MAP FOR EXTERIOR B

          PLANE  L =  30

   J         1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
 K
 1 ..........................................................................
 2 AA.aa.....................................................................
 3 AA.aa.....................................................................
 4 AA.aa.....................................................................
 5 AA.aa.....................................................................
 6 AA.aa.....................................................................
 7 AA.aa.....................................................................
 8 AA.aa.....................................................................
 9 AA.aa.....................................................................
10 AA.aa.....................................................................
11 AA.aa.....................................................................
12 AA.aa.....................................................................
13 AA*********************************.***************.*******.***.*..
14 AA*********************************.***************.*******.***.*..
15 AA.aa.....................................................................
16 AA.aa.....................................................................
17 AA.aaHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
18 AA.aHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
19 AA.HH#####################################################################.
20 AA.HH#####################################################################.
21 AA.HH#####################################################################.
22 AA.HH#####################################################################.
23 AA.HH#####################################################################.
24 AA.HH#####################################################################.
25 AA.HH#####################################################################.
26 ..........................................................................
```

d. Page 4

Figure G-18. Continued.

```
MAP FOR CAVITY A

          PLANE  L =  56

    J         1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
   K
   1 ################################
   2 ################################
   3 ################################
   4 #####*a*a*a*a*a*a*a*a*a*a*a*####
   5 ###**a*a*a*a*a*a*a*a*a*a*a***###
   6 ###aa.....................dd.DD
   7 ##a*#.....................dd.DD
   8 ###*#.....................dd.DD
   9 ###*#.....................dd.DD
  10 #####.....................dd.DD
  11 ###*#.....................dd.DD
  12 ###*#.....................dd.DD
  13 #####.....................dd.DD
  14 ###*#.....................dd.DD
  15 #####.....................dd.DD
  16 ###*#.....................dd.DD
  17 #####.....................dd.DD
  18 ###*#.....................dd.DD
  19 ###*#.....................dd.DD
  20 #####...........FFFFFFFFFFFFFF
  21 ###*#.........FFFFFFFFFFFFFFFF
  22 #####........FFF###############
  23 #####.......FF#################
  24 ###*#......FF##################
  25 #####.....FF###################
  26 ###*#....FF####################
  27 #####f...FF####################
  28 #####...FF####################
  29 ###*#...FF####################
  30 ###a**.FF####################
  31 #####.........................
```

e. Page 5

Figure G-18. Continued.

```
MAP FOR CAVITY B

          PLANE  L =  56

   J         1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ################################
  2 ################*##*############
  3 ################*##*###########
  4 a*a*a*a*a*a*a*a************b*#
  5 #*a***a*a*a*a*a*a************b##
  6 CC.cc.....................ee.EE
  7 CC.cc.....................ee.EE
  8 CC.cc.....................ee.EE
  9 CC.cc.....................ee.EE
 10 CC.cc.....................ee.EE
 11 CC.cc.....................ee.EE
 12 CC.cc.....................ee.EE
 13 CC.cc.....................ee.EE
 14 CC.cc.....................ee.EE
 15 CC.cc.....................ee.EE
 16 CC.cc.....................ee.EE
 17 CC.cc.....................ee.EE
 18 CC.cc.....................ee.EE
 19 CC.cc.....................ee.EE
 20 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFE
 21 FFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
 22 ################################F
 23 #################################
 24 #################################
 25 #################################
 26 #################################
 27 #################################
 28 #################################
 29 #################################
 30 #################################
 31 ................................
```

f. Page 6

Figure G-18. Continued.

```
MAP FOR CAVITY C

         PLANE  L =  56

   J       1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
 K
 1 ################################
 2 ################################
 3 ################################
 4 ###########################b#####
 5 ###########################b*###
 6 DD.dd.....................bb###
 7 DD.dd.....................*####
 8 DD.dd.....................*####
 9 DD.dd.....................*####
10 DD.dd.....................#####
11 DD.dd.....................*####
12 DD.dd.....................*####
13 DD.dd.....................#####
14 DD.dd.....................*####
15 DD.dd..*****g............#####
16 DD.dd.....................*####
17 DD.dd.....................#####
18 DD.dd.....................*####
19 DD.dd.....................*####
20 FFFFd.....................#####
21 FFFFF.........GGGG........*####
22 ####FFFFGGGGGGGGGG........#####
23 #####FFFGGGGG####GG.......#####
24 ####################GG.....*####
25 ######################GG....#####
26 ####################GG....*####
27 ######################GG....#####
28 ####################GG....#####
29 ######################GG...*####
30 ####################GG...bb###
31 ........................#####
```

g. Page 7

Figure G-18. Continued.

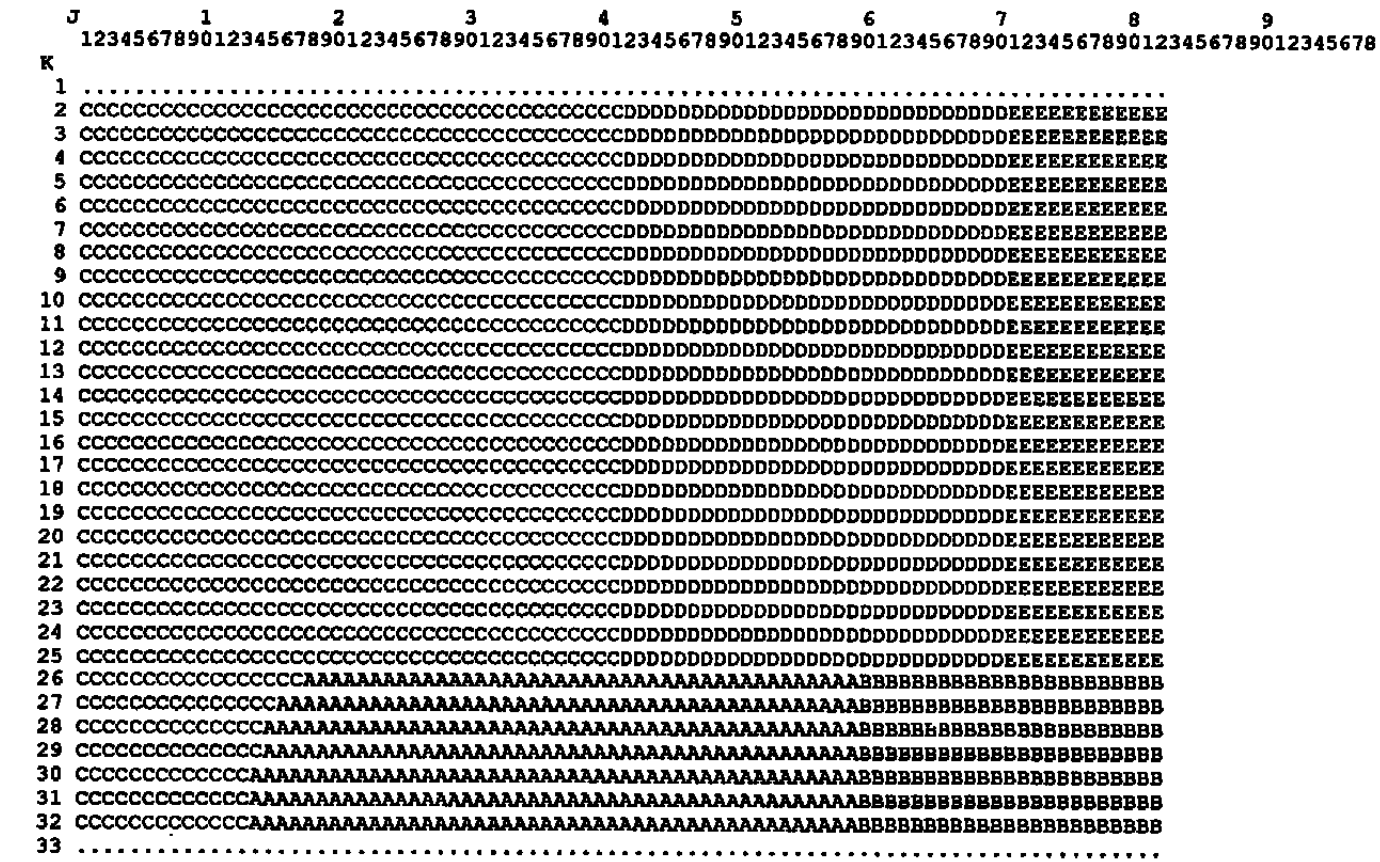
```
MAP FOR CAVITY C

          PLANE  L =  80

   J        1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  2 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
  3 DD.dd.....................BB
  4 DD.dd.....................BB
  5 DD.dd.....................BB
  6 DD.dd.....................BB
  7 DD.dd.....................BB
  8 DD.dd.....................BB
  9 DD.dd.....................BB
 10 DD.dd.....................BB
 11 DD.dd.....................BB
 12 DD.dd.....................BB
 13 DD.dd.....................BB
 14 DD.dd.....................BB
 15 DD.dd........g**.**.......BB
 16 DDf***f*******g.....*g.....BB
 17 DD.dd.......*........gg.....BB
 18 DD.dd..*..**.............BB
 19 DD.dd..*.................BB
 20 DD.dd..*........GG.........BB
 21 DD.dd..*.......GGGG........BB
 22 DD.dd..*......GG##GG.......BB
 23 DD.dd..*.....GG####GG......BB
 24 DD.dd..*.....GG#####GG.....BB
 25 DD.dd..*.....GG#####GG.....BB
 26 DD.dd..*....GG######GG....g.BB
 27 DD.dd..*....GG######GG.....BB
 28 DD.dd.......GG#######GG.....BB
 29 DD.dd..*....GG#######GG.....BB
 30 DD.dd.......GG#######GG.....BB
 31 ............................
```

h. Page 8

Figure G-18. Continued.

MAP FOR STORE

PLANE L = 29



i. Page 9

Figure G-18. Continued.

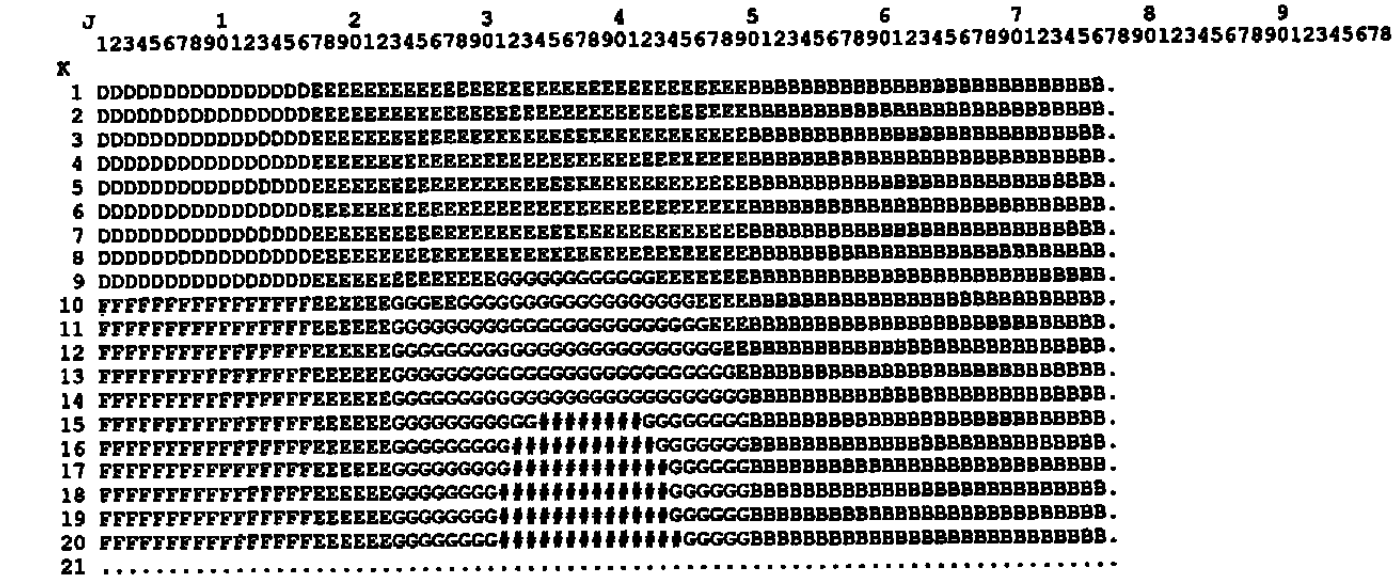
```
MAP FOR STING

          PLANE  L =  29

   J        1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ..........................................................................................
  2 EEEEEEEEEEEEEEEEE#BBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  3 EEEEEEEEEEEEEEEEE#BBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  4 EEEEEEEEEEEEEEEEE#BBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  5 EEEEEEEEEEEEEEEEE#EEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  6 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  7 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  8 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
  9 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 10 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 11 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 12 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 13 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 14 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 15 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 16 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 17 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 18 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 19 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 20 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 21 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 22 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 23 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 24 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 25 EEEEEEEEEEEEEEEEEEEEEEEEEBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 26 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 27 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 28 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 29 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 30 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 31 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 32 BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH.
 33 ..........................................................................................
```

j. Page 10

Figure G-18. Continued.

MAP FOR INTERFACE

PLANE L = 1



k. Page 11

Figure G-18. Continued.

```
MAP FOR INTERFACE

          PLANE  L =  10

   J        1         2         3         4         5         6         7         8         9
    12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
 K
 1 AAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB.
 2 AAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB.
 3 AA.....
 4 AA.....
 5 AA.....
 6 AA.....
 7 AA.....gg**********.....
 8 AA.....g***g*.....*g****g.....
 9 AA.....***g.....g****.....
10 AA.....**.....g***.....
11 AA.....g*.....***.....
12 AA.....*g.....GGGGGG.....***.....
13 AA.....*.....GGGGGGGGGG.....****.....
14 AA.....g*.....GGG#####GGGG.....g******gg.gg.g.g.g..gg.gggg.
15 AA.....GG#########GGGG.....
16 AA.....**...GG##########GGGG...********gg*.*.*.*..**.****.
17 AA.....*g...GG##########GGGG..*********.*.*.*.*.gg*.****.
18 AA.....GG##########GGGG.....
19 AA.....GG##########GGGGG.....
20 AA.....*g..GG##########GGGGG.*********.*.*.*.*.*.*.****.
21 .....
```

1. Page 12
Figure G-18. Continued.

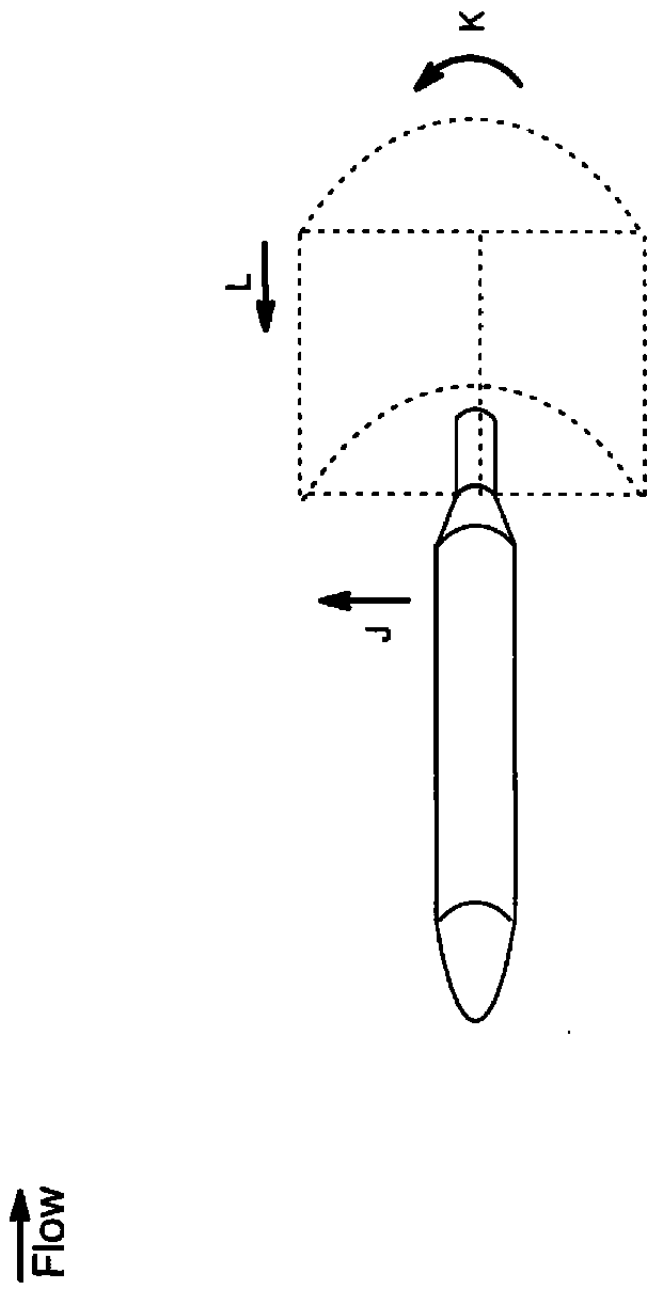
MAP FOR INTERFACE

PLANE L = 15

```
 J          1         2         3         4         5         6         7         8         9
   12345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
 K
 1 AAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB.
 2 AAAAAAABBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB.
 3 AA............................................................................
 4 AA............................................................................
 5 AA............................................................................
 6 AA............................................................................
 7 AA.........................................g..g.***.***.......................
 8 AA.....................................g...g..g...............................
 9 AA.................................g....g.....................................
10 AA...............................gg...........................................
11 AA............................................................................
12 AA............................gg...............GGGGGG..............G.G..G.....
13 AA............................*.............GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG.
14 AA............................g...........GGGGG######GGGGGGGGGGGGGG#G#GG#GGGG.
15 AA......................................GGGG##################################
16 AA............................g........GGGG###################################
17 AA....................................GGGG####################################
18 AA...................................GGGG#####################################
19 AA..................................GGGGG#####################################
20 AA..................................GGGGG#####################################
21  .............................................................................
```

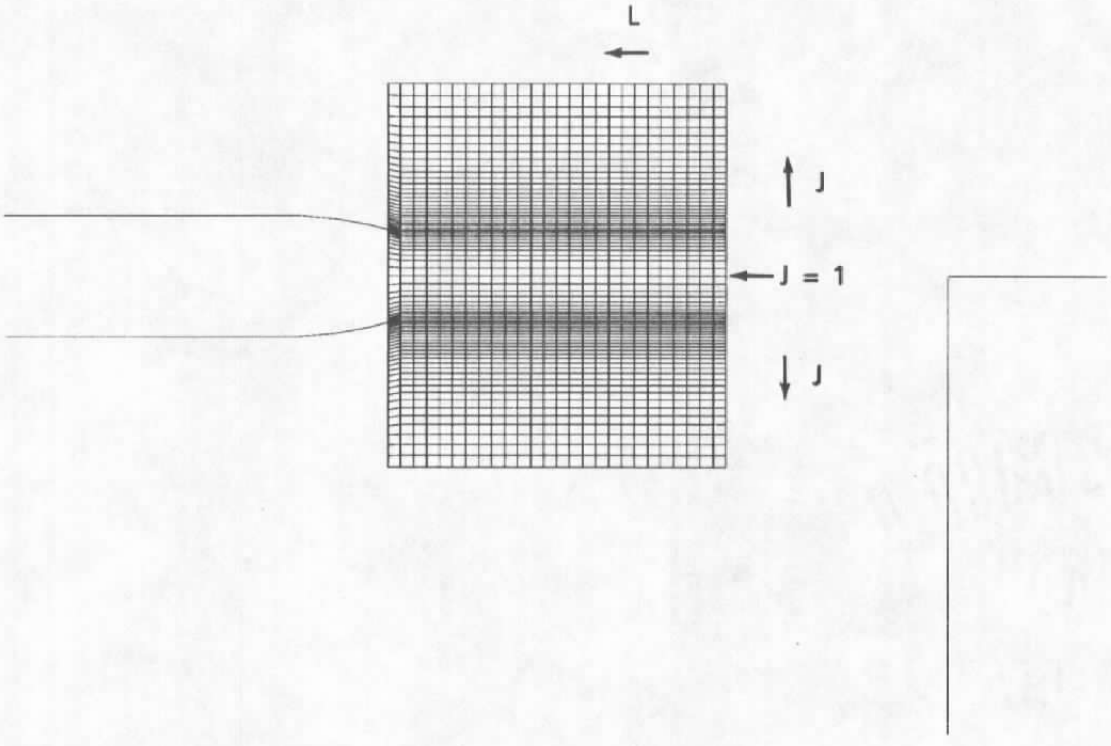
m. Page 13

Figure G-18. Concluded.

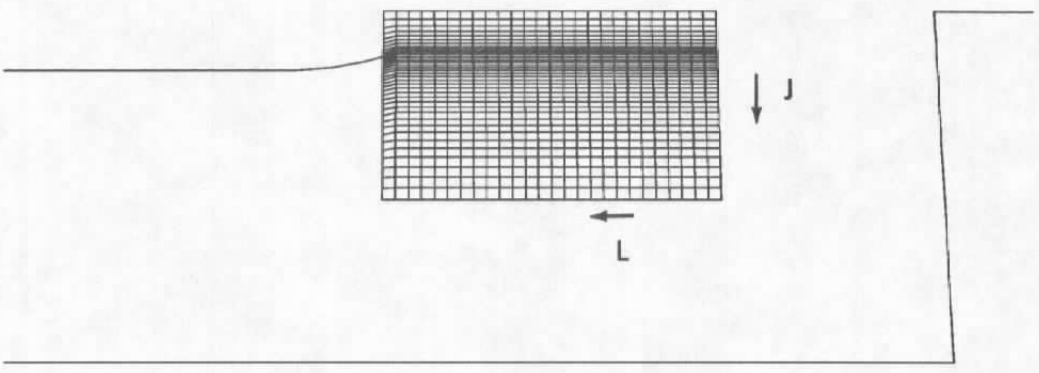


a. Graphical depiction

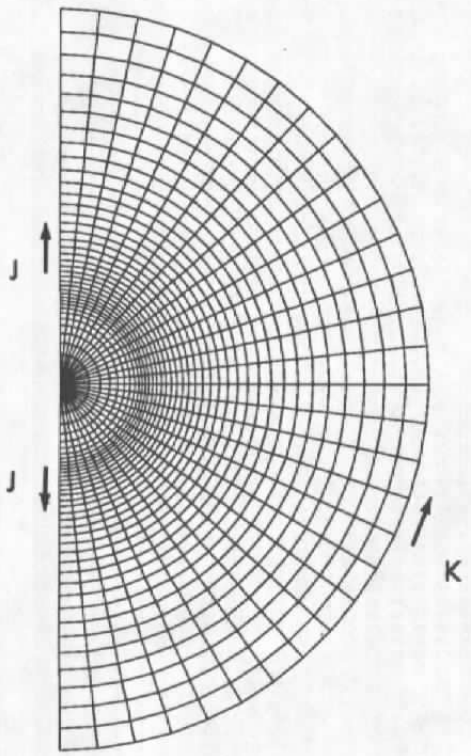
Figure G-19. 'STORE CAP' mesh.



b. Side view
Figure G-19. Continued.



c. Top view
Figure G-19. Continued.



d. Front view
Figure G-19. Concluded.

```
************************************************************
C
C                    PEGSUS 4.0 INPUT
C
C            EXAMPLE G-3 STORE ALONE IN A CAVITY
C                      7 MESH CASE
C            STING AND INTERFACE MESHES SUBTRACTED.
C            STORE CAP MESH ADDED.
C
C************************************************************


C************************************************************
C
C  MESH DEFINITIONS
C
C************************************************************

 $MESH NAME = 'STING',
       MODE = 'SUB',
                                    $END
 $MESH NAME = 'INTERFACE',
         MODE = 'SUB',
                                    $END


 $MESH NAME = 'STORE CAP',
       MODE = 'ADD',
       LINK = 'STORE',
              'CAVITY C',
              'EXTERIOR B',
              'CAVITY B',
              'EXTERIOR A',
              'CAVITY A',
       ADDLINK = 'STORE',
                 'EXTERIOR A',
                 'EXTERIOR B',
                 'CAVITY A',
                 'CAVITY B',
                 'CAVITY C',
       X0 = 15.468, Y0 = 0.0, Z0 = 0.0,
                                    $END
```

a. Page 1

Figure G-20. User input for store alone in a cavity.

```
C*************************************************************
C
C  INTERPOLATION BOUNDARY DEFINITIONS
C
C*************************************************************


C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  STORE CAP
C+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C---------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C---------------------------------------------------------------


 $BOUNDARY  NAME = 'STORE CAP OUTER BOUNDARY',
            ISPARTOF = 'STORE CAP',
                                  $END
 $SURFACE ISPARTOF = 'STORE CAP OUTER BOUNDARY',
           JRANGE = 12,39,
           KRANGE = 2,32,
           LRANGE = 26,27,
                                  $END
 $SURFACE ISPARTOF = 'STORE CAP OUTER BOUNDARY',
           JRANGE = 38,39,
           KRANGE = 2,32,
           LRANGE = 1,27,
                                  $END
 $SURFACE ISPARTOF = 'STORE CAP OUTER BOUNDARY',
           JRANGE = 1,39,
           KRANGE = 2,32,
           LRANGE = 1,2,
                                  $END
C---------------------------------------------------------------
C  INTERIOR REGION DEFINITION
C---------------------------------------------------------------

 $REGION NAME = 'STORE CAP REGION',
           TYPE = 'INTR',
           ISPARTOF = 'STORE CAP',
                                  $END

 $VOLUME ISPARTOF = 'STORE CAP REGION',
           JRANGE = 1,11,
           KRANGE = 1,33,
           LRANGE = 26,27,
                                  $END
```

b. Page 2
Figure G-20. Continued.

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  STORE
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C--------------------------------------------------------------------
C  OUTER BOUNDARY DEFINITION
C--------------------------------------------------------------------

 $BOUNDARY  NAME = 'AFT STORE OUTER BOUNDARY',
            MODE = 'ADD',
            ISPARTOF = 'STORE',
                                $END

 $SURFACE ISPARTOF = 'AFT STORE OUTER BOUNDARY',
          JRANGE = 77,82,
          KRANGE = 2,32,
          LRANGE = 1,2,
                                $END
```

```
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C  CAVITY C
C++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++

C--------------------------------------------------------------------
C  SUBTRACTED OUTER BOUNDARY DEFINITION
C--------------------------------------------------------------------

 $BOUNDARY NAME = 'AFT CAVITY HOLE BOUNDARY',
           MODE = 'SUB',
                                $END

C  END OF STORE ALONE IN A CAVITY CASE
```

c. Page 3
Figure G-20. Concluded.

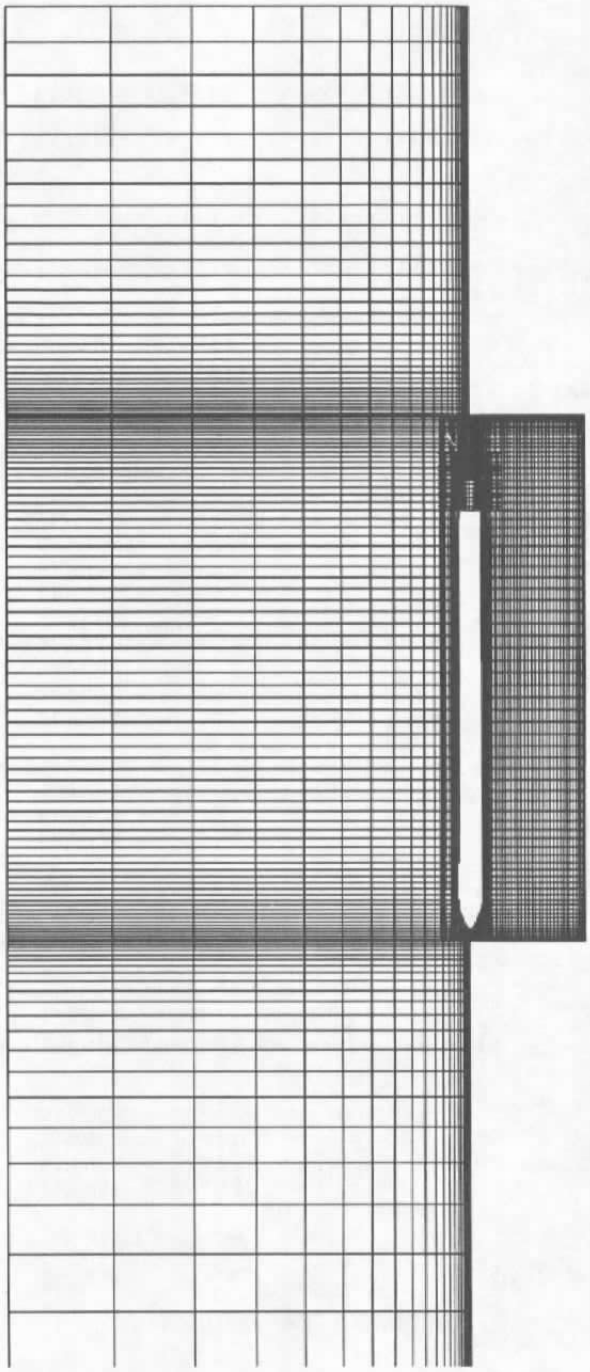
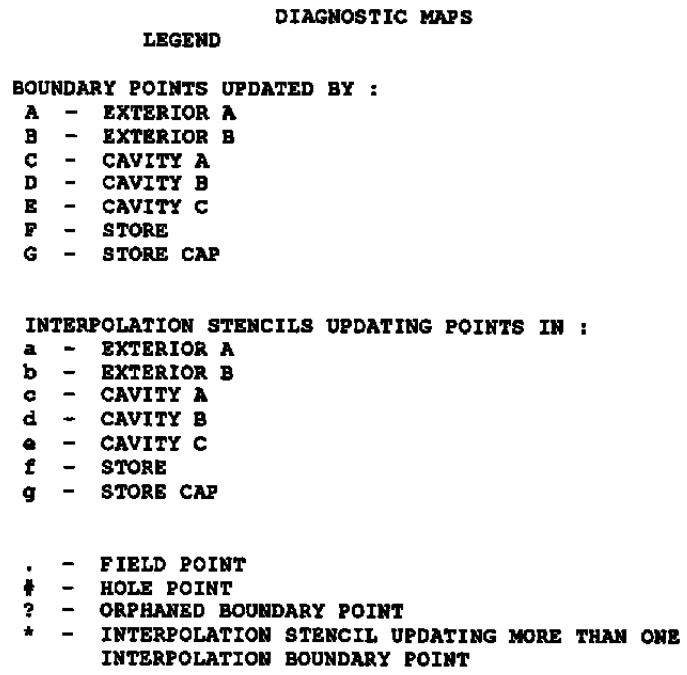


Figure G-21. Hole made by a store.



a. Page 1

Figure G-22. Sample diagnostic maps for store alone in a cavity.

```
MAP FOR CAVITY C

          PLANE  L =  56

   J        1         2         3         4         5         6         7         8         9
    123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ################################
  2 ################################
  3 ################################
  4 ###########################b#####
  5 ###########################b*###
  6 DD.dd.....................bb###
  7 DD.dd.....................*####
  8 DD.dd.....................*####
  9 DD.dd.....................*####
 10 DD.dd.....................#####
 11 DD.dd.....................*####
 12 DD.dd.....................*####
 13 DD.dd.....................#####
 14 DD.dd.....................*####
 15 DD.dd.....................#####
 16 DD.dd.....................*####
 17 DD.dd.....................#####
 18 DD.dd.....................*####
 19 DD.dd.....................*####
 20 FFFFd.....................#####
 21 FFFFF.....................*####
 22 ####FFFFG.................#####
 23 #####FFFGG................#####
 24 #########GG...............*####
 25 #########GG...............#####
 26 #########GG...............*####
 27 #########GG.....,.....,....#####
 28 #########GG..*...,.....,....#####
 29 #########GG...............*####
 30 #########GG..*............bb###
 31 ..........................#####
```

b. Page 2

Figure G-22. Continued.

```
MAP FOR STORE CAP

          PLANE  L =  27

    J         1         2         3         4         5         6         7         8         9
     123456789012345678901234567890123456789012345678901234567890123456789012345678901234567890123456789012345678
  K
  1 ###########.............................
  2 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  3 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  4 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  5 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  6 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  7 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  8 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
  9 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 10 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 11 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 12 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 13 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 14 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 15 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 16 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 17 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 18 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 19 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 20 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 21 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 22 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 23 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 24 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 25 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEEE
 26 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFEBB
 27 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFBBB
 28 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFBBB
 29 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFBBB
 30 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFBBB
 31 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFBBB
 32 ###########FFFFFFFFFFFFFFFFFFFFFFFFFFBBB
 33 ###########.............................
```

c. Page 3

Figure G-22. Concluded.

Table G-1. Mesh Dimensions

Mesh Name	JMAX	KMAX	LMAX
EXTERIOR A	67	26	35
EXTERIOR B	67	26	35
CAVITY A	30	31	85
CAVITY B	30	31	85
CAVITY C	30	31	85
STORE	82	33	29
STING	82	33	29
INTERFACE	77	21	35
STORE CAP	39	33	27

APPENDIX H
SAMPLE CRAY JOB CONTROL LANGUAGE

The Cray job control language for each example given in Appendix G is shown in Figs. H-1, H-2, and H-3, respectively.

```
JOB,JN=B30276A,T=900,MFL.
ACCOUNT,AC=58000502,US=B30276.
FETCH,DN=PEGSUS,DF=CB,^
    TEXT='DSN=B30276.PEGSUS40.PORT,DISP=SHR'.
 FETCH,DN=GRID1,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM16(CAVVLG1),DISP=SHR'.
 FETCH,DN=GRID2,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM16(CAVVLG2),DISP=SHR'.
 FETCH,DN=GRID3,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM16(CAVVLG3),DISP=SHR'.
 FETCH,DN=GRID4,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM16(CAVVLG4),DISP=SHR'.
 FETCH,DN=GRID5,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM16(CAVVLG5),DISP=SHR'.
*
*
COPYR,I=GRID1,O=INGRID,NR=3.
COPYR,I=GRID2,O=INGRID,NR=3.
COPYR,I=GRID3,O=INGRID,NR=3.
COPYR,I=GRID4,O=INGRID,NR=3.
COPYR,I=GRID5,O=INGRID,NR=3.
REWIND,DN=INGRID.
*
*
CFT77,I=PEGSUS.ON=F.
LDR,SET=INDEF.
*
*
  DISPOSE,DN=MAPS,DC=ST,DF=CB,^
    TEXT='DSN=B30276.PEGOUT.DATA(MAPSTR1),DISP=SHR'.
  DISPOSE,DN=ORPHAN,DC=ST,DF=CB,^
    TEXT='DSN=B30276.PEGOUT.DATA(ORPHTR1),DISP=SHR'.
*
 DISPOSE,DN=COMPOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(COMPTR1),DISP=SHR'.
 DISPOSE,DN=INTOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(INTRTR1),DISP=SHR'.
 DISPOSE,DN=RSTOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(RSTTR1),DISP=SHR'.
 DISPOSE,DN=IBPLOT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(IBLKTR1),DISP=SHR'.
*
*
EXIT.
```

Figure H-1. Cray job control language for empty cavity.

```
JOB,JN=B30276A,T=900,MFL.
ACCOUNT,AC=58000502,US=B30276.
FETCH,DN=PEGSUS,DF=CB,^
    TEXT='DSN=B30276.PEGSUS40.FORT,DISP=SHR'.
 FETCH,DN=GRID1,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM13(STOREPM8),DISP=SHR'.
 FETCH,DN=GRID2,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM10(STINGPM6),DISP=SHR'.
 FETCH,DN=GRID3,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM16(CAVVLIF),DISP=SHR'.
*
 FETCH,DN=COMPIN,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(COMPTR1),DISP=SHR'.
 FETCH,DN=INTIN,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(INTRTR1),DISP=SHR'.
 FETCH,DN=RSTIN,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(RSTTR1),DISP=SHR'.
*
*
COPYR,I=GRID1,O=INGRID,NR=3.
COPYR,I=GRID2,O=INGRID,NR=3.
COPYR,I=GRID3,O=INGRID,NR=3.
REWIND,DN=INGRID.
*
*
CFT77,I=PEGSUS1.ON=F.
LDR,SET=INDEF.
*
*
  DISPOSE,DN=MAPS,DC=ST,DF=CB,^
    TEXT='DSN=B30276.PEGOUT.DATA(MAPSTR2),DISP=SHR'.
  DISPOSE,DN=ORPHAN,DC=ST,DF=CB,^
    TEXT='DSN=B30276.PEGOUT.DATA(ORPHTR2),DISP=SHR'.
*
 DISPOSE,DN=COMPOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(COMPTR2),DISP=SHR'.
 DISPOSE,DN=INTOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(INTRTR2),DISP=SHR'.
 DISPOSE,DN=RSTOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(RSTTR2),DISP=SHR'.
 DISPOSE,DN=IBPLOT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(IBLKTR2),DISP=SHR'.
*
*
EXIT.
```

Figure H-2. Cray job control language for store/sting added to empty cavity.

```
JOB,JN=B30276A,T=900,MFL.
ACCOUNT,AC=58000502,US=B30276.
FETCH,DN=PEGSUS,DF=CB,^
    TEXT='DSN=B30276.PEGSUS40.FORT,DISP=SHR'.
 FETCH,DN=GRID1,DF=TR,^
    TEXT='DSN=B30276.CAVITY.GEOM13(STORECAP),DISP=SHR'.
 FETCH,DN=COMPIN,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(COMPTR2),DISP=SHR'.
 FETCH,DN=INTIN,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(INTRTR2),DISP=SHR'.
 FETCH,DN=RSTIN,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(RSTTR2),DISP=SHR'.
*
*
COPYR,I=GRID1,O=INGRID,NR=3.
REWIND,DN=INGRID.
*
*
CFT77,I=PEGSUS1,ON=F.
LDR,SET=INDEF.
*
*
  DISPOSE,DN=MAPS,DC=ST,DF=CB,^
    TEXT='DSN=B30276.PEGOUT.DATA(MAPSTR3),DISP=SHR'.
  DISPOSE,DN=ORPHAN,DC=ST,DF=CB,^
    TEXT='DSN=B30276.PEGOUT.DATA(ORPHTR3),DISP=SHR'.
*
 DISPOSE,DN=COMPOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(COMPTR3),DISP=SHR'.
 DISPOSE,DN=INTOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(INTRTR3),DISP=SHR'.
 DISPOSE,DN=RSTOUT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(RSTTR3),DISP=SHR'.
 DISPOSE,DN=IBPLOT,DC=ST,DF=TR,^
    TEXT='DSN=B30276.PEGSUS40.GEOM(IBLKTR3),DISP=SHR'.
*
*
EXIT.
```

Figure H-3. Cray job control language for subtracted sting and added store cap.