Spring 2000

Curvilinear Interface Methodology for Finite-Element Applications

Ollie James Rose

Old Dominion University

CURVILINEAR INTERFACE METHODOLOGY

FOR FINITE-ELEMENT APPLICATIONS

by

Ollie James Rose
B. A. June 1969, East Carolina University
M. A. May 1975, University of Maryland
M. S. June 1976, East Carolina University

A Dissertation Submitted to the Faculty of
Old Dominion University in Partial Fulfillment of the
Requirements for the Degree of

DOCTOR OF PHILOSOPHY

ENGINEERING MECHANICS

OLD DOMINION UNIVERSITY
May 2000

Approved by:

Oktay Baysal (Director)

Norman F. Knight, Jr. (Co-Director)

Osama A. Kandil (Member)

Chuh Mei (Member)

ABSTRACT

CURVILINEAR INTERFACE METHODOLOGY
FOR FINITE-ELEMENT APPLICATIONS

Ollie James Rose
Old Dominion University, 2000
Director: Dr. Oktay Baysal
Co-Director: Dr. Norman F. Knight, Jr.

Recent trends in design and manufacturing suggest a tendency toward multiple centers of specialty which results in a need for improved integration methodology for dissimilar finite element or CFD meshes. Since a typical finite element or CFD analysis requires about 50% of an engineer's effort to be devoted to modeling and input, there is a need to advance the state-of-the-art in modeling methodology. These two trends indicate a need to for the capability to combine independently-modeled configurations in an automated and robust way without the need for global remodeling. One approach to addressing this need is the development of interfacing methodology which will automatically integrate independently modeled subdomains.

The present research included the following objectives: (i) to develop and implement computational methods for automatically remodeling non-coincident finite element models having a pre-defined interface, (ii) to formulate and implement a parametric representation of general space curves and surfaces with a well-defined orientation, and (iii) to demonstrate the computational methodology with representative two- and three-dimensional finite element models.

Methodology for automatically remodeling non-coincident subdomains was developed and tested for two- and three-dimensional, independently modeled subdomains. Representative classes of applications have been solved which gave good agreement with reference solutions obtained with conventional methods. The two-dimensional classes of problems solved included flat and curved membranes multiple subdomains having large gaps between the subdomains and general space curves representing an interface for re-modeling the portions of subdomains adjacent to the interface. The three-dimensional classes of problems solved includes multiple three-dimensional subdomains having large three-dimensional gap between previously modeled subdomains. The interface was represented by general surfaces with a well-defined orientation and having curvature in possibly more than one direction.

The results demonstrated the re-modeling methodology to be general, flexible in use, highly automated, and robust for a diverse class of problems. The research reported represents an important advancement in the area of automated re-modeling for computational mechanics applications.

This dissertation is dedicated to my loving wife, Detra, and my dear sons Eric and Carl for their support and encouragement during the long years that I have pursued this goal.

ACKNOWLEDGMENTS

First, I extend sincere thanks to my committee: Drs. Baysal, Knight, Kandil, and Mei for their patience, support, and guidance in helping me to bring this educational experience to successful fruition. Second, I acknowledge the support of the Computational Structures Branch at NASA Langley Research Center for providing a research grant in the early stage of this work. Third, I extend thanks to Jonathan Ransom of LaRC/CSB and to Dr. Tiwari of ODU/ME for hosting me as a NASA/ASEE Summer Fellow.

TABLE OF CONTENTS

Page

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

LIST OF SYMBOLS

a_1, a_2, a_3, a_4 - coefficients of local cubic spline basis functions of parametric coordinates; defined in equations (3.1) - (3.4).

$[A]$ - coefficients array for linear least-squares system.

b_k - one dimensional basis function for interface definition.

bs_k - interface basis function in s-direction for two-dimensional surfaces.

bt_k - interface basis function in t-direction for two-dimensional surfaces.

$[B]$ - matrix expressing the relation between strain and displacement derivatives with respect to element isoparametric coordinate variables.

$[B_E]_i$ - element level Boolean array that extract the face of an element on the interface for subdomain i.

$[B_G]_i$ - subdomain global Boolean array that extracts the elements on the interface for subdomain i.

$[B_{NZ}]_i$ - Boolean array that extracts non-zero contributions of elements and element faces that are on the interface for subdomain i; formed as a product of $[B_E]_i$ and $[B_G]_i$.

c_j - incremental chord length j between points along an interface curve.

$C^{(n)}$ - indicates continuity of a function through the nth derivative.

$\{c\}$ - vector of unknown coefficients in least-squares system.

dof - Number of generalized displacement degrees-of-freedom for the nodes in a finite element subdomain.

D - Euclidean distance; used as objective function in projection algorithm.

$\{D_i\}$ - vector of generalized nodal displacements (displacements or rotations) for subdomain i.

$[D_N]$ - array of shape function partial derivatives with respect to isoparametric variables; defined by equation (4.8) for two dimensions and equation (4.25) for three dimensions.

$[D_{NE}]$ - array of shape function partial derivatives extended with zero entries to conform to associated matrix dimensions; defined by equation (4.12) in two dimensions and equation (4.32) in three dimensions.

e - represents residual least-squares error norm.

E - Young's modulus for a given material.

$[E]$ - material property or constitutive matrix.

f_x, f_y, f_z - generic functions representing space-curve coordinates.

$[F_i]$ - vector of body forces per unit volume for subdomain i.

$[G_i]$ - matrix which that associates traction degrees-of-freedom to displacement degrees-of-freedom of the interface for subdomain i.

$[G_{block}]$ - block diagonal matrix with elements of inverse Jacobean matrix; defined by equation (4.11) in two dimensions and equation (4.30) in three dimensions.

$[H]$ - Boolean matrix used in defining the relation between strain and derivatives of displacement in isoparametric coordinates; defined by equation (4.10) in two dimensions and equation (4.28) in three dimensions.

J - determinate of Jacobean matrix relating physical coordinates to local isoparametric coordinates.

$[J]$ - Jacobean matrix relating physical coordinates to local isoparametric coordinates; defined by equations (4.7) and (4.24).

$[k_e]$ - element stiffness matrix; defined by equations (4.6) and (4.23).

$[K_i]$ - conventional assembled stiffness matrix for subdomain i.

$[M_i]$ - matrix that associates traction degrees-of-freedom with displacement degrees-of -freedom for subdomain I; defined by equation (2.20).

$[N_i]$ - vector of element shape functions for elements in subdomain i.

NP - number of points used to represent an interface curve or surface.

NB - number of breakpoints used in defining partition for set of interface points.

NIS(j) - number of subdomains connected to interface j.

NS - number of subdomains in model.

$\{P_i\}$ - externally applied point forces or moments for subdomain i.

$\{q_i\}$ - nodal degrees-of-freedom associated with displacement field for subdomain i.

$\{q_s\}$ - degrees-of-freedom associated with displacement field on interface surface.

$\{r\}$ - vector for right-hand-side of least-squares system.

$[R_i]$ - matrix of shape functions relating traction degrees-of-freedom to Lagrange multipliers for subdomain i.

s_j - normalized incremental chord length j along interface curve; taken as the curve parameterization in s-direction.

sb_k – breakpoint for defining cubic spline shape functions on the parametric space in the s-direction.

t_j - normalized incremental chord length j along interface curve; taken as the curve parameterization in t-direction.

tb_k – breakpoint for defining cubic spline shape functions on the parametric space in the t-direction.

$[T]$ - array of shape functions for interface representation.

u, v, w - displacements in physical coordinates along the x, y, and z directions.

x_0, y_0, z_0 - represents coordinates in physical space of interface point to be projected onto interface surface or curve.

$\{u_i\}$ - displacement field vector for subdomain i.

$\{v\}$ - displacement field vector on interface surface.

Greek

$\{\alpha_i\}$ - vector of traction degrees-of-freedom for subdomain i.

$[\Gamma]$ - inverse of Jacobian matrix.

$\{\varepsilon\}$ - strain field vector; defined by equation (2.12).

ζ - local dimensionless element coordinate.

η - local dimensionless element coordinate.

$\{\lambda_i\}$ - vector of Lagrange multipliers for constraint integral associated with subdomain i.

ν - Poisson's ratio for a material.

ξ - local dimensionless element coordinate.

ξ_k - partition point for piecewise definition of cubic spline.

Π - combined potential energy functional; defined by equation (2.4).

Π_i - potential energy functional for subdomain i.

$\{\Phi_i\}$ - vector of applied surface tractions for subdomain i.

Ω_i - represents subdomain i.

CHAPTER I
INTRODUCTION

1.1 Overview and Motivation

The finite element method of today (2000) is a powerful, versatile, and widely-used technique for solving engineering problems computationally. Three central reasons for the power, versatility, and widespread use of the method are: (i) the basic idea of the method is to replace a complex continuous problem with a simpler discrete one, (ii) the method, as practiced today, takes advantage of rapidly evolving computational technology involving computer hardware, software integration, and network communications, and (iii) the finite element method derives power from its ability to handle arbitrary configurations comprised of complex mechanical components, boundary conditions, and loading states.

The capability to treat a diversity of complex configurations is a two-edged sword, since such problems often require significant modeling effort by the analyst to define the transition from one geometric shape to another (e.g., circular to rectangular) or from one discretization level to another (e.g., coarse mesh to refined mesh). This situation has been highlighted in studies conducted by Clerk and Muller.[1] From their investigations, they concluded that the cost of a typical finite-element structural solution is divided into approximately 80% engineering and 20% computing cost. Of the engineering cost, 65% is modeling and input preparation and 35% is interpretation of results. One may therefore infer that just over half (52%) of the cost can be attributed to modeling and input preparation.

Recent trends in approaches to design and manufacturing suggest that the role of engineering computation in general, and that of finite element analysis in particular, may be changing. One emerging new feature of modern manufacturing is the tendency toward multiple collaborating centers of specialty with the resulting need for integration methodology. For instance, finite element

The journal model for this dissertation is *the International Journal for Numerical Methods in Engineering.*

discrete models of components may be created by various groups anywhere in the world and brought together at a central destination for integration and analysis. This trend is increasing as a result of companies subcontracting not only manufacturing tasks, but also engineering design and analysis tasks. These activities have been made possible by widespread growth of computer networks within and between companies. Structurally distinct subdomains are often modeled differently for convenience, except their interfaces must be defined with coincident nodes for analysis using "standard" finite-element methods. Several avenues have been explored in response to the need for combining these independently modeled subdomains, a particular avenue being the development of finite-element interface methodology. Consequently, a need exists for the capability to combine, re-model, and analyze independently modeled finite element domains in a convenient manner. The ability to combine independently modeled subdomains in an automated manner would open the possibility for creating and having available a library of standard finite element models for use in defining more complex configurations by combinations of these models. This would result in considerable savings.

A second trend is that, as labor costs rise and computation costs fall, there is growing interest in including finite element analysis as part of the design process, whereas it has traditionally been used as a design validation tool.

Taken together, these trends indicate a need to minimize the cost of modeling, especially to minimize the frequency of global remodeling. As methods to address these needs are developed, accuracy should be maintained. Indeed, it would be desirable to have adaptive modeling and analysis methods that *improve* the accuracy and reliability of results. Of course, this happy situation is rarely possible and inevitable compromises have been made in attempts to address the problem of reducing modeling labor.

1.2 Literature Survey

Discussion of the literature begins with a brief historical perspective of finite elements, intended to provide a context for interface research. More detailed historical perspectives and bibliographies are provided by Zienkiewicz,[2,3] Gupta and Meek,[4] Robinson,[5] Gallagher,[6] and Noor.[7,8] This general literature background is followed by a more detailed discussion of previous interface-related research in finite elements. The interface literature review is organized according to the different approaches to addressing the problem of achieving the transition from a fine to a coarse spatial discretization. Nearly all the literature in this area deals with applications to structural problems in solid mechanics.

A similar but less extensive survey of approaches to the mesh refinement problem in finite differences concludes the literature survey. Most of the literature in combining dissimilar meshes for finite differences or finite volumes deals with applications in computational fluid dynamics (CFD).

1.2.1 General Background

Although the term "finite element" was not coined until 1960, the idea of replacing a continuous problem by a simpler discrete one and making use of what amounts to "elements" as they are called today has been around since antiquity. For example, one of the greatest of the early mathematicians, Archimedes, is credited by H. Eves[9] with the earliest known applications of this idea to solve geometrical problems. In the first application, *ca.* 240 B.C., Archimedes approximated the perimeter of a circle using regular inscribed and circumscribed polygons. There the "elements" were straight-line segments comprising the polygons. With this technique, and without the advantage of computer technology, he obtained a value for π between 377/71 and 22/7 (3.14 to two decimals). The second application by Archimedes was in approximating the volume of a solid sphere. There the "elements" were thin slices of the sphere, treated as squat circular cylinders. Martin and Carey[10] credit Tsu Ch'ung Chih, an early Chinese engineer, with having approximated a circle with slender rectangular "elements" ca. 480 A.D. and having obtained, without computer technology, a rational approximation of π as 355/113 = 3.1415929; a value correct to six decimal places.

Threads for the tapestry of ideas that form the present subject of finite element analysis can be traced back to the late seventeenth and early eighteenth centuries. Variational formulations together with approximation functions form the core of today's finite-element theory. One of the first contributions to variational calculus was the brachistochrone problem[11] posed by Johann Bernoulli in 1696. Euler developed much of the subsequent mathematical theory for the calculus of variations in the eighteenth century[11] (e.g., the famous Euler-Lagrange equation). The use of approximating functions to obtain useful solutions of complicated vibrating systems was originated by Lord Rayleigh[12] (John William Strutt) in 1873. This procedure, which makes use of expressions for the maximum potential and kinetic energies, was generalized by W. Ritz[13] in 1909 to include equilibrium problems, and is now known as the Rayleigh-Ritz method.

Some of the ideas forming present-day finite elements were expressed in 1943 in a foundational paper by R. Courant.[14] This paper originated as a talk delivered before the May 3, 1941 meeting of the American Mathematical Society and was subsequently published in the Society Bulletin. In his paper, Courant formulated the two-dimensional elasticity problem (membranes) in variational form. He reviewed solution techniques based on classical Rayleigh-Ritz procedures and the method of gradients (steepest descent). There was a short section discussing statistical methods (now called Monte Carlo methods) and what he termed *finite differences*, but which contained the essence of certain finite-element concepts. For example, the ideas of "net-points" (nodes) and "linear interpolation functions" (shape functions) were introduced and discussed in that paper. Almost as an afterthought, in an appendix he proposed breaking a two-dimensional continuum domain into triangular regions and replacing the continuous fields with piecewise approximations over the triangles; in the next-to-last sentence he even calls the triangles *elements*, but not finite elements.

Very little in the way of applications was done with Courant's ideas for about ten years. A possible reason for this, in the opinion of G. Barron,[15] was that computers large enough and fast enough to exploit the method were not available until the 1950's. Another possible reason was that many engineers, who might have been interested in applications, were simply unaware of this work. After all, there was a wartime atmosphere during this era, and many engineers were doing critical national business. Because of the wartime atmosphere, extensive communication about certain subjects was probably discouraged. Even today, with electronic publishing, Internet databases, etc., it is not unusual for one to overlook work in another discipline. Whatever the real reason, interest waned until work by Poyla[16] (1952), Hersch[17] (1954), and Weinberger[18] (1956) applied Courant's method to compute bounds for eigenvalues, which marked a renewed interest in the subject by mathematicians. Also. by this time, computational capability and aerospace technology needs were fostering interest among engineers, particularly aerospace engineers.

The advent of electronic computers with sufficient capacity, reliability, and speed in the 1950's came about just as interest in advancing the development of jet aircraft was growing.[19] The stringent analysis requirements associated with jet aircraft development provided impetus for growth in fundamental aerospace technology. Because of weight and strength requirements in aircraft, part of the interest in more advanced structural analysis methods was surely inspired by the needs of the aerospace industry during this era of technological innovation. One of the first responses to these needs was groundbreaking work by Turner et al. in a classic paper[20] published in 1956. In this paper, the authors addressed the problem of deflection analysis for wing-box structures. In this work, the authors laid out many of the basic features of finite elements as known today. For example, bar elements were introduced, the classic stiffness matrix was assembled for the wing-box configuration, and the displacement/force equations were solved. Of course, some of the terminology was unlike the standard nomenclature in use today. Clough, one of the co-authors of this paper, subsequently coined the term "finite element" in a later (1960) paper[21] in which he applied the method to analyzing plane stress problems.

One year prior to Clough's paper (i.e., 1959), Greenstadt[22] outlined an approach using what he termed "cells." In his development, he described a procedure for representing the unknown function by a series of functions, each associated with one cell, which is reminiscent of today's "shape functions." Greenstadt's theory allowed for irregularly shaped cell meshes and included many of the fundamental mathematical ideas of present day finite element methodology. In just a few years after the basic theories and methods were published, computer codes were developed and the first finite-element textbooks were written.

Development of the first major general-purpose finite-element program, NASTRAN, was started by NASA in 1966, and was first released in 1969. By 1971, NASTRAN was available as a commercial product[23] and has been on the market since that date. Other major codes[15] (e.g., MARC, ANSYS, ABAQUS, ASKA, SESAM, and ADINA) soon followed, and the market has flourished to the present day. The first edition of a pioneering textbook by O. C. Zienkiewicz and Y. K. Cheung[24] was published in 1967. Having met with great success and widespread use, the book is now in its fourth edition and has grown from one volume to two volumes. By 1974, R. D. Cook[25] had produced the first edition of his classic textbook, which is now in its third edition. At present, researchers and students interested in finite elements have a plentiful selection of text and reference books from which to choose (e.g., Bathe,[26] Krishnamoorthy,[27] Huebner and Thornton,[28] Reddy,[29] Kardestuncer and Norrie[30]). In the two decades since the mid-1970's, as computer technology has advanced, finite-element methods have matured and have been widely applied as an engineering analysis tool. Further research directions for finite-element technology have progressed into the areas of nonlinear analysis, including buckling and collapse, and research into the computationally intensive field of optimization and design applications is active (e.g., Knight[31,32]).

Widespread application of the finite-element method and active research in the field inevitably inspired an interest in solving increasingly complex engineering problems, as well as creating a demand for flexibility and ease of use. Active research areas now include the development of new element formulations, new computational procedures, new numerical techniques for solving sets of algebraic equations, and advanced modeling methods. Modeling methods have evolved to include commercial pre- and post-processing software systems such as PATRAN, IDEAS, FEMB, and many others. The next section describes the methods used to model and interface multiple subdomains for structural and fluid applications and reviews the previous work in this area.

1.2.2 Methods for Modeling Multiple Structural Subdomains

The desire to solve problems arising from configurations with complex mechanical components led to sophisticated meshing requirements that often involved local cutouts and discontinuities. Attempts to address these challenges resulted in a variety of loosely related approaches. These approaches have been treated in the literature under one or more of the following categories: global/local analysis, zooming, substructure modeling, submodeling, mesh transition modeling, multiple methods, and interface technology. These methods are all approaches to solving the central problem of where, when, and how to introduce mesh refinement for accuracy and they differ in various ways and performance, although the methods are related. Particular features and attributes of each category are discussed next.

1.2.2.1 Global/Local Analysis Methods

Global/local analysis, in the usual sense as discussed by Ransom,[33] Ransom and Knight,[34] and Knight et al.,[35] is defined as a procedure to determine local, detailed stresses using information obtained from a previous, independent global analysis. In the three papers just cited, the authors presented four key components that are necessary for a successful global/local analysis procedure. First, there must be an "adequate" global analysis in that the global structural behavior must be accurately determined and local details must be included at least crudely. Second, there must be a strategy for identifying regions in the global model requiring refined analysis. Third, there needs to be a procedure for defining suitable "boundary conditions" along the global/local interface boundary. Fourth, the local analysis must be "adequate" in that the local detailed stress state is accurately determined and compatibility requirements along the global/local interface are satisfied. The authors of references 33-35 suggested that if critical regions requiring refined models are not known *a priori*, i.e., obvious regions with a high-stress gradient such as near cutouts, then stresses or strain energy computed from the global model may be used as a guide for where to introduce a local refined grid. Boundary conditions for the local analysis are obtained by an interpolation of solution variables based on the global model. Ransom[36] suggested that various global/local approaches differ mainly in methods of interpolation for the boundary conditions. In this method, no direct interfacing of the discretized models occurs. The interfacing is performed through the specification of "boundary conditions" on the local model based on the global model solution. Krishnamurthy and Raju[37] extended this method to develop an independent refinement and integration procedure for coupling a finite element and boundary element procedure that used a frontal solver approach.

1.2.2.2 Zooming Methods

The term "zooming" in connection with finite-element computations was introduced in a paper by Hirai,[38] wherein the method of zooming was applied to computing stress concentration factors for a benchmark case of a plate-with-hole in tension. The method is a type of adaptive grid refinement based on triangular elements. An extension of the method to allow for successive local zooming in areas of high stress gradient was published in a subsequent paper by Hirai et al.[39] There does not appear to be significant use of this method beyond these two papers.

1.2.2.3 Substructuring and Submodeling Methods

The concept of substructure modeling, in the sense of simplifying a structure as an assemblage of components, is a standard engineering approach that predates the computer-based finite-element method. One of the early papers discussing the use of substructures in the context of finite-element analysis was written by J. S. Przemieniecki[40] in 1963, in which he applied the method to an aircraft structure. Later (1966), I. C. Taig[41] extended those ideas and presented a systematic procedure for inclusion of substructure modeling into finite-element computer programs.

The concept of "macro elements" for substructures was introduced in 1977 by Cavendish and Gordon.[42] Substructure modeling of components typically requires coincident nodes along the substructure interface or boundary, and the resulting system of equations is solved using procedures that exploit the special matrix structure.

Submodeling, as discussed by Ransom[33] for the ANSYS analysis code, includes any method that has node-by-node correspondence at global/local interface boundaries. On the other hand, the term submodeling as used by Hibbitt et al.[43] is more in the sense of a global/local analysis as previously discussed. Hence, there is not complete standardization of terminology. Submodeling in ABAQUS is based on executing multiple (usually two) successive simulations. First the global model is solved for the entire structure, and then a "submodel" is solved using a refined mesh of a subregion of interest. The submodel boundary conditions are obtained by interpolating the global model solution.

1.2.2.4 Mesh Transition Modeling

Connecting a region having a refined mesh (local grid) and a region having a coarse mesh (global grid) while maintaining node-by-node correspondence has been accomplished in four ways. The four approaches are illustrated in Figure 1.1. Transitioning from a coarse mesh to a refined mesh can be accomplished using quadrilateral base elements and triangular transition elements as shown in Figure 1.1(a) or using all the same type and order of elements is shown in Figure 1.1(b), with distorted quadrilateral elements effecting the transition between the refined and coarse regions. The transition accomplished by variable-order quadrilateral elements is illustrated in Figure 1.1(c). Finally, the linking of displacements by multi-point constraints (interpolation) is presented in Figure 1.1(d).

Variable-order quadrilaterals are discussed, usually in the context of isoparametric elements, in many finite-element textbooks (e.g., Bathe[26] and Cook et al.[25]). In a brief note, Somervaille[44] applied them to mesh grading for plate bending problems. Gupta[45] treated two-dimensional transition elements and elements with a form of constraint nodes. Variable-order elements and triangular transition elements were discussed and applied to defining a so-called macro-element by Cavendish et al.,[42] which included an extension of the method to three-dimensional elements.

The mesh transition strategies shown in Figure 1.1 suffer from several drawbacks. If constant-strain triangles are used as the triangular transition elements, there is a possibility of severe error if the transition region happens to be in a high-gradient location; to a lesser extent, a similar danger exists with distorted quadrilateral transition elements. Furthermore, formulation and implementation of transition elements is quite difficult in three dimensions. A more serious weakness applying to any of these mesh transition strategies is that their use with independently-modeled subdomains is unlikely to be worthwhile because of the extensive modeling required in the transition region. It is desirable to have an alternative methodology free from these weaknesses.

1.2.2.5 Multiple Methods Integration

An alternative approach based on the concept of interface elements has been sponsored for several years by the NASA Langley Research Center. The rationale and basic concepts for this work were proposed by Housner et al.[46] at a

1990 NASA conference, and was subsequently published in a 1991 NASA conference proceeding. A concept termed "multiple methods integration" was discussed, and in this context, the idea of interfaces in finite-element structural analysis was introduced. The interface element idea is depicted in Figure 1.2, which shows two independently modeled finite-element domains having non-matching nodes along their adjacent edges with an associated "interface element." The interface element is intended to insure displacement compatibility and traction continuity in an integrated or variational sense. Additional early developments of interface methodology were also reported in 1991 by Housner et al.[47]

1.2.2.6 Interface Technology

Part of the motivation for the interface element concept stems from work published in 1988 by Maday et al.[48] in which the concept of "mortar elements" was introduced. Although this paper treated only spectral methods of solution and did not use Lagrange-multiplier constraints, it did present the basic idea of interfaces between contiguous subregions. Further motivation was derived from a subsequent paper in 1990 by Giles and Norwood[49] in which they discussed the concept of coupling data between structural regions obtained by different analysis methods, i.e., the concept of multiple methods. Other researchers have investigated the coupling of finite element models with boundary element models and finite difference models, particularly for modeling different physical phenomena in different domains (i.e., fluid-structure or acoustic-structural interactions). Hybrid analysis methods have also been developed (e.g., Rao et al.[50]).

Later, in 1991, Farhat and Roux,[51] in a paper primarily devoted to parallel solution algorithms, introduced the use of constraints and Lagrange multipliers to enforce compatibility at interface nodes. Since they were mainly interested in parallel algorithms, Farhat and Roux did not attempt a global solution using these methods. This work was extended by Farhat and Geradin[52] in 1992 to require fewer Lagrange multipliers with an associated increase in computational performance.

Use of the ideas of an interface element, Lagrange multiplier compatibility constraints, and a variational formulation based on minimum potential energy for obtaining a finite element solution for a structural configuration with two subdomains was presented by Aminpour et al.[53,54] The interface element presented in these two papers was a one-dimensional element comprised of straight, planar line segments. The interface element was developed to connect independently modeled subdomains by enforcing displacement continuity across the interface in a variational sense. The structure of the assembled finite-element equations was shown to become more complex than the assembled matrix for a standard finite-element model. In addition, the positive-definite feature of the assembled matrix was lost. The vector of unknowns in the assembled equations included the nodal displacement degrees of freedom from each subdomain, coefficients of the displacement approximation along the interface, and traction parameters for each subdomain. This one-dimensional interface element formulation has provided the inspiration for the present research.

Subsequently, Ransom et al.[55] extended this interface-element research to include an arbitrary number of connected two-dimensional subdomains and possible nesting of interfaces using the one-dimensional interface element. As such, these results illustrated the use of the interface technology in the same manner as telescoping substructures or the zooming method. The authors discussed three applications in Reference 55: (i) a composite laminated panel having two circular holes and loaded in tension, (ii) a composite laminated cylindrical panel having a central circular hole and loaded in compression, and (iii) a free-edge composite laminate loaded in tension. Solutions for models with interfaces were compared to solutions from a reference globally refined model without an interface. Results included normalized axial stress contours and line plots of axial stress along the panel centerlines. The interface results shown in this paper correlated well with the reference results.

In 1994, Davila et al.[56] further extended the one-dimensional interface-element capability to include subdomains with a cross-surface connection, i.e., domains having non-aligned or skewed grids and one-dimensional interfaces possibly curved in a plane. Reference 56 considered three applications of this extended capability.

The first application was a cantilevered plate with stiffener support and a tip-bending load applied to one end. The authors stated that this case represented a very stringent interface problem since the stiffener introduces a severe discontinuity in the moment and transverse force for the plate elements. Tip deflection results were shown in this paper and compared to "bracketing" non-interface cases where the stiffener was located at the nearest exact line of element edges on either side of the location for the coupled model. The authors argued that errors in the interface-element computed results were always smaller than errors introduced by modeling the stiffener on an exact line of element edges to avoid having a need for an interface. By this line of reasoning, the authors concluded that the method was therefore verified.

The second application considered in this paper was a blade-stiffened composite panel with a central hole loaded in axial compression. Results showing out-of-plane displacement contours and line plots of normalized axial stress for the interface case were compared to results from a reference solution having a refined global model without an interface. The interface results were in close agreement with the reference results. The third application was a fuselage panel with a window and two frames. The authors did not state what the loading condition on the panel was, nor were results from a reference solution shown for comparison. Radial displacement contours for the interface model were presented and briefly discussed. Evidently, this case was included just to illustrate the potential use of the cross-surface interface element for airframe-type substructures.

Further extensions to include three-dimensional structures with two-dimensional interfaces were discussed by Aminpour and Krishnamurthy.[57] Two applications were considered by the authors in this paper: (i) a solid, rectangular cantilever beam loaded in tension, bending, shear, and torsion and (ii) a solid flat plate with a central circular hole loaded in tension. The cantilever beam was modeled with two dissimilar subdomains, and cases with flat and curved interfaces (curvature in one direction) were shown. A reference model with a single global refined grid was used for comparison to results from the interface solutions. Qualitative displacement contours were presented which showed good agreement between the interface results and the reference results. Results from the interface model for the plate with hole were compared to a single-grid reference model. Axial displacement contours were given in the paper and there was close agreement between the reference and interface results. The authors of this paper implied that the capability was limited to models with interfaces that can be projected onto a plane and retain rectangular structure after projection.

The representation of the interface for these computations was accomplished using the software package, FITPACK. The extensions discussed by the authors did not include automatic local re-modeling to a specified interface surface; it did, however, allow for limited "noise" to be present in the element node coordinates along the interface.

Following this work, Aminpour et al.[58] removed the previous restriction of requiring the finite-element meshes on the interface to have a rectangular structure. With this paper, the two-dimensional interface could take on an arbitrary shape and the finite-element mesh could have an unstructured form. Simple test cases were considered wherein the interface element always remained in a plane.

Other applications of the one-dimensional interface element methodology include a proof-of-concept problem of a plate with a circular cutout[59] and a complex, stiffened crown panel[60] showing computed displacements and stresses. Housner et al.[61] gave a detailed summary of the interface-element methodology and reviewed applications through 1995. Applications to geometrically nonlinear problems were given by Ransom.[62]

1.2.3 Modeling Methods in Computational Fluid Dynamics

Solution procedures in computational fluid dynamics (CFD) are generally based on finite-difference or finite-volume algorithms, in contrast to solution procedures in solid mechanics that are usually finite-element based.

Thermal loading, which can occur in either fluids or solids, is treated by finite elements in solids and by finite differences/finite volumes in fluids.

The physical underpinning of finite elements is generally a variational statement expressing the principle of minimum total potential energy. The corresponding discrete mathematical expression of this formulation results in a node (or grid point) and its associated degrees of freedom being of central importance, as it represents the place at which generalized displacements are calculated. Additional physical quantities of interest, such as the stress field, are computed from the displacements using the basic elasticity equations and the constitutive relations.

By contrast, the physical underpinning of the finite difference/finite volume CFD method is a statement of conservation of mass, momentum, and energy. The flux of these various quantities through the walls of the grid cells is of central importance and cell-centered methods are widely used. Related quantities of interest are obtained from the constitutive relations and equations of state.

The contrasting needs of finite-element based solid mechanics computations and finite-difference-based fluid mechanics computations naturally give rise to different approaches to grid generation over the physical domains of interest. Moreover, the development process in each discipline appears to have proceeded more-or-less independently. In the case of fluids for example, a physical object (e.g., aircraft, turbine blade, projectile, etc.) serves as the inner boundary of the computational domain, and the outer boundary is either a containing vessel (e.g., a pipe) or the "far field." Thus, for the case of fluids, there is typically a very large volumetric region that must be discretized. If supersonic speeds are involved, shock waves must be captured or fitted, with either case requiring clustered grids or some type of interface. With viscous fluids, regions near solid bounding surfaces must have grid refinement in the surface-normal direction to resolve the boundary-layer velocity gradients adequately.

For reasons such as these, there has been a long-standing need for domain partitioning and corresponding interfacing procedures in CFD. Practitioners of CFD had been struggling with the interfacing problem long before it was addressed in solid mechanics. The "interface-type" technology in CFD is thus more mature than in solid mechanics computations. There are obvious differences in the disciplines, but there are some striking parallels and an opportunity for synergistic interaction may exist. This possibility is discussed in detail in Chapter VI, where recommendations for further research are presented.

In CFD, any method for generating and interfacing multiple complex grids is generally called a domain-decomposition technique or DDT (e.g., Fouladi-Semnani[63]). In essence, a DDT partitions the global computational domain into simpler subdomains on which subgrids with possible grid refinement or non-similar topologies are constructed. These subgrids are generally independent of each other and are created by any standard grid modeler, for example the GRIDGEN code (see Steinbrenner et al.[64]). For structured grids, there are three widely used methods of domain decomposition: (i) multiblock grids, (ii) zonal grids, and (iii) overlapped grids. Each method is examined next.

1.2.3.1 Multiblock Grid Methods

The multiblock approach partitions the grid domain into a set of mutually disjoint subgrids that completely fill the computation space without overlaps or voids. At the interfaces where the subgrids meet, the nodes match in a one-to-one manner. Information is passed between the subgrids by boundary conditions at the cell faces, which express one or more conservation laws.

An example of the multiblock approach was discussed by Arabshahi et al.[65] in which the authors presented solutions to the three-dimensional unsteady Euler equations for a wing/pylon/store configuration. The solutions were for a transonic flight condition (free-stream Mach number = 0.85) and include local shock waves as well as significant aerodynamic interference between the store, pylon, and wing. The Euler equations were written in conservative form and discretized using an implicit finite-volume formulation. The solution algorithm used flux difference splitting. Results shown in the paper compared predicted surface pressures with experiment and were in good agreement except near where there were large adverse pressure gradients. In such regions, the flow is probably separated, and an Euler method is not expected to give correct predictions.

A second representative example of the multiblock approach was presented by Nishida et al.[66] in which the method was applied to an Euler analysis of a wing-mounted propfan with slipstream effects included. The solution algorithm is based on a second-order central-difference scheme with artificial dissipation for stability. Results were given as pressure distributions compared with experiment and were in good agreement except in regions where Euler theory is clearly inadequate. Two noteworthy features of this investigation were that it employed a multigrid algorithm for convergence acceleration and it uses out-of-core storage for all blocks except the one currently being solved. The concept of a *multi-grid* algorithm is distinct from the concept of *multiblock grids* in the sense that multigridding is a systematic grid coarsening/refinement technique which uses a proper subset of a fine grid for the purpose of accelerating solution convergence (e.g., see Brandt[67] and Zhu and Craig[68]). The capability of out-of-core storage of inactive blocks is a possibility for any DDT.

1.2.3.2 Zonal Grid Methods

The zonal approach, also called grid patching, partitions the grid domain into a set of subgrids that completely fill the computation space without voids, but the requirement of one-to-one node matching between nodes of adjacent subgrids is relaxed. Communication between neighboring blocks is achieved by an interpolation procedure based on a one or two-cell overlap between the adjacent subgrids. The information required at the boundary of one zone is interpolated from the interior of another zone.

Two examples of the zonal approach were given by Holst et al.[69,70] The authors of these papers treated Euler and Navier-Stokes equations for transonic flow over wing/fuselage configurations. The Navier-Stokes solutions were based on a Baldwin-Lomax[71] turbulence model with a thin-layer approximation to the full Navier-Stokes equation (TLNS). The solution method used an alternating direction implicit (ADI) algorithm similar to the Beam-Warming[72] scheme. The procedure also employed out-of-core storage of inactive blocks. Results presented included computed pressure distributions compared with measured pressures and computed particle paths compared with oil-flow photographs. The pressure distributions were in good agreement with experiment except right where there was shock-induced separation. The authors of the paper asserted that the computed position of the separation was in good agreement with the oil-flows, as were many other qualitative details of the flow field.

1.2.3.3 Overlapping Grid Methods

The overlap method, also called the overset method or Chimera scheme, defines independent grids based on the local topology of a configuration component. For example, grids for a finite wing might be defined with a C-H topology, a fuselage with O-H topology, and a nacelle with O-O topology; all of which might be immersed in a global grid of H-H topology. The nomenclature: C-H, O-O, etc. is a standard CFD shorthand derived from the geometrical appearance of a grid topology. The essence of this nomenclature can be understood by reference to Figures 1.3 and 1.4, in which Figure 1.3 has two-dimensional examples and Figure 1.4 has three-dimensional examples. Figure 1.3(a) shows an airfoil section together with a grid having C-topology, Figure 1.3(b) shows a two-dimensional channel with a grid of H-topology, and Figure 1.3(c) shows a cylinder section having a grid of O-topology. Figure 1.4 (a) presents a finite wing having combined a grid of combined C-H topology and Figure 1.4(b) has a cylinder with an O-H grid. Other combinations are, of course, possible and are covered by the nomenclature.

Following the individual grid definitions, a preprocessor is then used to "cut holes" in the various grids where configuration components are defined. The preprocessor also sets up boundary-condition specifications to facilitate communication between neighboring blocks. The hole cutting procedure maintains overlap between the various grids at least two or three cells deep for information transfer based on an interpolation strategy.

Overlapped grids allow great versatility in modeling complex, realistic configurations. A disadvantage of the method is that it requires sophisticated bookkeeping to label and keep up with the various node types arising from the method. A second disadvantage, which may go away with better algorithms, is that it generally takes more computation time than a similar solution based on one-to-one multiblock methods. On the other hand, the multiblock approach is more labor intensive in the modeling phase, which will likely become a more serious weakness as computer technology improves and computation time is less of an issue.

One of the earliest papers using the overlap method was by Atta,[73] in which he developed a method for constructing a two-dimensional grid system for solving for the transonic flow about an airfoil wherein the airfoil grid was embedded in a rectangular global grid. A subsequent paper by Atta et al.[74] extended this two-dimensional overlap scheme to three dimensions and presented solutions for transonic flow over a wing/pylon/nacelle configuration.

Apparently, the first use of the term "chimera" was in a paper by Steger et al.[76] on grid generation in which the overlap method was used. The use of the word chimera in this regard is evidently an allusion to Greek mythology. In a standard dictionary edited by Daves,[76] there were two definitions given for the word chimera: (i) "a fire-breathing monster with the head of a lion, the body of a goat, and the tail of a serpent" and (ii) "an impossible or foolish fancy". It is not clear to which meaning the authors of Reference 76 were alluding, perhaps both. Under the auspices of U. S. Air Force sponsorship, this work was extended and resulted in a series of papers by Benek et al.[77,78,79] discussing various refinements and applications of the method.

A significant gain in efficiency for the overlap method was achieved by Baysal et al.[80] in an approach having two noteworthy features: (i) the method was extended to include multigridding in combination with overlapping and (ii) the method was employed as part of a hybrid scheme using the best features of each DDT technique. The basic grid-overlapping tool for this approach was the MaGGie code developed by Lessard.[81] Significant applications of the hybrid approach were accomplished by Fouladi-Semnani,[63] Newman and Baysal,[82] and Baysal et al.[83] The work by Fouladi-Semnani addressed the problem of supersonic viscous flow around stores exiting a cavity. Newman and Baysal[82] solved for transonic viscous flow around a wing/pylon/finned-store configuration. Baysal et al.[83] discussed supersonic viscous flow for a cylinder/fin/sting/cavity assembly. The computed pressure distributions in the paper of Reference 83 were compared to wind-tunnel measurements and showed excellent agreement except at the cavity rear face where there was experimental evidence of massive flow separation. The hybrid approach has also been applied to time-dependent problems for moving objects by Baysal et al.[84,85] and to aerodynamic shape optimization by Eleshaky and Baysal.[86]

Two other recent innovations for the overlap method include: (i) the introduction of so-called collar grids by Parks et al.[87] for resolving viscous details in confined regions with several close-fitting components, and (ii) an application to simulate rotor aerodynamics by Meakin.[88] As an indication of the maturity and acceptance of the method, it presently exists as a standard option in the widely used CFL3D flow solver developed and supported by the NASA Langley Research Center.

1.2.3.4 Dynamic Grid Methods

A currently active area of CFD grid methodology research is in that of dynamic mesh evolution for grids around bodies in relative motion. An advancement of the DDT research is Dynamic Domain Decomposition (D^3M) for structured grids and Dynamic Unstructured Technique (DUT) for unstructured grids. A discussion these two advances is given by Baysal et al.[89] in a useful review paper. A representative example of an application of the D^3M technique is presented by Yen and Baysal,[90] in which solutions for an oscillating cylinder are described. Typical examples of the DUT method are given by Yen and Baysal[91] and Baysal and Luo.[92]

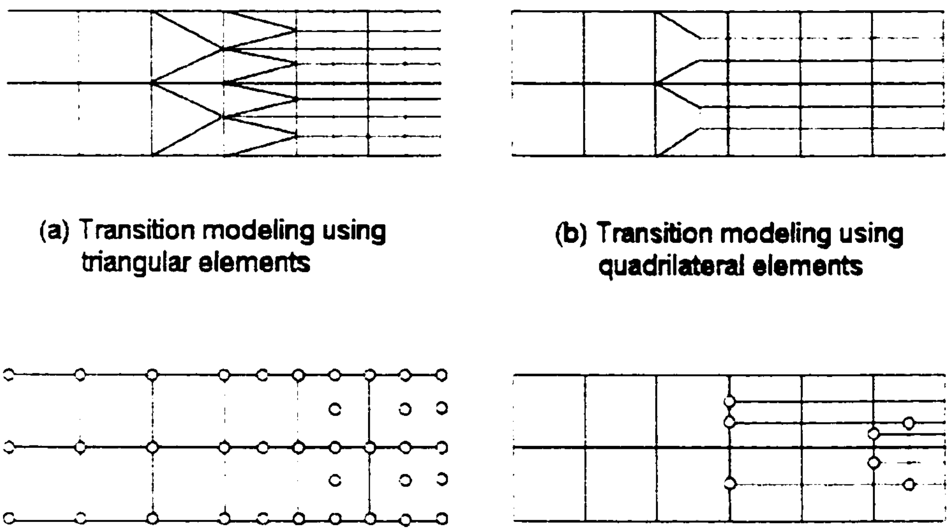
1.3 Objective and Scope of Present Research

The overall objective of this research is to develop a methodology for obtaining curvilinear interfaces between independently modeled and dissimilar two-dimensional or three-dimensional finite-element models. Specific goals of this research include the following:

1. Develop techniques for treating independently modeled subdomains with possibly non-coincident regions or interface regions having deviations in interface-coordinate locations.
2. Formulate and implement a parametric representation for general one-dimensional space curves and for two-dimensional surfaces, including a 360-degree enclosure.
3. Develop methodology for the projection of the subdomain interface nodes to a common interface-geometry surface.
4. Develop computational algorithms for evaluating displacement and traction-constraint surface integrals.
5. Demonstrate the interface methodology on selected representative geometries that model structural applications.

1.4 Organization

The dissertation is organized as follows. Chapter I presents a motivating overview of the background and the need for interface methodology and gives a historical general and specific review of related research conducted by other investigators. Chapter II presents a general formulation of the interface-element theory based on a hybrid variational principle. This chapter also includes a discussion of the steps followed in the solution process. Chapter III gives a formulation of the theory, original to this research, used to interface and automatically re-model independently modeled subdomains. Chapter IV presents a detailed discussion of the methodology for evaluating sub-matrices that couple two or more domains and the assembly of these sub-matrices into the global generalized stiffness matrix. Chapter IV concludes with a discussion of the solution methodology for the generalized stiffness matrix. Chapter V presents and interprets results from selected cases obtained by applying the methodology. Finally, Chapter VI presents conclusions drawn from the research and gives recommendations for future investigation.



(a) Transition modeling using triangular elements

(b) Transition modeling using quadrilateral elements

(c) Transition modeling using variable-order elements

(d) Transition modeling using constraints

Figure 1.1 Commonly used transition-modeling strategies

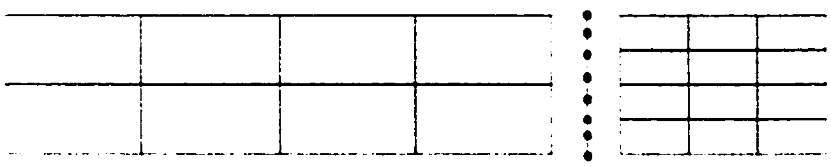
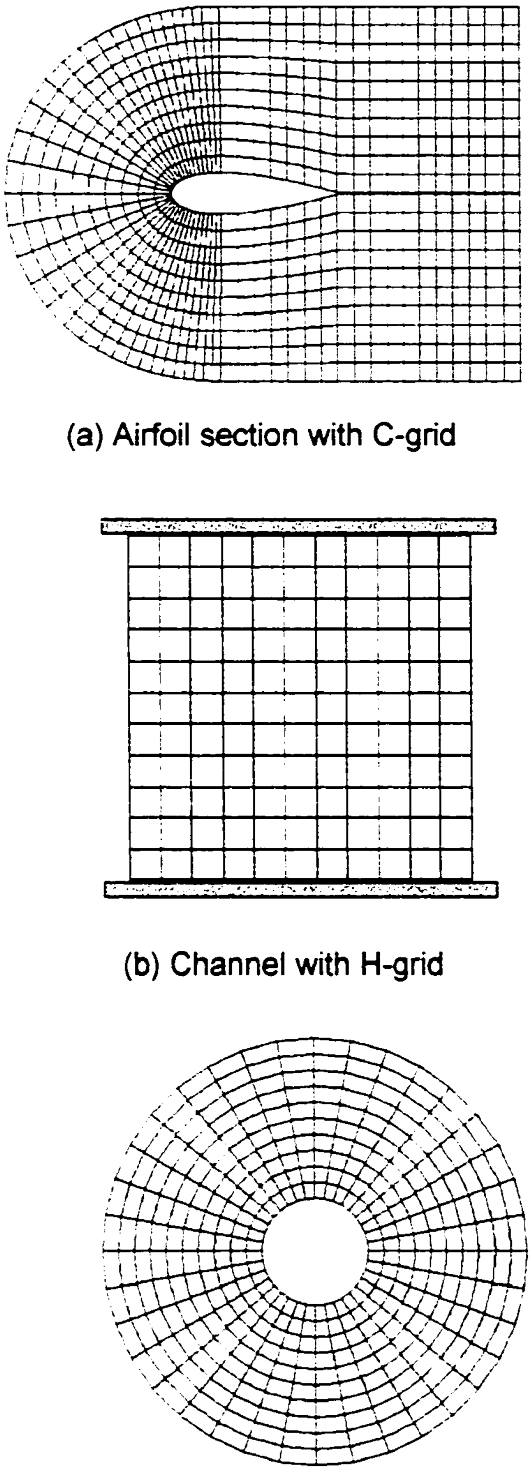


Figure 1.2 Interfacing dissimilar finite-element meshes

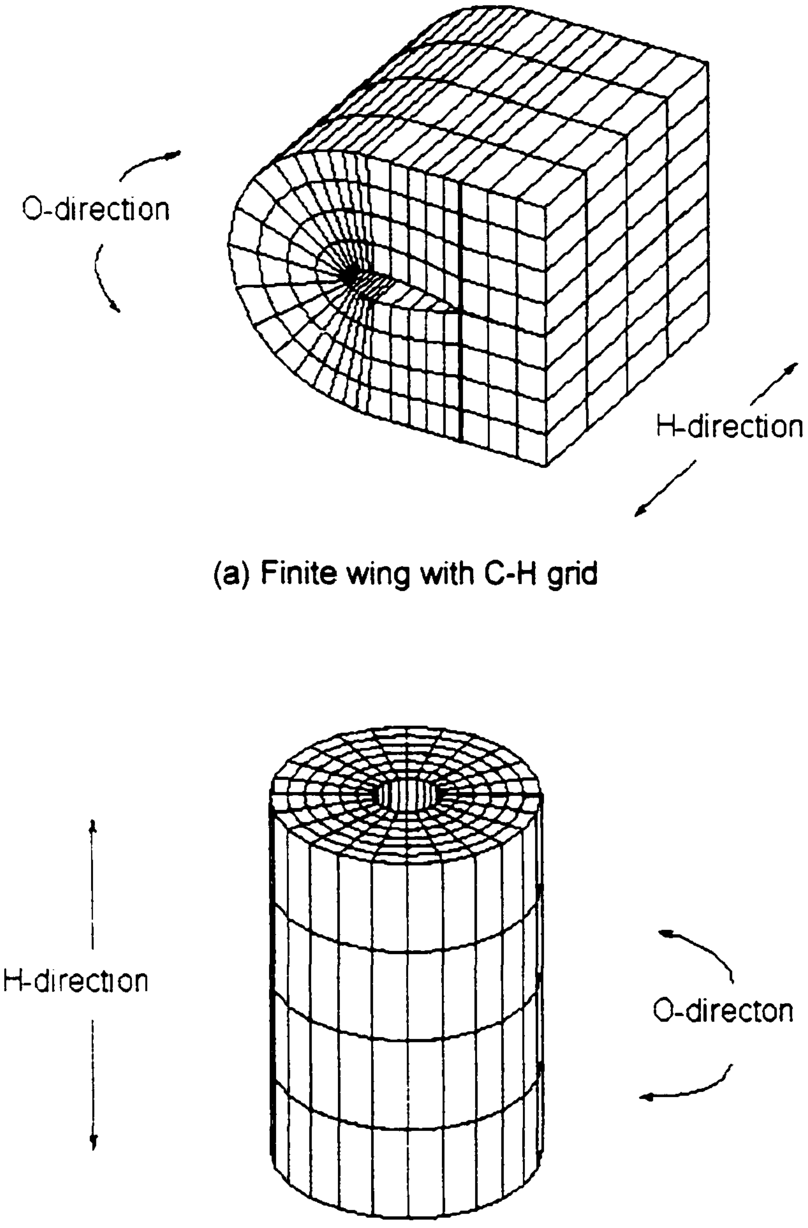


(a) Airfoil section with C-grid

(b) Channel with H-grid

(c) Infinite cylinder with O-grid

Figure 1.3 Example nomenclature for two-dimensional CFD grid topologies.



(a) Finite wing with C-H grid

(b) Cylinder with O-H grid

Figure 1.4 Example nomenclature for three-dimensional CFD grid topologies.

CHAPTER II

THEORETICAL FORMULATION

2.1 Introduction

Early formulations of finite element methods leaned strongly toward physically based reasoning and the direct stiffness method with assumed displacements (e.g., Turner et al.[20]). Current formulations are usually based on a variational statement from solid mechanics or on a weighted-residual method in general field problems (e.g., Cook et al.,[25] Reddy,[29] Huebner and Thornton[28]). For structural formulations, there is an assortment of approaches that can be used to obtain variational equations (e.g., Shames and Dym[11]). The approach used in the present research is based on the method of minimum total potential energy together with Lagrange multipliers to enforce integrated traction and displacement constraints across the interface between multiple subdomains. The present formulation belongs to a family of methods generally called *hybrid variational methods*. Examples of this approach have been discussed at various levels of detail by Zienkiewicz and Taylor,[93] Aminpour et al.,[55] Ransom et al.,[56] and Housner et al.[62] A discussion of the previous formulation and extensions for the present research are given in this chapter.

2.2 Variational Formulation

Consider a finite-element model of some generic structure with the finite-element model consisting of at least two subdomains, Ω_1 and Ω_2, whose union comprises the model for the structure as shown in Figure 2.1. If the subdomains fit together exactly and have coincident nodes where the subdomains touch, and if the displacement approximations along this interface are identical, then the models are compatible and there is no need for an interface. Traditional substructuring follows this approach. If the nodes are not coincident at the interface as shown in Figure 2.2, then maintaining continuity of displacements requires special treatment. Within each subdomain, the principle of minimum total potential energy is used. Continuity of displacements along the interface surface between the dissimilar finite-element models is enforced in an integral sense using Lagrange multipliers. By imposition of the stationary conditions on this constrained variational statement, the subsidiary conditions define the Lagrange multipliers to be surface tractions.

If the subdomain geometries do not mate identically (as illustrated for example in Figure 2.3), and then a common geometrical interface must be defined. Such geometry errors or mismatches may occur due to different discrete surface representations, subdomain modeling flexibility, or finite precision in the geometric modeler. It is assumed that gaps between the *models* are a purely modeling phenomenon and do not represent gaps in the *physical structure* itself as such gaps would need to be treated with a different approach. It is further assumed, for the present, that an interface surface has been independently defined and interface nodes from both domains have been adjusted to lie on the common mathematical surface of the interface as shown in Figure 2.4. Detailed discussion of the methodology to define a parametric representation for a common interface surface is given in Chapter III.

If all externally applied loads (body forces, surface tractions, and point loads) on an elastic structure are conservative, then the total work done on the structure during a virtual displacement can be written as a potential energy function. Assuming no initial stresses or strains and that only two subdomains are involved, the total potential energy for each subdomain is expressed as

$$\Pi_1 = \int_{\Omega_1} \tfrac{1}{2}\{\varepsilon_1\}^T [\mathbf{E}_1]\{\varepsilon_1\}dV - \int_{\Omega_1}\{\mathbf{u}_1\}^T[\mathbf{F}_1]dV - \int_{S_1}\{\mathbf{u}_1\}^T\{\Phi_1\}dS - \{\mathbf{D}_1\}^T\{\mathbf{P}_1\} \qquad (2.1)$$

$$\Pi_2 = \int_{\Omega_2} \tfrac{1}{2}\{\varepsilon_2\}^T [\mathbf{E}_2]\{\varepsilon_2\}dV - \int_{\Omega_2}\{\mathbf{u}_2\}^T[\mathbf{F}_2]dV - \int_{S_2}\{\mathbf{u}_2\}^T\{\Phi_2\}dS - \{\mathbf{D}_2\}^T\{\mathbf{P}_2\}. \qquad (2.2)$$

The integrations on Ω_i are volume integrals over the subdomain volumes, whereas integrations over S_j are surface integrals over the subdomain surfaces (i.e., the bounding surfaces of Ω_i). The connection between the subdomains is obtained by enforcing displacement continuity between the subdomains through a common interface surface. The displacement field on the interface surface is defined as {v}, and displacement continuity along the interface is achieved when {v} - {u$_i$} = {0} for each subdomain, i = 1,2.

To enforce the constraints expressed by the equation {v} - {u$_i$} = {0}, (i = 1,2) in an integral sense, Lagrange multipliers are introduced and constraint integrals are added to the total potential energy of the configuration resulting in the following functional to be minimized:

$$\Pi = \Pi_1 + \Pi_2 + \int_{S_0} \{\lambda_1\}^T (\{v\} - \{u_1\}) dS + \int_{S_0} \{\lambda_2\}^T (\{v\} - \{u_2\}) dS. \qquad (2.3)$$

A mathematical statement of the hybrid variational formulation for two subdomains connected along a single interface is given by equation (2.3). This relation is generalized to include NS subdomains and NI interfaces where each interface boundary involves a list of NIS subdomains connected to each interface. Hence, a general form of equation (2.3) is given by

$$\Pi = \sum_{j=1}^{NS} \Pi_j + \sum_{i=1}^{NI} \sum_{n=1}^{NIS(i)} \int_{(S_0)_i} \{\lambda_{k(n)}\}^T (\{v_i\} - \{u_{k(n)}\}) dS, \qquad (2.4)$$

where k(n) is the index array connecting the nth subdomain to the ith interface. In the present discussion involving two subdomains and one interface, NS = 2, NI = 1, NIS(i) = 2, and the corresponding index array k(n) contains values {1,2}.

Implicit in equation (2.3) is the fact that Π_1 and Π_2 represent the total potential energies for their respective subdomains rather than element-level values as in the traditional variational statement. This difference is due to the presence of the interface and the imposition of constraints along it wherein the interface involves multiple element surfaces along each subdomain. Hence, terms contained in Π_j represent global, assembled quantities instead of element matrices and vectors. In equation (2.3), $\{\lambda_j\}$ represents the vectors of Lagrange multipliers for each domain and S$_0$ denotes the common parametrically defined interface surface along which subdomains 1 and 2 are connected. This modified form of the principle of minimum potential energy involves three independent unknown fields: the displacement field within the subdomains, the displacement field along the interface, and the tractions along the interface. As such, a three-field hybrid variational statement is involved rather than the usual single-field principle of minimum total potential energy. The modified functional of equation (2.3), a specialized form of equation (2.4) for two subdomains and one interface, forms the basis of a variational procedure for obtaining the generalized finite-element formulation of the equations which will be solved in the present work. Derivation of the finite-element equations from the variational statement given by equation (2.3) is discussed next.

2.3 Finite-Element Derivation

To obtain a finite-element representation from the modified total potential energy functional requires three basic steps. First, the strain field is expressed in terms of the displacement field through the strain-displacement relations. Second, all field variables, including displacements, are approximated from nodal degrees-of-freedom with suitable shape functions. Third, the approximate field variables are substituted into equation (2.3), and the first variation of the modified total potential energy functional with respect to the various degrees-of-freedom (dof) is obtained and set to zero (i.e., stationary conditions are imposed). This process results in a system of linear algebraic equations, the solution of which provides the nodal values for each degree of freedom. For this hybrid variational statement, the stationary condition gives rise to three sets of equations due to the three fields being used.

2.3.1 The Strain Field

Expressions for the linear strain field in terms of displacement are found in standard textbooks on elasticity (e.g., Timoshenko and Goodier[94]). For a three-dimensional solid, the strain-displacement relations are generally written as,

$$\varepsilon_x = \frac{\partial u}{\partial x} \qquad (2.5)$$

$$\varepsilon_y = \frac{\partial v}{\partial y} \qquad (2.6)$$

$$\varepsilon_z = \frac{\partial w}{\partial z} \qquad (2.7)$$

$$\gamma_{xy} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \qquad (2.8)$$

$$\gamma_{xz} = \frac{\partial u}{\partial z} + \frac{\partial w}{\partial x} \qquad (2.9)$$

$$\gamma_{yz} = \frac{\partial v}{\partial z} + \frac{\partial w}{\partial y} \ . \qquad (2.10)$$

For finite-element derivations, equations (2.5) - (2.10) are more usefully written in a compact form using a linear differential operator matrix, [∂], as,

$$\{\varepsilon\} = [\partial]\{\mathbf{u}\}, \qquad (2.11)$$

where the symbols in equation (2.11) have the following definitions:

$$\{\varepsilon\} = \{\varepsilon_x, \varepsilon_y, \varepsilon_z, \gamma_{xy}, \gamma_{xz}, \gamma_{yz}\}^T, \qquad (2.12)$$

$$\{\mathbf{u}\} = \{u, v, w\}^T, \qquad (2.13)$$

and

$$[\partial]^T = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} & 0 \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & 0 & \frac{\partial}{\partial z} \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial y} \end{bmatrix} . \qquad (2.14)$$

For two-dimensional plane elasticity problems (assuming the x-y plane), only two displacement fields are present (namely u and v) and only three strains are nonzero: ε_x, ε_y, and γ_{xy}. The linear differential operator in this case is given by

$$[\partial]=\begin{bmatrix}\frac{\partial}{\partial x} & 0\\ 0 & \frac{\partial}{\partial y}\\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x}\end{bmatrix}. \tag{2.15}$$

The next step is to express all unknown field variables as interpolations on the degrees-of-freedom using suitable shape functions.

2.3.2 Field-Variable Approximations

The displacement field within the j^{th} domain, { $\mathbf{u}_j$ }, is written in terms of the nodal dof, $\{q_j\}$, using standard element shape functions, $[\mathbf{N}_j]$, as

$$\{\mathbf{u}_j\}=[\mathbf{N}_j]\{\mathbf{q}_j\}, \tag{2.16}$$

where the subscript j denotes the subdomain and no summation is implied. Expressions for the Lagrange multiplier vectors, $\{\lambda_j\}$, and the interface displacement field, { $\mathbf{v}$ }, are given by

$$\{\lambda_j\}=[\mathbf{R}_j]\{\alpha_j\}, \tag{2.17}$$

and

$$\{\mathbf{v}\}=[\mathbf{T}]\{\mathbf{q}_s\}, \tag{2.18}$$

where $[\mathbf{R}_j]$ is a global matrix of polynomial shape functions one order less than the polynomials used for the displacement shape functions $[\mathbf{N}_j]$, and $[\mathbf{T}]$ is a matrix of interpolation functions for defining the interface geometry surface and are also used as shape functions for the displacement along the interface. For consistency, the shape functions in $[\mathbf{R}_j]$ for the Lagrange multipliers should be one order lower that those for [$\mathbf{N}_j$] because they are identified as tractions. Tractions are related to the stresses that are obtained from the displacements by differentiation to obtain strains and multiplied together with the constitutive equations representing material properties. Next, a three-field hybrid variational statement is used with these approximations to derive the resulting finite element equations.

2.3.3 Three-Field Hybrid Finite Element Model

By substituting equations (2.16) - (2.18) into equation (2.3), the following expression for the hybrid variational functional is obtained:

$$\begin{aligned}\Pi &= \tfrac{1}{2}\{q_1\}^T[K_1]\{q_1\}-\{q_1\}^T\{P_1\}+\tfrac{1}{2}\{q_2\}^T[K_2]\{q_2\}-\{q_2\}^T\{P_2\} \\ &+\{\alpha_1\}^T[M_1]^T\{q_1\}+\{\alpha_2\}^T[M_2]^T\{q_2\}+\{\alpha_1\}^T[G_1]^T\{q_s\}+\{\alpha_2\}^T[G_2]^T\{q_s\}, \qquad (2.19)\end{aligned}$$

where {$\mathbf{q}_j$} represents the assembled global dof vector for subdomain j, [$\mathbf{K}_j$] represents the assembled global stiffness matrix for subdomain j, and { $\mathbf{P}_j$ } represents nodal boundary conditions in the form of applied loads or displacements, but not both on the same dof at the same time. The vector, {$\mathbf{q}_s$}, represents the global dof for the interface, and $\{\alpha_j\}$ represents the global traction vector for domain j. The matrices [$\mathbf{M}_j$] and [$\mathbf{G}_j$] for subdomain j have the following integral definitions (repeated subscripts do not imply summation):

$$\left[\mathbf{M}_j\right] = -\int_{S_\sigma}\left[\mathbf{N}_j\right]^T\left[\mathbf{R}_j\right]dS \qquad (2.20)$$

and

$$\left[\mathbf{G}_j\right] = \int_{S_\sigma}[\mathbf{T}]^T\left[\mathbf{R}_j\right]dS. \qquad (2.21)$$

The hybrid variational functional of equation (2.19) was minimized by applying the first variation with respect to each dof and then setting the result equal to zero. This process resulted in the following system of linear algebraic equations for the unknown degrees-of-freedom:

$$\begin{bmatrix} [\mathbf{K}_1] & [0] & [0] & [\mathbf{M}_1] & [0] \\ [0] & [\mathbf{K}_2] & [0] & [0] & [\mathbf{M}_2] \\ [0] & [0] & [0] & [\mathbf{G}_1] & [\mathbf{G}_2] \\ [\mathbf{M}_1]^T & [0] & [\mathbf{G}_1]^T & [0] & [0] \\ [0] & [\mathbf{M}_2]^T & [\mathbf{G}_2]^T & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{\mathbf{q}_1\} \\ \{\mathbf{q}_2\} \\ \{\mathbf{q}_s\} \\ \{\alpha_1\} \\ \{\alpha_2\} \end{Bmatrix} = \begin{Bmatrix} \{\mathbf{P}_1\} \\ \{\mathbf{P}_2\} \\ \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix}. \tag{2.22}$$

The generalized system of linear algebraic equations given by equation (2.22) is the key starting point for the numerical finite-element solution of coupled subdomains by the interface-element approach. This matrix equation involves the assembled global stiffness matrices from each subdomain (the K-matrices), matrices to enforce traction continuity (the M-matrices), and matrices to couple the displacements at the interface (the G-matrices). The matrix on the left-hand-side of equation (2.22) is considered in the present work to be a "generalized global stiffness matrix", although some of the desirable features of a conventional stiffness matrix are lost. The matrix of equation (2.22) is symmetric but not positive definite and not banded. The property of non-positive-definiteness meant that conventional direct solvers generally used and available in most finite element codes, were not suitable and another approach had to be followed. For the present research, an approach that deflates the matrix by removing null rows and columns followed by Gauss elimination with full pivoting was implemented. Details of the numerical implementation for creating and solving the linear algebraic system given by equation (2.22) are provided in Chapter IV. The process of calculating the component assembled matrices and vectors given in equation (2.22); combining these component matrices and solving the resulting system constitutes the theoretical approach taken in the present research. This process is discussed next.

2.4 Solution Strategy

The approach taken in the present research for constructing and solving equation (2.22) is the time-honored method of "divide and conquer," in that the process focused on each component of equation (2.22) in a step-by-step fashion.

The solution procedure was broken down in a natural way into seven major steps.

For the present research, each of these steps was treated as a separate software module and information was passed from one step to another in the form of computer files. This approach had at least three advantages over an approach based on a single large program. First, it isolated each step and provided a well-focused, structured environment for program coding and trouble shooting. Second, it allowed the results of each step in the process to be monitored and any errors can then be corrected immediately. Third, existing code for computing the K-matrices was utilized and required only minimum modification.

These seven major steps are briefly identified in the following list, and each is discussed in detail in subsequent chapters:

1. Define the interface surface and modify the subdomains to fit the interface.
2. Compute and assemble each subdomain stiffness matrix, $[\mathbf{K}_j]$.
3. Compute and assemble each subdomain traction continuity matrix, $[\mathbf{M}_j]$.
4. Compute and assemble each subdomain displacement-coupling matrix, $[\mathbf{G}_j]$.
5. Assemble the global generalized stiffness matrix and the global boundary-condition vector, thus defining equation (2.22).
6. Solve the global generalized linear algebraic system for the nodal displacement degrees-of-freedom.
7. Recover subdomain stresses from the subdomain nodal displacement vector, $\{\mathbf{q}_j\}$.

The interface modeling methodology (Step 1) is an essential part of the research reported herein, and, therefore, Chapter III is entirely devoted to a detailed discussion of the theory concerning this part of the research. The remaining steps are discussed in Chapter IV covering the remainder of the numerical implementation.

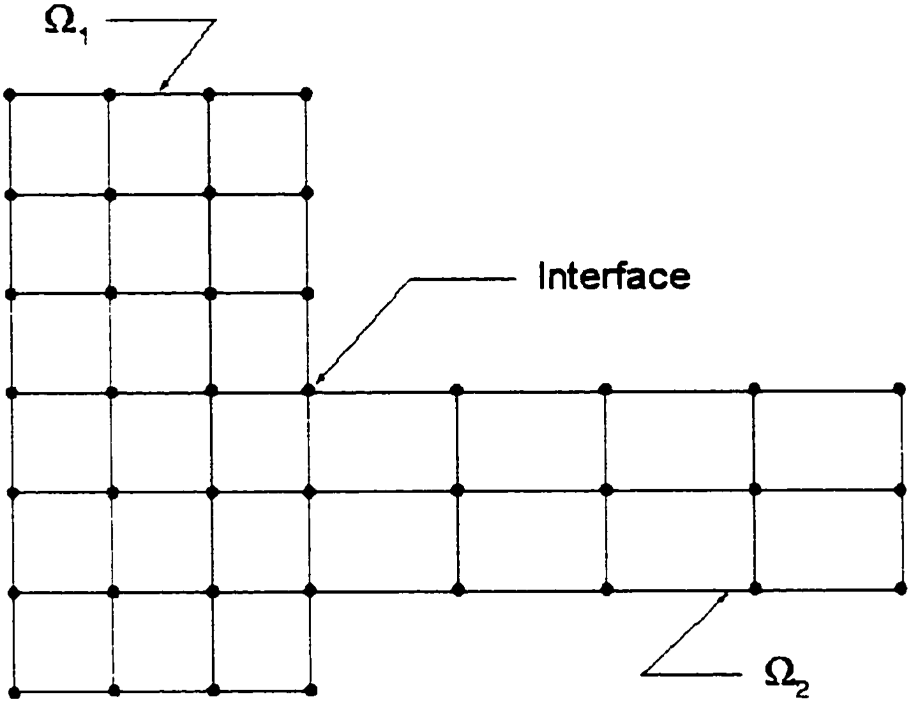


Figure 2.1 Finite element model with matching nodes at interface.

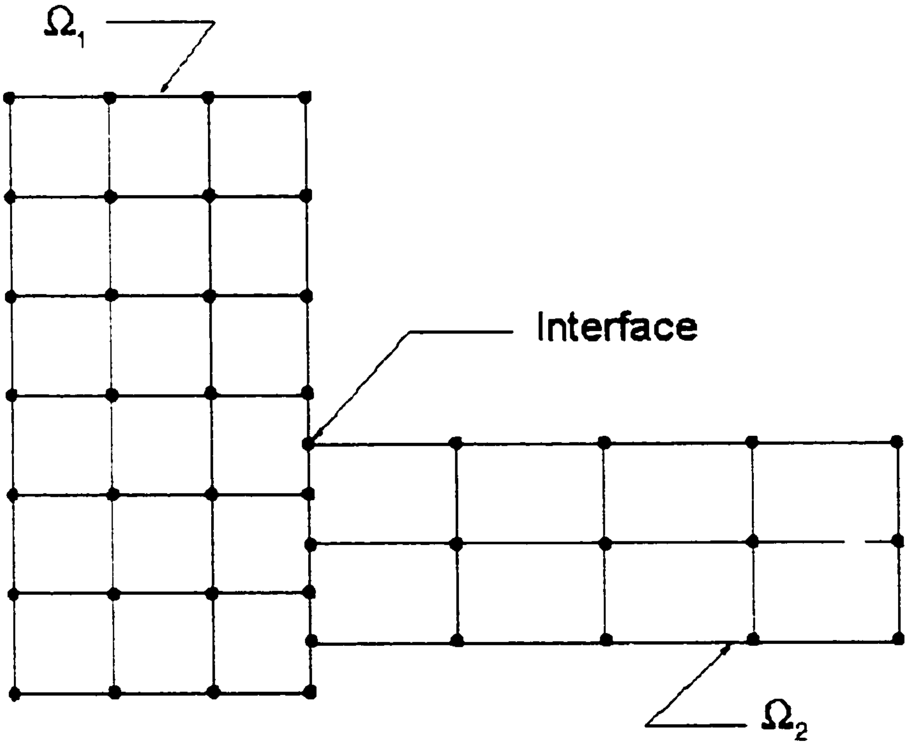


Figure 2.2 Finite element model with non-matching nodes at interface.

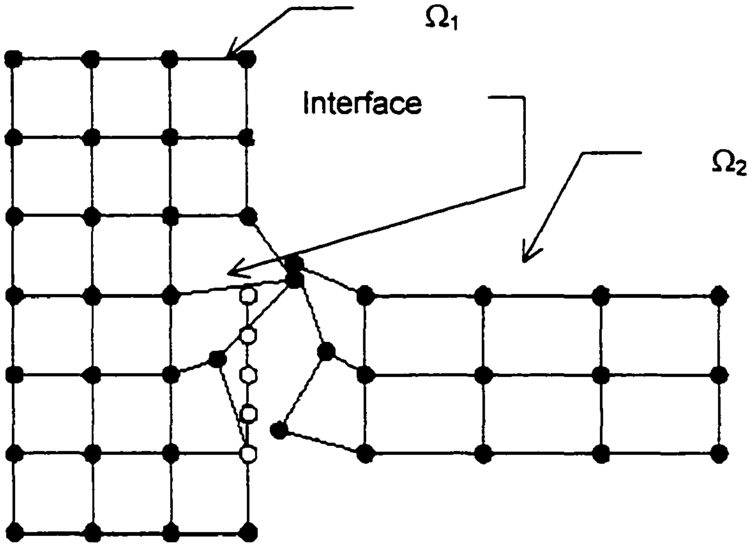


Figure 2.3 Finite element model with geometrical gaps between subdomains.

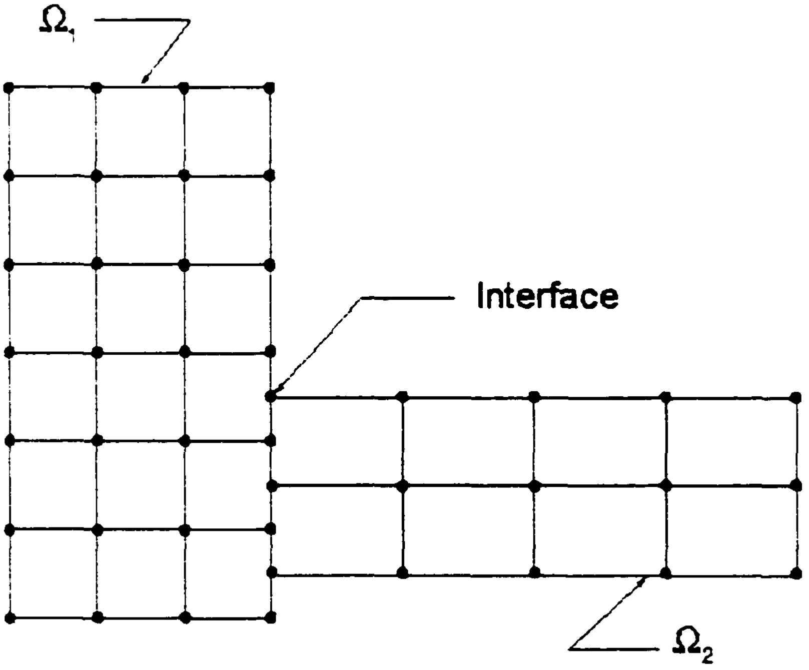


Figure 2.4 Finite element model with subdomains adjusted to a common interface.

CHAPTER III

INTERFACE MODELING

3.1 Introduction

The rationale, theory, and algorithms for modeling geometric interfaces between subdomains are discussed in this chapter. First, there is a discussion of two-dimensional plane elasticity problems having one-dimensional interfaces. Next, the methods are generalized to treat three-dimensional elasticity problems having two-dimensional interfaces.

3.2 Problem Definition

Suppose a complete structure consists of separate components or can be partitioned for modeling convenience into two or more subdomains. It is frequently of interest to create independent finite-element models of the structural subdomains. Alternatively, a single structure may have high-stress regions that require a high-resolution model, and for computational efficiency, a lower resolution model is used for the rest of the structure. When the independently modeled subdomains are combined, the interface boundaries where they are to be adjacent to each other may not fit exactly or may even exhibit small to moderately large gaps between the subdomains. For example, see Figure 3.1. A situation like this brings up several issues to be addressed and resolved. Addressing and resolving these issues forms much of the motivation of this dissertation.

First, a way to represent the interface in a mathematically accurate but convenient manner is needed. Second, the representation technique must be able to redefine the "noisy" or ill-fitting interface nodes from the subdomains in an automated way and thus minimize user intervention. Third, the representation technique must be of sufficient generality to treat realistic interface shapes that are likely to be encountered in practice. In the case of one-dimensional interfaces, the technique should be able to handle reasonable but general space curves, with plane curves treated as a specialization of space curves. In the case of two-dimensional interfaces, the technique should be able to handle full 360^{o} open surfaces such as cylinders resulting from drilled holes. Furthermore, the technique should be able to treat closed surfaces (e.g., a void in a solid). Closed surfaces are assumed to be partitioned into a small number of disjoint open surfaces, and the algorithm derived herein will treat these open surfaces.

3.3 Solution Approach: One-Dimensional Interface

Concepts to be discussed in this section are illustrated in Figure 3.2 which depicts finite-element models of two subdomains, Ω_1 and Ω_2, and a typical interface curve, S_0, for connecting the two subdomains. The geometrical definition of this interface curve is through a parametric representation of a general space curve using selected points, called interface data points (open symbols in Figure 3.2) to define the interface curve (i.e., the interface-element geometry). These defining points may be chosen arbitrarily and may consist of any convenient subset of the interface nodes from the finite-element models of either one or both subdomains or chosen in any other convenient manner. The approach taken in this research recognizes four distinct classes of points, each of which plays a role in the interface modeling. These four classes of points are designated as:

- *node points* on the interface (from a subdomain finite element model)
- *interface data points*
- *parameterization points*
- *spline breakpoints*

The *node points* are the pre-existing finite-element model nodes at the interface element location, which are to be redefined or mapped onto the mathematical interface element geometry. The *interface data points* represent a possible subset of the node points on the interface selected to define the interface surface. The *parameterization points* are the points used to define a new independent variable used to locate the interface data points. The *spline* *breakpoints* are points at which spline shape-function segments are fitted. In general, these four sets of points are inter-related but remain distinct. Each of these sets of points and the role they play in the interface modeling are discussed in detail in the subsections that follow.

3.3.1 Interface Data Points

The information presumed to define the interface curve, which is a general space curve, consists of a given set of NP coordinates in three-dimensional space: $\{x_i, y_i, z_i\}$ for i = 1,2,3, . . . NP. The given set of coordinates for defining the interface geometry is referred to in this work as the *interface data points*. Previous approaches, (e.g., Aminpour et al.[56]) assume the interface to be defined by the union of nodes comprising adjacent edges of the original finite-element model of each subdomain. No such restriction on the interface definition is presumed in the present discussion, and the term *node* is reserved for the nodes comprising the finite-element models. The nodes along the adjacent edges of the subdomain finite element models may also be used as the interface data points, but the interface is not *required* to be defined in this way. Indeed, many advantages are accrued by not choosing the nodes for the interface geometry definition. Once a set of interface points is defined, a mathematical representation based on the interface data points is obtained in order to have a useful definition of the interface for subsequent remodeling computations. It is generally not feasible to represent space curves without some kind of parameterization, and this is discussed next.

3.3.2 Parameterization and Parametric Points

For theoretical discussion of the differential geometry of general space curves, differential arc length, $ds = (dx^2 + dy^2 + dz^2)^{1/2}$, is often used for parameterization. On the other hand, for applied computational purposes, differential arc length is not the best choice because (i) it is an infinitesimal quantity, (ii) its defining equation is non-linear, and (iii) it is known only implicitly.

These characteristics result in complicated iterative algorithms for computing arc length. A better choice for computation is cumulative chord length based on the finite Euclidean distance formula between the interface data points. An example of the distinction between arc length and chord length is shown in Figure 3.3.

Parameterization based on cumulative chord length is straightforward to define and compute. The parameterization process involves three main tasks. First, define incremental chord length: $c_i = [(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2 + (z_{i+1}-z_i)^2]_{1/2}$, for i = 1,2,3, . . ., NP-1. Next, define cumulative chord length as follows: $s_1 = 0$, and $s_{i+1} = s_i + c_i$, over the same range of i. Finally define *normalized* cumulative chord length: $s_i = s_i / s_{NP}$. The set $\{s_i\}$ ranges over values from 0 to 1 and nicely characterizes the coordinate set $\{x_i, y_i, z_i\}$ in parametric form. An example of this type of curve parameterization is shown in Figure 3.4. This parameterization has the advantages of being robust and easy to compute, in contrast to the differential arc-length approach. The set of numbers comprising the set $\{s_i\}$ of normalized cumulative chord lengths between physical points along a general space curve are termed *parametric points* in the present discussion. Thus, the i^{th} discrete physical point, defined in terms of its coordinates (x_i, y_i, z_i), has a corresponding parametric point defined in terms of its normalized cumulative chord length, s_i, for each of the NP discrete points.

3.3.3 Interface Data Points, Parametric Points and Breakpoints

A natural approach to representing general space curves defined by a set of NP interface points is with cubic splines (or any spline, for that matter), treating the representation as an interpolation problem. Points at which the spline segments are joined are called ***breakpoints*** in the present discussion. Hence, for a set of NP interface data points and their corresponding parametric values in terms of normalized cumulative chord lengths, a set of NB breakpoints can be defined. These breakpoints are then used to develop a parametric representation for the general space curve. Within the present notation and terminology, direct interpolation results when the interface data points are also taken as breakpoints.

In the case of cubics, there are sufficient polynomial constants available to enforce $C^{(1)}$ continuity at each breakpoint. As such, the interpolation process generates a general space curve which passes through each interface data point and maintains slope continuity at each interface data point. If the interface data points are free of noise and are reasonably distributed, then using interface data points also as breakpoints works well as indicated by the example shown in Figure 3.5(a). On the other hand, if the interface data points are contaminated with noise, or if there is a high degree of data-point clustering, then this approach results in interpolation curves with unacceptable oscillations as shown in Figure 3.5(b). In addition, the number of cubic segments is governed by the number of data points, giving no independent control over the number of segments, and the effect of variations in the data points on the curve representation is not localized. Noise effects can be minimized to some extent by pre-smoothing the interface data points. Apart from not having control over the number of segments and lack of localized impact on the curve, pre-smoothing generally requires making arbitrary assumptions about the nature of the data.

An alternative approach has been developed that overcomes each of these disadvantages. First, a set of breakpoints is selected over which the cubic segments are defined independently of the given data points. The resulting cubic spline segments are then fit to the interface data points in a least-squares manner. As such, the interpolation process generates a general space curve, which passes through each breakpoint, maintains slope and curvature continuity at the breakpoints, and minimizes the interpolation error in a least-squares sense at each interface data point. Results using this approach are shown in Figure 3.6, wherein the interface data points are identical to those shown in Figure 3.5. The difference between the two approaches is in the choice of breakpoints and the least-squares fit for the splines, in contrast to interpolation between every interface data point. Comparing of Figures 3.5(a) and 3.6(a) leads to the conclusion that the least-squares method represents the smooth data in a very acceptable manner with far fewer breakpoints than data points, and thus fewer cubic spline segments are needed. Furthermore, examination of Figures 3.5(b) and 3.6(b) reveals that, while the traditional method gives rise to unwanted oscillations in the interpolated curve, the present method gives a quite reasonable curve. In addition to offering these practical advantages, the approach of having distinct breakpoints together with a least-squares fit of interface data points is in harmony with the variational philosophy inherent in the finite element modeling and interface theory developed in Chapter II.

The least-squares approach is more efficiently carried out when cubic basis functions (shape functions) are defined in an analogous manner to the shape-function approach used in the finite-element method. The basis functions for the parametric representation of the common interface element geometry are defined as the geometry approximations used in higher-order isoparametric elements with the breakpoints playing the role of nodes. The interplay between interface data points and breakpoints is discussed in the next section.

3.3.4 Breakpoint Distribution

For the parameterization discussed in Section 3.3.2, the set of parametric points, $\{s_i\}$, defined in terms of normalized cumulative chord length, may turn out to be highly clustered depending on the spatial distribution of the original interface data points, $\{x_i, y_i, z_i\}$. However, these parametric values are defined in such a way that the set $\{s_i\}$ will always be strictly monotonic increasing as long as the interface data points are distinct. For a given parametric set, $\{s_i\}$, i = 1, 2, 3... NP, a corresponding set of breakpoints, $\{sb_j\}$, j = 1,2,3, . . . NB, must be defined which will then give rise to a set of cubic polynomial basis functions, and these functions provide the subsequent representation of the interface curve. Two important questions must be answered in defining the set of breakpoints. First, how should the breakpoints be distributed relative to the parametric points? Second, what ratio of number-of-parametric-points to number-of-breakpoints provides the best performance or best fit? For the present research, an algorithm based on the idea of "equally populated breakpoint cells" has been developed. A breakpoint cell is the interval between breakpoints; the union of these cells thus completely covers the parametric interval, [0,1]. In mathematical terminology (e.g., James and James[95]), the breakpoints and associated cells form a *partition* for the interval [0,1]. The approach of equally-populated breakpoint cells has the virtue of clustering breakpoints in a way that reflects the clustering inherent in the interface data points themselves. An example of breakpoint definition based on this approach is illustrated in Figure 3.7.

To determine the near optimum number of interface data points needed per breakpoint cell, numerical experiments were performed using several benchmark cases. There is a natural lower bound of four interface data points per breakpoint cell, since at least four independent conditions are needed to define a cubic segment. The tests revealed that five interface data points per cell resulted in a significant improvement over four points per cell. Curves defined by four points were too "slack" and had a tendency toward spurious oscillation. Very little improvement is noted when the number of points per cell is increased above five. Beyond 8 or 10 points per cell, the accuracy of the resulting curves tended to degrade. Thus, five points per cell was chosen as a default value for the present study, although the algorithm allows other values to be selected as a user-defined option.

3.3.5 Basis Function Definition

For a given set of interface points, once corresponding breakpoints are defined, the basis functions correlated with the breakpoints are then defined. For a given number of breakpoints, say NB, a question that must be resolved is how many independent basis functions need to be defined (i.e., how many functions are required in order to span the space?). For a non-intersecting space curve with NB breakpoints, there are NB-1 subintervals along the curve. For a cubic to be uniquely defined on each subinterval implies 4(NB-1) degrees-of-freedom since each cubic has four polynomial constants to be determined. There are NB-2 interior breakpoints at which smoothness constraints may be applied to obtain a desired level of smoothness in the interface representation. A sufficient number of degrees-of-freedom are available with cubics to impose $C^{(2)}$ continuity, which usually results in acceptable smoothness for the parametric representation of the interface geometry. With NB-1 interior breakpoints, each having three imposed conditions, the final number for the degrees-of-freedom is, ndof = 4(NB-1) - 3(NB-2), or ndof = NB+2. The function space thus is of dimension NB+2 and requires NB+2 linearly independent basis functions to span it.

The individual basis functions are defined on the breakpoints in the spirit of finite-element analysis. Shown in Figure 3.8 is a sketch for the k^{th} basis function defined over five breakpoints denoted by: $\{sb_{k-2}, sb_{k-1}, sb_k, sb_{k+1}, sb_{k+2}\}$ to comprise a "5-node element." The "shape functions" are cubic segments satisfying $C^{(2)}$ continuity at the three interior breakpoints. The function is required to vanish at the boundary points, sb_{k-2} and sb_{k+2} and to take on the value of unity at the central point, sb_k. Outside the set $\{sb_{k-2}, sb_{k-1}, sb_k, sb_{k+1}, sb_{k+2}\}$, the function is taken to be identically zero. These conditions are sufficient to specify completely the basis function as a $C^{(2)}$ function over the entire open interval (-1, +1), but being non-zero only over the interval (sb_{k-2}, sb_{k+2}). This is a very convenient feature known in the mathematical literature as "compact support" (e.g., Faux and Pratt,[96] and James and James[95]). In summary for a set of NB breakpoints, $\{sb_j\}$, j = 1,2,3, . . . NB, there corresponds a function space of dimension NB+2 which is spanned by cubic basis functions having compact support.

It is convenient for both the theoretical discussion and coding efficiency to have a one-to-one correspondence between the indices of the basis functions and the indices of the breakpoints. One way to obtain this one-to-one correspondence is to extend the breakpoints at each end of the set and renumber them, as depicted in Figure 3.9. By extending the original breakpoint set at each end and renumbering in this manner, a basis function with index, k , will correspond to the center breakpoint of the k^{th} 5-point group over which the basis function is non-zero. The new breakpoint indices start at the first extended point just to the left of the original set and end at the first extended point just to the right. There will thus be NB+2 indexed points in this set and the same number of basis functions in the spanning set, { $b_i(s)$ } (i = 1, NB+2). It is observed from Figure 3.9 that further extension and function definition adds nothing, since all functions beyond $b_1(s)$ and $b_{NB+2}(s)$ make only a zero contribution to the interval defined by the original breakpoint set. However, b_1 and b_{NB+2} do make a non-zero contribution, as do all others defined between these functions. A generic expression for the kth basis function is obtained using four cubic shape functions defined piecewise over the four segments of the five-breakpoint set, {sb_k }, (k=1,2, 3, 4, 5). These four cubic shape functions are given by equations (3.1) - (3.4):

$$Y_1(s) = a_1(s - sb_{k-2})^3, \quad (sb_{k-2} \le s < sb_{k-1}) \tag{3.1}$$

$$Y_2(s) = a_1(s - sb_{k-2})^3 + a_2(s - sb_{k-1})^3, \quad (sb_{k-1} \le s < sb_k) \tag{3.2}$$

$$Y_3(s) = a_3(sb_{k+2} - s)^3 + a_4(sb_{k+1} - s)^3, \quad (sb_k \le s < sb_{k+1}) \tag{3.3}$$

$$Y_4(s) = a_3(sb_{k+2} - s)^3, \quad (sb_{k+1} \le s \le sb_{k+2}). \tag{3.4}$$

Because of the form in which the Y_i are defined, it is clear by close inspection of equations (3.1) - (3.4) that they automatically satisfy $C^{(2)}$ conditions at sb_{k-1} and sb_{k+1}. The coefficients, {a_i}, i = 1 - 4, are determined by enforcing $C^{(2)}$ continuity conditions at the kth break point and by normalizing the basis function to unity at $s = sb_k$. With these four $C^{(2)}$ functions thus defined, the generic kth basis function, b_k (s), is defined over the entire interval [sb_{k-2} , sb_{k+2}] by the union of these shape functions, i.e.,

$$b_k(s) = \begin{Bmatrix} Y_1(s), & (sb_{k-2} \le s < s_{k-1}) \\ Y_2(s), & (sb_{k-1} \le s < sb_k) \\ Y_3(s), & (sb_k \le s < sb_{k+1}) \\ Y_4(s), & (sb_{k+1} \le s \le sb_{k+2}) \end{Bmatrix}. \tag{3.5}$$

An example plot of such a basis function and its first two derivatives, defined in this way for an arbitrary set of five breakpoints, is shown in Figure 3.10. The basic features of compact support and continuity through the second derivative are clearly demonstrated in this figure. Having defined a family of NB+2 basis functions over the extended breakpoint set, the next task is to fit these basis functions to the NP interface data points. The procedure to accomplish this task is discussed in the next subsection.

3.3.6 Basis Function Representation of Interface Curve

Taking advantage of the compact-support feature, each basis function, $b_j(s)$, is treated as a function defined over the complete parametric interval, [0,1]. The x-coordinate, y-coordinate, and z-coordinate of the interface curve are then expressed as linear combinations of the basis functions to obtain a parametric mathematical representation of the curve. These representations are written as,

$$f_x(s) = \sum_{j=1}^{NB+2} \alpha_j b_j(s) \tag{3.6}$$

$$f_y(s) = \sum_{j=1}^{NB+2} \beta_j b_j(s) \tag{3.7}$$

$$f_z(s) = \sum_{j=1}^{NB+2} \gamma_j b_j(s) \tag{3.8}$$

where the coefficients, α_j, β_j, and γ_j are determined from the least-squares procedure. Each of the functions, f_x, f_y, and f_z are obtained by a similar process, so the present discussion will treat a general expression similar to equations (3.6) - (3.8) and representing any of the three coordinates. Thus, the generic equation is written as,

$$f(s) = \sum_{j=1}^{NB+2} c_j b_j(s). \tag{3.9}$$

At the k^th parametric point, s_k, the symbol ϕ_k represents the k^th value of either x, y, or z. With this convention, the sum of the squared errors between values of $f(s_k)$ and ϕ_k is defined to be

$$e = \sum_{k=1}^{NP}\left[f(s_k) - \phi_k\right]^2 . \tag{3.10}$$

Upon substituting equation (3.9) for f(s$_k$), equation (3.10) becomes

$$e = \sum_{k=1}^{NP}\left[\sum_{j=1}^{NB+2} c_j b_j(s_k) - \phi_k\right]^2 . \tag{3.11}$$

The error sum expressed by equation (3.11) is minimized following the usual least-squares method, and this process results in the following linear system, the solution of which provides the coefficients, c_j,

$$[\mathbf{A}]\{\mathbf{c}\} = \{\mathbf{r}\}, \tag{3.12}$$

where {**c**} is the vector of unknown coefficients, [**A**] is the least-squares matrix, and {**r**} is the corresponding right-hand-side vector. The entries in the coefficient vector are obviously just the c_j occurring in equation (3.11). The elements of the A-matrix and right-hand-side vector are given in equations (3.13) and (3.14), respectively,

$$A_{ij} = \sum_{k=1}^{NP} b_i(s_k) b_j(s_k) \tag{3.13}$$

and

$$r_i = \sum_{k=1}^{NP} \phi_k b_i(s_k) . \tag{3.14}$$

The coefficients obtained from three linear systems similar to equation (3.12) are used to define the general space curve in terms of the previously discussed basis functions, giving a convenient mathematical representation of the interface curve. The final task, which completes the interface model, is to project the interface-nodes from each of the original finite-element models onto the mathematical curve which is now the common interface. The projection algorithm for the one-dimensional interface is discussed in the next section.

3.3.7 One-Dimensional Projection Algorithm

The problem addressed in this subsection is illustrated in Figure 3.11 which shows a typical interface curve and two representative examples of discrete points to be projected onto the interface curve. The question to be answered is, given a general space curve and a discrete point not on the curve, where on the curve should the point be placed in order to minimize distorting the finite elements in the vicinity of the interface? The approach taken in the present research was to select the point on the curve closest to the discrete point consistent with any user-defined movement constraints. This approach worked well for the representative cases studied. The algorithm used a form of the Newton-Raphson iteration to minimize the Euclidean distance function expressed as

$$D = \sqrt{[f_x(s) - x_0]^2 + [f_y(s) - y_0]^2 + [f_z(s) - z_0]^2}\,, \tag{3.15}$$

where x_0, y_0, and z_0 are the coordinates of the discrete point to be projected, and f_x, f_y, and f_z are the functions defining the interface curve as in equations (3.6) - (3.8).

3.4 Solution Approach: Extensions for Two-Dimensional Interface

An illustration for the two-dimensional case is given in Figure 3.12, which shows two independently modeled three-dimensional finite-element subdomains and an interface surface for connecting these two subdomains. As in the one-dimensional case, points defining the interface surface are called interface data points, and the geometrical definition of this interface surface is through a parametric representation similar to that used for a single independent variable, but extended to two independent variables. The geometrical interface surface is assumed to be open and to have no self-intersections so that the surface Jacobian determinant is well-defined and did not change sign on the surface.

It is further assumed that the set of interface data points is not randomly arranged but instead possessed an i-j indexing structure. Requiring the two-dimensional interface data points to have this level of indexing structure allowed the extended methodology to build upon the one-dimensional procedures. This beneficial leveraging is accomplished by means of a tensor product representation for the interface surface based on independent one-dimensional parameterizations in the i- and j-directions. Since the two-dimensional interface theory is built upon the one-dimensional methods, the entire theory discussed in Section 3.3 is not recapitulated in the presentation for the two-dimensional case, but only the necessary extensions are described in detail.

3.4.1 Interface Data Points

The information needed to define the interface surface consists of a given set of NPI · NPJ coordinates in three-dimensional space: $\{x_{ij}, y_{ij}, z_{ij}\}$ for i = 1,2,3, . . ., NPI, and j = 1,2,3, . . ., NPJ. As for the one-dimensional case, these points are referred to as *interface data points.*

3.4.2 Parameterization

The parameterization is based on cumulative normalized chord lengths in the i-direction and in the j-direction in a manner entirely analogous to the one-dimensional case. A complication arises in that there is not a unique parameterization except in fortuitous situations. For example, when the parameters were computed in the i-direction (which is defined as the s-parameterization), there is, in general, a different set of parametric points for each j-value. Similarly, when parameters in the j-direction were computed (which is defined as the t-parameterization), there is a different set of parametric points for each i-value. A unique parameterization is required for successful implementation of the procedure, and the approach followed for the present research is to define averaged chord-length parameters. Thus, for the s-parameter, the parameters were computed for a fixed i, s-parameters were computed for every j-value and then averaged over the j-index (NPJ values were averaged). A similar process is followed for the t-parameterization, averaging in the i-direction over NPI values.

3.4.3 Breakpoints

The one-dimensional breakpoint selection algorithm, previously discussed, is applied in each of the two parametric directions resulting in a net of rectangular cells in the s-t parameter space. Since the default cell population is five interface data points per cell in the one-dimensional case, the analogous two-dimensional cells contained twenty-five default interface data points per cell. The breakpoint net is thus an NBI ‹ NBJ set consisting of the tensor product of two one-dimensional breakpoint sets.

3.4.4 Two-Dimensional Basis Functions

The breakpoint set in the i (or equivalently, s) direction, having NBI points, is extended, and a space of one-dimensional basis functions of dimension NBI+2 is defined as previously discussed in Section 3.3.5. This set of basis functions is denoted in the present discussion by: $\{bs_i(s)\}$, for i = 1,2,3, . . . , NBI+2. The breakpoint set in the j (or t) direction resulted in another, independent space of basis functions of dimension NBJ+2. This second set of basis functions is denoted in the present discussion by: $\{bt_j(t)\}$, for j = 1,2,3, . . . , NBJ+2. To obtain a set of basis functions for representing the two-dimensional interface surface, an indexing algorithm is first defined that assigned a one-dimensional index, k, running through the entire tensor product of basis-function indices, i. e., k = 1,2,3, . . ., (NBI+2) ‹ (NBJ+2). For a given k-value this algorithm returned a unique pair, (i,j), which indicated the correct pair of basis functions, $bs_{i(k)}(s)$, and $bt_{j(k)}(t)$. A new, two-dimensional space of basis functions of dimension (NBI+2) ‹ (NBJ+2) is then defined by the product of these functions.This definition is expressed as

$$btp_k(s,t) = bs_{i(k)}(s)bt_{j(k)}(t) \tag{3.16}$$

for k = 1,2,3, . . . , (NBI+2) ‹ (NBJ+2). This set of basis functions is used to form a mathematical representation of the interface surface using the least-squares approach.

3.4.5 Basis Function Representation of Interface Surface

The coordinates of arbitrary points on the interface surface were written in a representation similar to equations (3.6) - (3.9), but using the tensor-product basis functions as defined in Section 3.4.4. Using the defining equation (3.16), a generic functional form for the surface coordinates is written as

$$f(s,t) = \sum_{k=1}^{(NBI+2)X(NBJ+2)} c_k \, btp_k(s,t). \tag{3.17}$$

Following a least-squares procedure similar to that discussed in the one-dimensional case, a system of linear algebraic equations similar to equation (3.12) is obtained

$$[\mathbf{A}]\{\mathbf{c}\} = \{\mathbf{r}\}, \tag{3.18}$$

where {**c**} again represents the vector of unknown coefficients, [**A**] is the least-squares matrix, and {**r**} is the corresponding right-hand-side vector. The linear algebraic system of equations (3.18) is generally of a much higher dimension than that of equation (3.12). In addition, the A-matrix and right-hand-side vector have extended definitions based on equations (3.17) and (3.18). In particular, the elements, $A_{m\ n}$ of the coefficients matrix, [**A**], are given by

$$A_{mn} = \sum_{k=1}^{(NBI+2)X(NBJ+2)} bs_{i(m)}(s_k) bt_{j(m)}(t_k) bs_{i(n)}(s_k) bt_{j(n)}(t_k), \tag{3.19}$$

and the elements of the right-hand-side vector, {**r**}, are given by

$$r_m = \sum_{k=1}^{(NBI+2)X(NBJ+2)} bs_{i(m)}(s_k) bt_{j(m)}(t_k) \phi_{i(k)j(k)}. \tag{3.20}$$

3.5 Representative Modeling Examples

To illustrate the methodology discussed in this chapter, two representative examples were considered. The first example is a two-domain model of a quadrant from a two-dimensional plate with central circular hole as shown in Figure 3.13. The region in the immediate vicinity of the hole is modeled using a circular topology, whereas the remainder of the plate (L-shaped region) is modeled with rectangular topology. Each of these separate models is straightforward and convenient to generate independently, as they followed the natural topology of their respective physical regions. Modeling each part in this way, however, left a substantial, unmodeled gap between the two models. Although no competent engineer would create a model having a gap like this, it is conceivable that it might result from an automated computational procedure. Alternatively, it is so convenient to follow the local topology in modeling, if a way existed to conveniently blend two or more such models, it wold be a worthwhile gain in ease of modeling.

For this case, the nodes on the outer ring of the circular model were selected as the interface data points. Results from applying the interface modeling procedure to this case are given in Figure 3.13(b). The gap has been eliminated, and nodes adjacent to the interface from each model have been re-defined to lie on the interface. This case showed minimal distortion of the inner circular model with some distortion of the elements from the L-shaped part near the interface. Distortion of either domain is controllable by a choice of the interface location and definition, and for this case it is anticipated for physical reasons that stress gradients would dominate in the vicinity of the hole, so distortion is controlled in the circular part. Actual effects of interface location on the solution for this case are presented and examined in Chapter V which gives solution results for several representative applications.

The second example illustrates the three-dimensional procedure (two-dimensional interface) using a solid plate with circular hole as shown in Figure 3.14. This three-dimensional case is equivalent to the case just discussed. Figure 3.14(a) illustrates the original independently modeled subdomains of circular and rectangular topology together with a circular interface. Figure 3.14(b) shows the domains after re-modeling with the interface modeling procedure. For this case, the interface is defined with independent interface data points as illustrated in Figure 3.14(a). The interface is defined using circular topology to again minimize distortion of the inner region. The interface is, however, defined with more points and thus greater resolution than is available by using nodes from the original models, which is an additional capability of the new methodology.

The third example, shown in Figure 4.15, is a solid plate with hole similar to example two, except that the interface definition has curvature in more than one direction. Figure 4.15(a) gives the L-shaped part of the model and the interface before automatic remodeling, and Figure 4.15(b) shows it after remodeling.

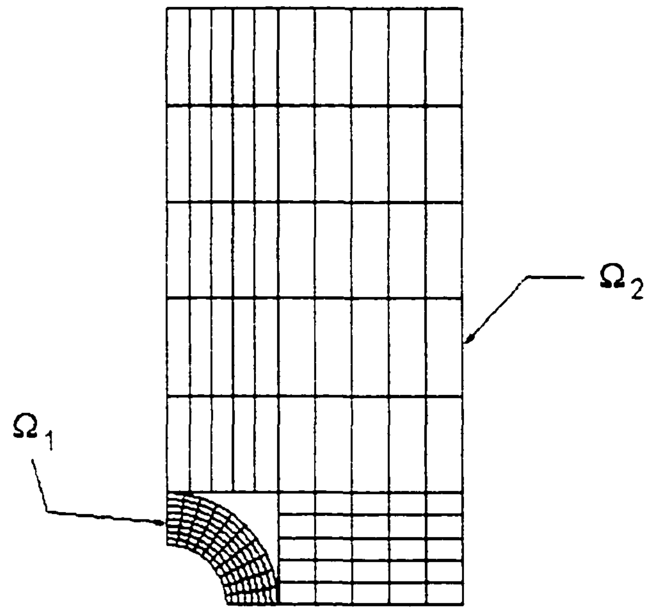


Figure 3.1 Independently-modeled, non-coincident subdomains.

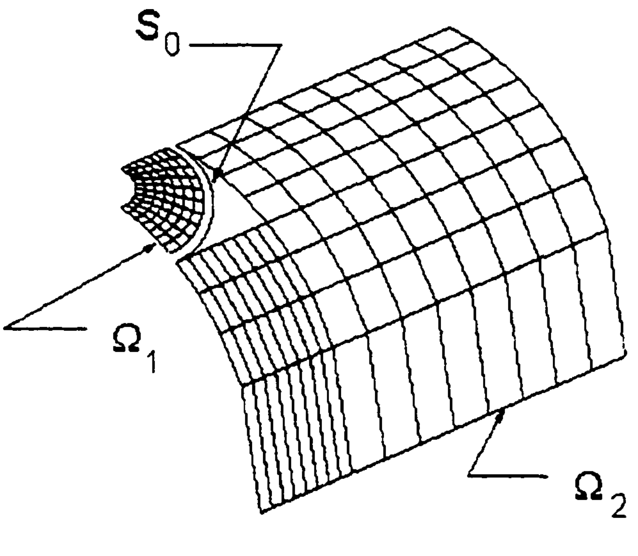


Figure 3.2 Relation between subdomains and interface definition.

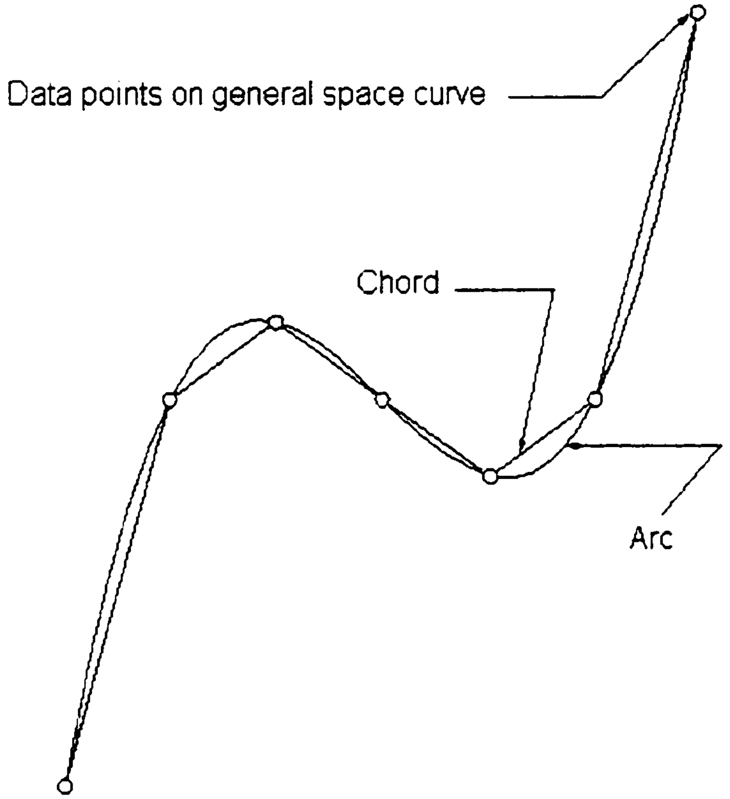
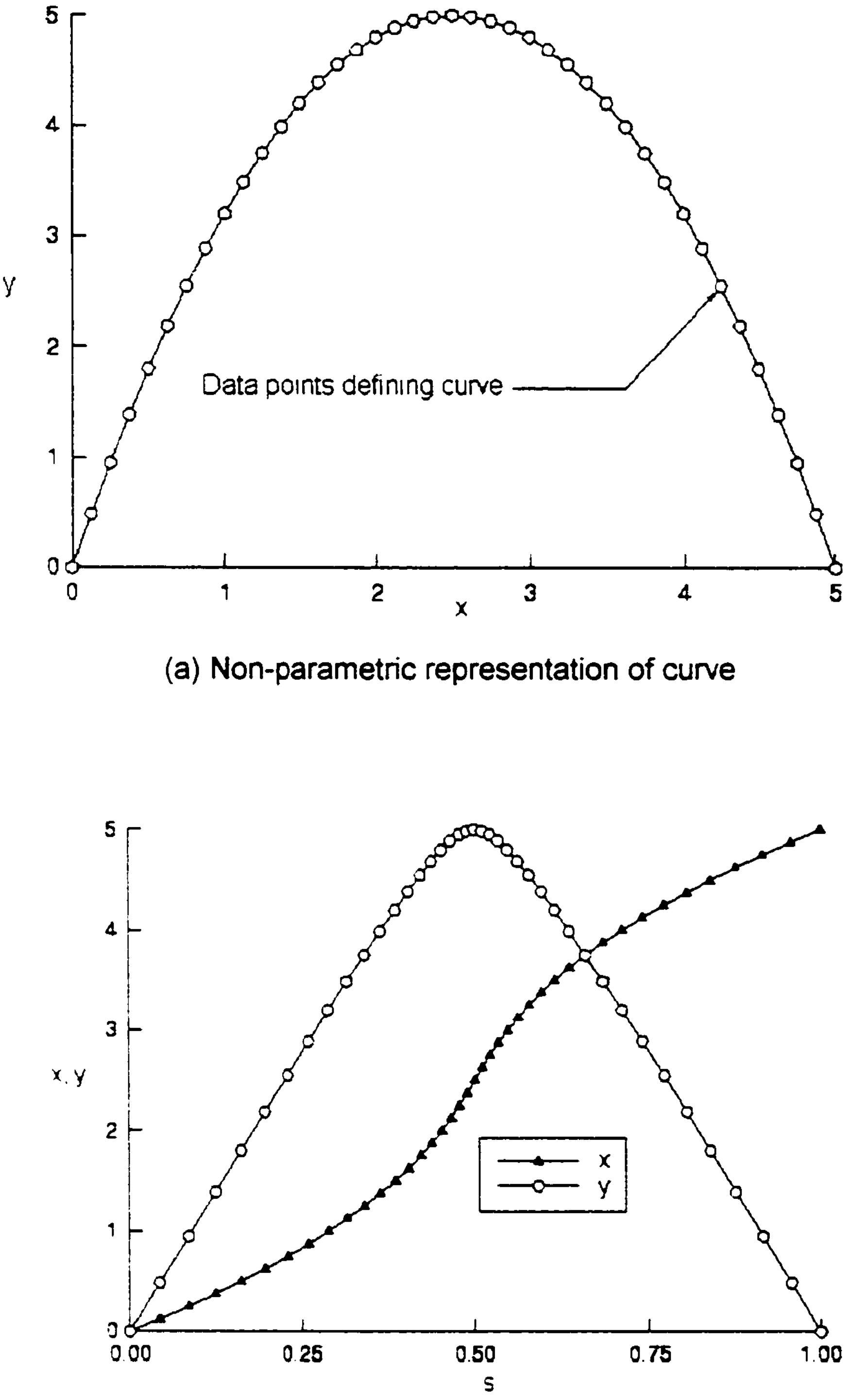


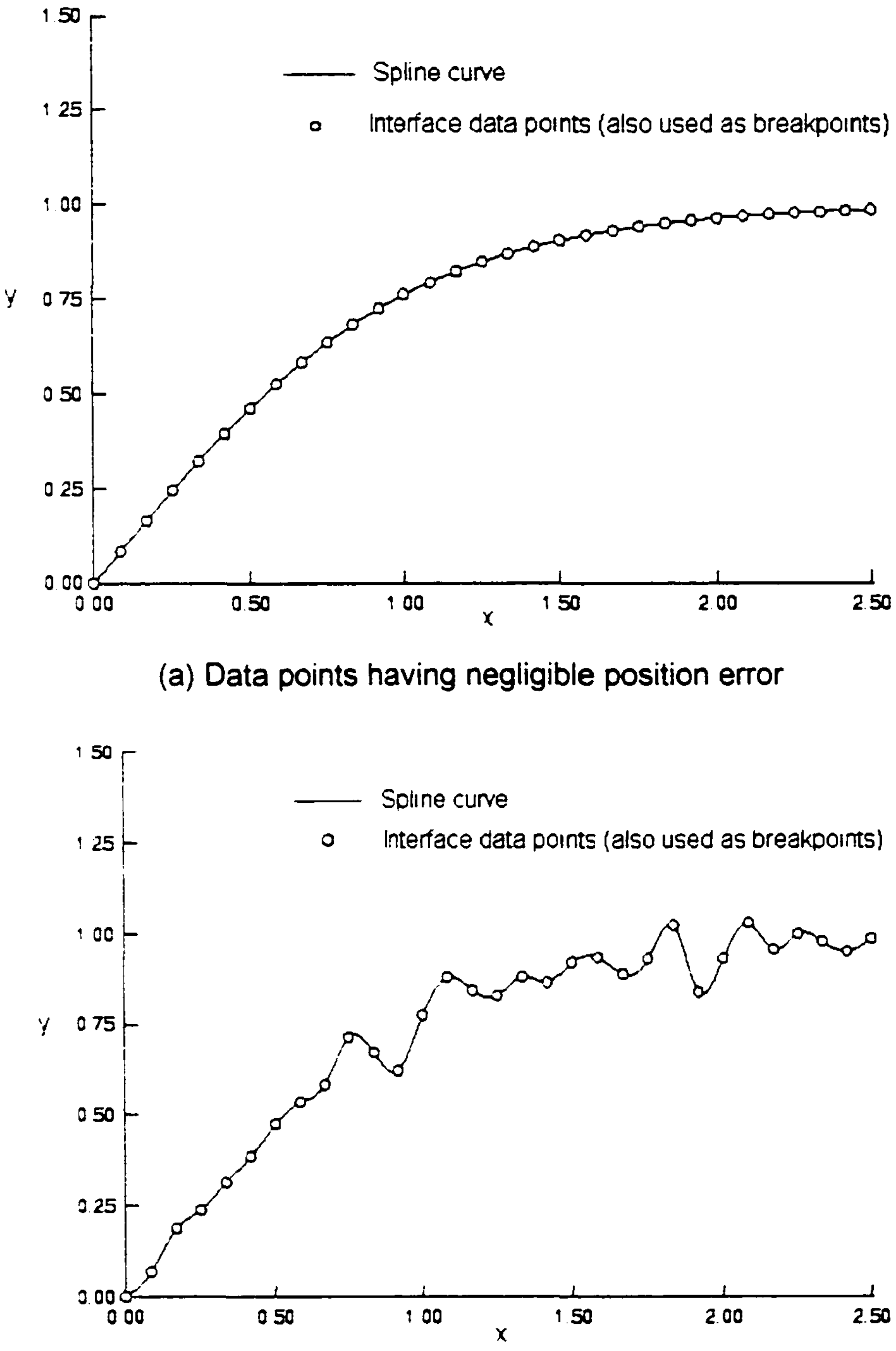
Figure 3.3 Distinction between arc-length and chord-length.



(a) Non-parametric representation of curve

(b) Parametric representation of curve

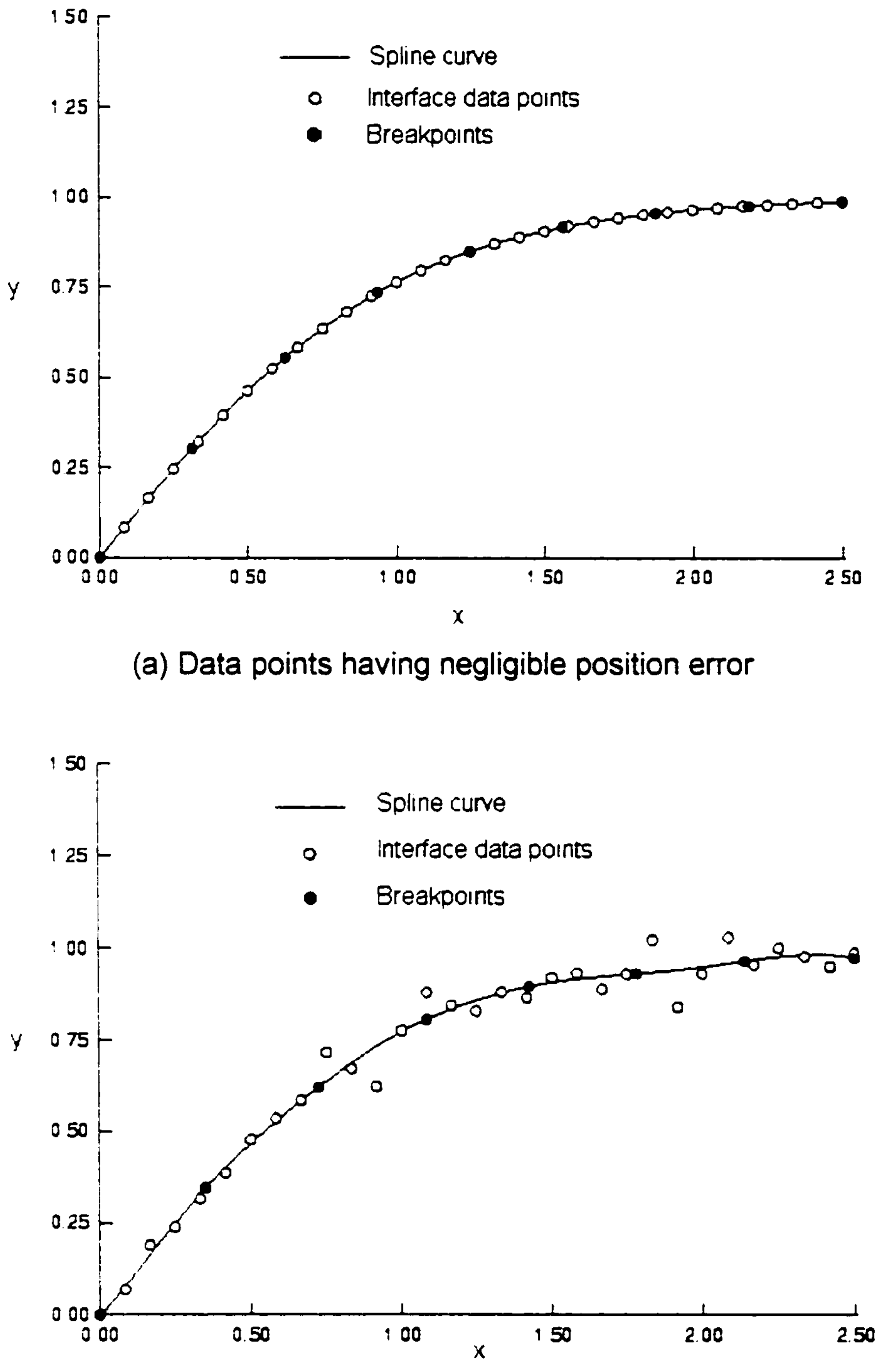
Figure 3.4 Example of parametric representation using normalized cumulative chord length.



(a) Data points having negligible position error

(b) Data points with random position error

Figure 3.5 Traditional cubic spline representation with interface data points taken as breakpoints (NB = NP).



(a) Data points having negligible position error

(b) Data points with random position error

Figure 3.6 Least-squares cubic spline representation with breakpoints distinct from interface data points (NB < NP).

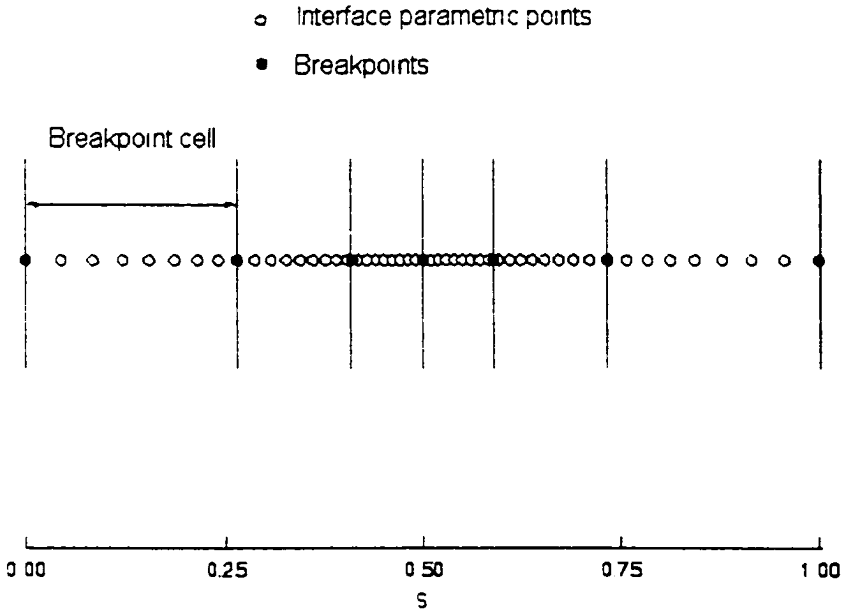


Figure 3.7 Breakpoint distribution using equally populated cells.

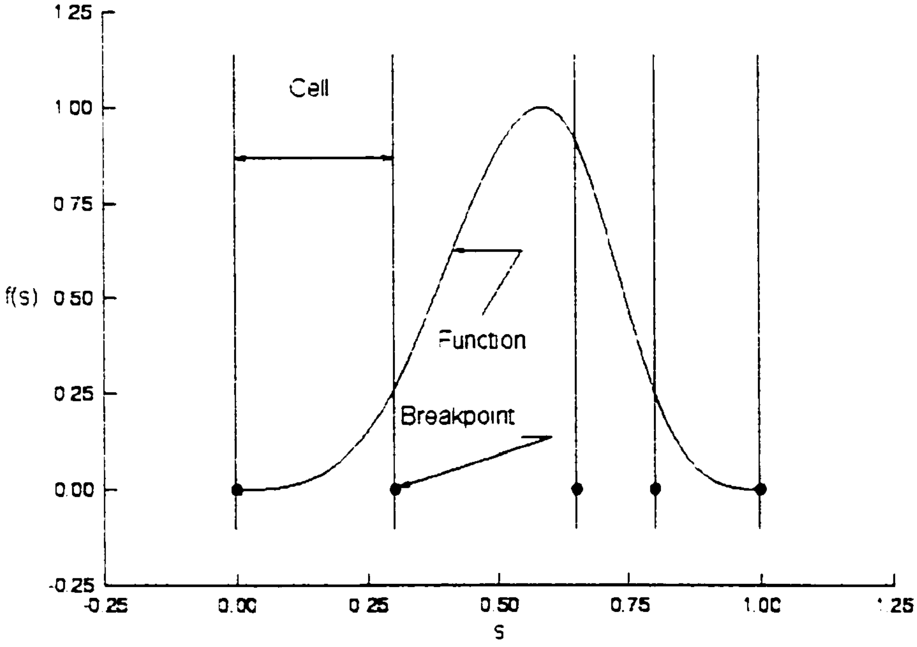


Figure 3.8 Typical basis function defined over five breakpoints.

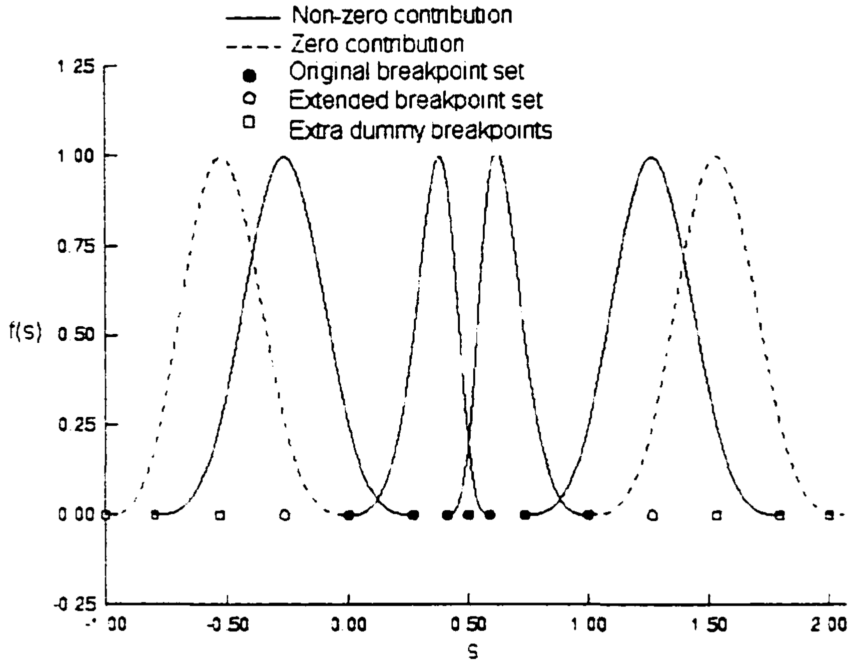


Figure 3.9 Extension of breakpoint set to agree with dimension of basis-function space.

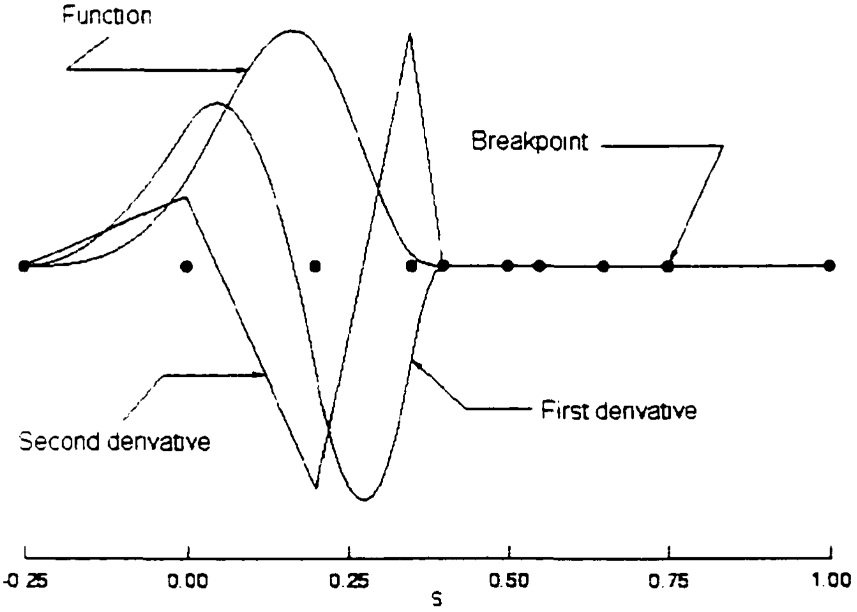


Figure 3.10 Typical basis function and first two derivatives.

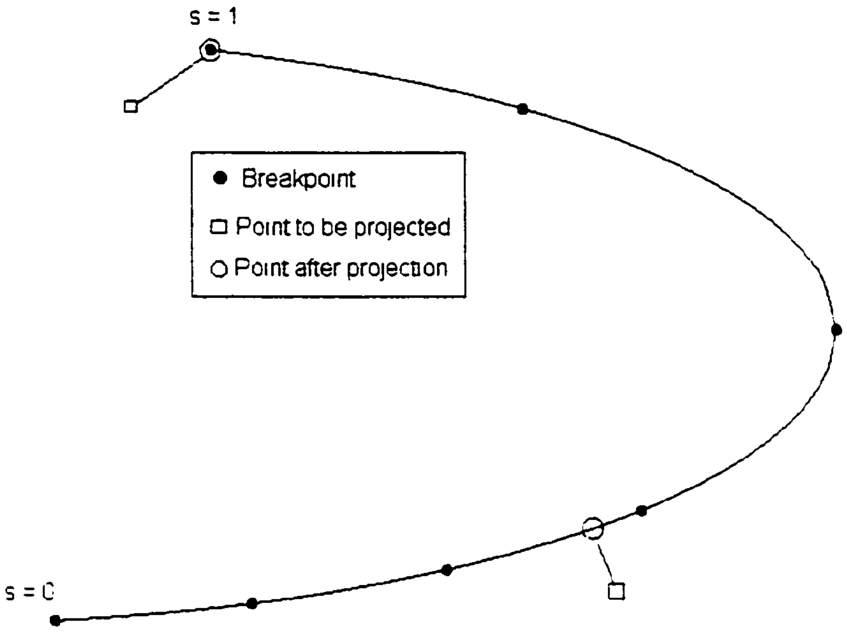


Figure 3.11 One-dimensional projection algorithm.

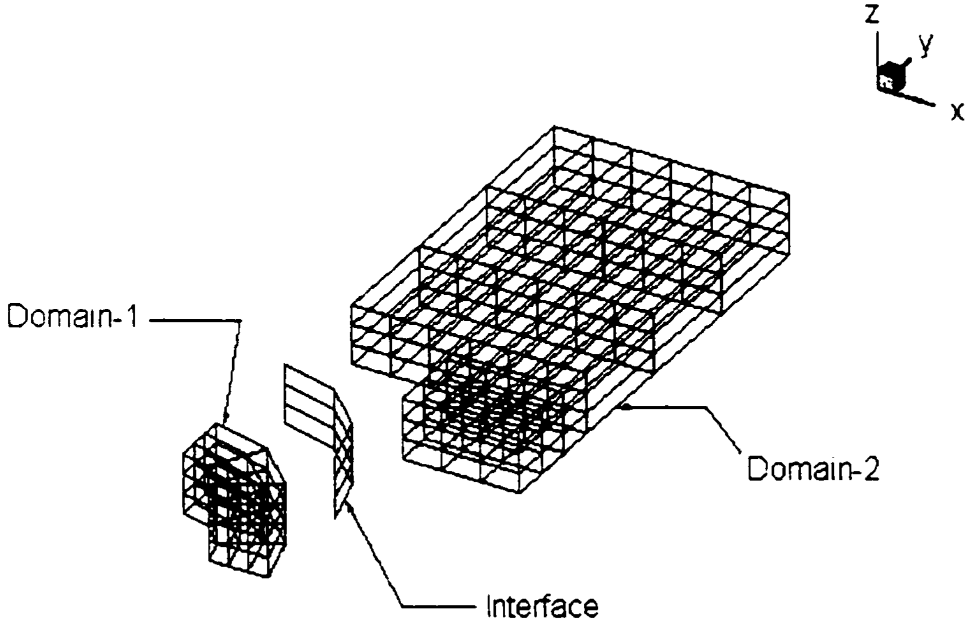
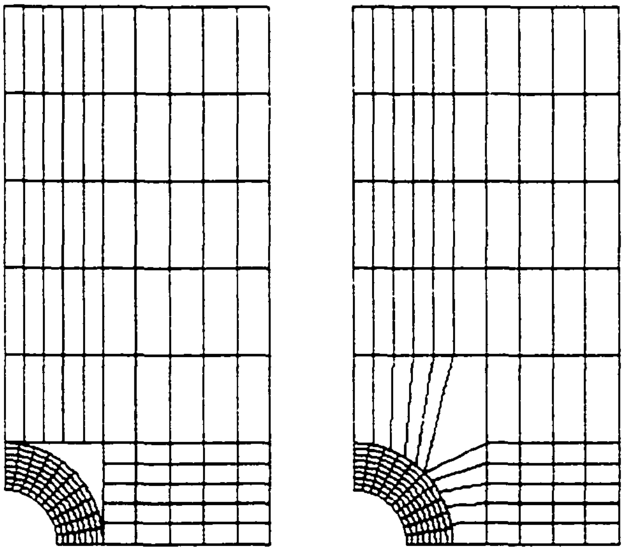


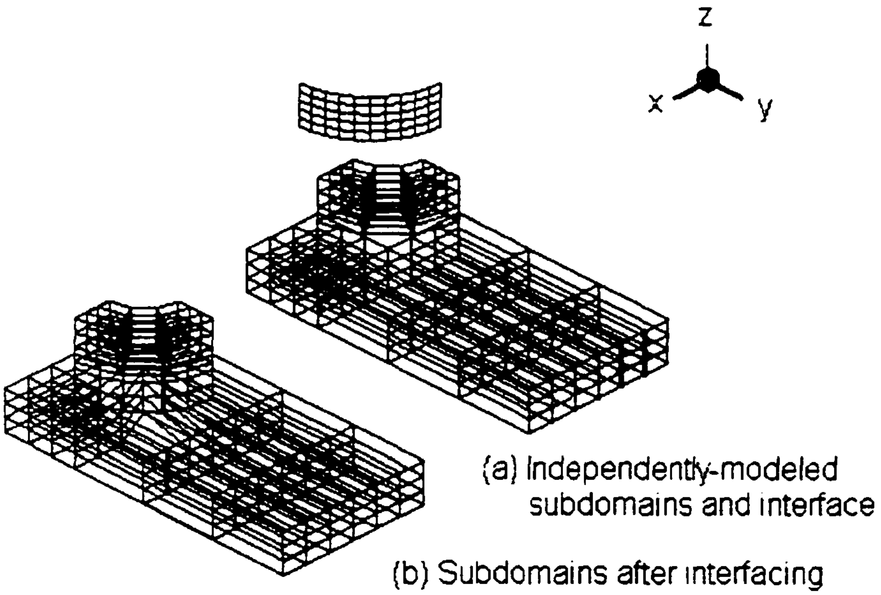
Figure 3.12 Independently-modeled subdomains and interface (exploded view for clarity).



(a) Independently modeled subdomains

(b) Interfaced subdomains

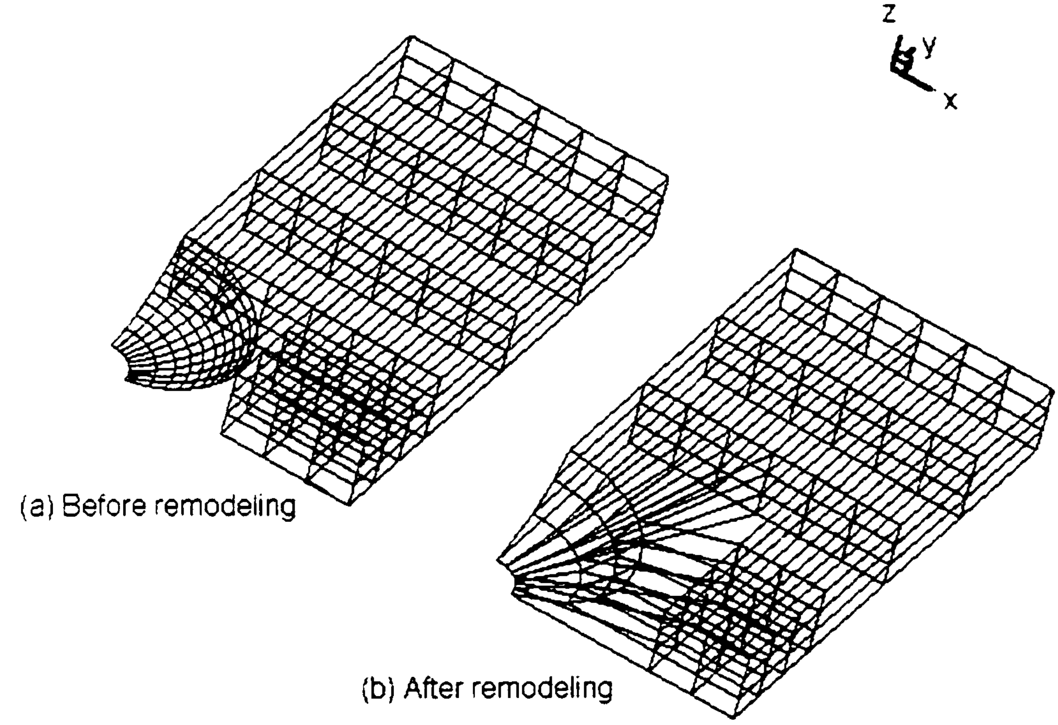
Figure 3.13 Automated interface modeling of two-dimensional flat plate with hole.



(a) Independently-modeled subdomains and interface

(b) Subdomains after interfacing

Figure 3.14 Automated interface modeling of solid plate with hole (exploded view for clarity).



(a) Before remodeling

(b) After remodeling

Figure 3.15 Example of automated remodeling for interface having multiple directions of curvature.

CHAPTER IV

GENERALIZED GLOBAL STIFFNESS MATRIX EVALUATION, ASSEMBLY AND SOLUTION

4.1 Introduction

A discussion of the approach and algorithms for creating the global generalized stiffness matrix, derived in Chapter II, is presented in this chapter. The global matrix and sub-matrices discussed herein are based on equation (2.20) of Chapter II for two interfaced sub-domains, which is reproduced here for convenient reference as equation (4.1):

$$\begin{bmatrix} [K_1] & [0] & [0] & [M_1] & [0] \\ [0] & [K_2] & [0] & [0] & [M_2] \\ [0] & [0] & [0] & [G_1] & [G_2] \\ [M_1]^T & [0] & [G_1]^T & [0] & [0] \\ [0] & [M_2]^T & [G_2]^T & [0] & [0] \end{bmatrix} \begin{Bmatrix} \{q_1\} \\ \{q_2\} \\ \{q_s\} \\ \{\alpha_1\} \\ \{\alpha_2\} \end{Bmatrix} = \begin{Bmatrix} \{P_1\} \\ \{P_2\} \\ \{0\} \\ \{0\} \\ \{0\} \end{Bmatrix}, \qquad (4.1)$$

where the $[K_i]$ arrays are the usual stiffness matrices for each sub-domain, the $\left[M_i\right]$ arrays are submatrices that couple the displacements and tractions, and the $[G_i]$ arrays are submatrices that couple tractions between subdomains as discussed in Chapter III. The $\{P_i\}$ arrays are sub-vectors representing loads or displacements imposed on the sub-domains. The $\{q_i\}$ are unknown degrees-of-freedom (dof) for each sub-domain and $\{q_s\}$ is a vector of unknown dof on the interface defined in terms of basis-function coefficients. The $\{\alpha_i\}$ vectors are compatibility dof corresponding to the displacement-coupling matrices, $[G_i]$.

The overall building process for equation (4.1) is to compute each submatrix of equation (4.1), assemble the sub-matrices into the global matrix, and finally compute and assemble the right-hand-side vector from imposed boundary conditions and constraints. A chart of the overall process for building and solving equation (4.1) showing six major steps is given in Figure 4.1.

The six steps of the process include: automated interface modeling; interface parametric representation; concurrent generation of $[\mathbf{K}_i]$, $[\mathbf{G}_i]$, and $[\mathbf{M}_i]$ sub-matrices for each domain (i=1, 2 in this example); global assembly of sub-matrices followed by preconditioning of the global matrix; solution of the generalized linear system for the displacement field; and finally recovery of the stress field from the displacements.

Since the automated interface modeling and parametric representation were discussed in Chapter III, the discussion in this chapter centers on evaluating the $[K_i]$, $[G_i]$, and $[\mathbf{M}_i]$ sub-matrices appearing in equation (4.1), assembly and pre-conditioning of the global generalized stiffness matrix, solution of the resulting linear algebraic system, and subsequent stress recovery techniques. The generation of the assembled linear stiffness sub-domain matrices (K-matrices) is accomplished using a previously existing finite-element code as a framework and will therefore be discussed somewhat briefly; the main focus of this chapter being on the remaining steps involved in evaluating and solving equation (4.1).

The manner of presentation for this chapter follows an organization similar to that of chapter III. Algorithms for two-dimensional configurations having one-dimensional interfaces are discussed first. This is followed by a discussion of algorithms for three-dimensional configurations having two-dimensional interfaces.

4.2 Evaluation of sub-matrices [K], [G], and [M]

4.2.1 Algorithms for two-dimensional domains

4.2.1.1 Standard linear stiffness matrices

The elements used for two-dimensional plane stress configurations in the present research were taken to be four-node plane quadrilaterals with bilinear, isoparametric shape functions as defined and discussed in many standard finite-element textbooks (e.g., Cook et al.,[25] Bathe,[26] and Krishnamoorthy[27]). The node arrangement for this element is shown in Figure 4.2, and the shape function definitions are given in equations (4.2) - (4.5):

$$N_1 = \tfrac{1}{4}(1-\xi)(1-\eta) \tag{4.2}$$

$$N_2 = \tfrac{1}{4}(1+\xi)(1-\eta) \tag{4.3}$$

$$N_3 = \tfrac{1}{4}(1+\xi)(1+\eta) \tag{4.4}$$

$$N_4 = \tfrac{1}{4}(1-\xi)(1+\eta). \tag{4.5}$$

The independent variables ξ and η used in the shape-function definitions are local isoparametric variables over the interval [-1,1], as shown in Figure 4.2.

Evaluation of the stiffness matrix, $[\mathbf{K}_i]$, for each sub-domain proceeds in a manner similar to that discussed by Cook et al.[25] and results from an assembly of element stiffness matrices which are evaluated according to the integral presented in equation (4.6):

$$[\mathbf{k}_e] = \int_{-1}^{+1}\int_{-1}^{+1} [\mathbf{B}]^T [\mathbf{E}][\mathbf{B}]\, t J \, d\zeta \, d\eta, \tag{4.6}$$

where [**B**] is written as a product of three subsidiary matrices and expresses the relation between strains and nodal displacements. The matrix [**E**] is the linear elastic constitutive material property matrix relating stress to strain through Hooke's Law. The factor, t, is the element thickness, and J is the determinate of the Jacobian of the transformation between global physical coordinates and local element isoparametric coordinates.

To evaluate the J-term of equation (4.6), the Jacobian matrix is conveniently written in the following form:

$$[J] = [D_N]\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix}, \tag{4.7}$$

where $[D_N]$ is an array of partial derivatives of the shape functions as defined in equation (4.8):

$$[D_N] = \begin{bmatrix} N_{1,\eta} & N_{2,\xi} & N_{3,\xi} & N_{4,\xi} \\ N_{1,\eta} & N_{2,\eta} & N_{3,\eta} & N_{4,\eta} \end{bmatrix}.$$

(4.8)

With $[J]$ thus defined, its determinate is easily evaluated as $J = J_{11}J_{22} - J_{21}J_{12}$. The inverse of $[J]$ is denoted by $[\Gamma] = [J]^{-1}$ and the elements of the inverse are denoted by $\Gamma_{11}, \Gamma_{12}$and so on.

The material property matrix, [**E**], of an isotropic, Hookean material in two dimensions, assuming a plane stress condition, is given by equation (4.9):

$$[E] = \left(\frac{E}{1-\nu^2}\right)\begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \dfrac{1-\nu}{2} \end{bmatrix}, \qquad (4.9)$$

where E is the elastic modulus and ν is Poisson's ratio for the material.

The B-matrix is conveniently evaluated as a product of three subsidiary matrices in the form, [**B**] = [**H**][**G**$_{\text{block}}$][**D**$_{\text{NE}}$]. These three matrices are defined respectively in equations (4.10), (4.11) and (4.12):

$$[H] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \qquad (4.10)$$

$$[G_{\text{block}}] = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & 0 & 0 \\ \Gamma_{21} & \Gamma_{22} & 0 & 0 \\ 0 & 0 & \Gamma_{11} & \Gamma_{12} \\ 0 & 0 & \Gamma_{21} & \Gamma_{22} \end{bmatrix} \qquad (4.11)$$

$$\left[D_{NE}\right]=\begin{bmatrix} N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} & 0 \\ N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} & 0 \\ 0 & N_{1,\xi} & 0 & N_{2,\xi} & 0 & N_{3,\xi} & 0 & N_{4,\xi} \\ 0 & N_{1,\eta} & 0 & N_{2,\eta} & 0 & N_{3,\eta} & 0 & N_{4,\eta} \end{bmatrix}. \qquad (4.12)$$

The H-matrix represents the relation between strain and displacement derivatives with respect to physical coordinates. The G_{block}-matrix couples displacement derivatives with respect to isoparametric coordinates and physical coordinates and the D_{NE}-matrix expresses displacement derivatives in terms of nodal displacements. Finally, the B-matrix expresses the relationship between strains and nodal displacements.

4.2.1.2 Traction continuity matrices

Evaluation of the M-matrices reduces immediately to evaluation of the defining integral given in equation (2.18) of Chapter II. For convenient reference, the integral is re-stated in equation (4.13):

$$\left[M_j\right]=-\int_{S_j}\left[N_j\right]^T\left[R_j\right]ds,$$

(4.13)

where $[N_j]$ represents the matrix of shape functions for the element type used in domain j and $[R_j]$ represents the corresponding interpolation function for traction. Since traction is related to displacement through derivatives, e.g. equations (2.4) -(2.14) of Chapter II, the R-functions are generally taken to be polynomials of one degree less than the degree of the polynomials defining the N-functions. Furthermore, since the N-functions in the present case are bilinear (e.g. equations 4.2 - 4.5), the values in the R-arrays will be constant on a given element and, in general, discontinuous between elements.

The integral defined by equation (4.13) is evaluated element-by-element for those elements on the interface. Thus, equation (4.13) is written as

$$\left[M_j\right]=-\sum_{i=1}^{NEL}\int_{-1}^{+1}\left[N_j\right]_i^T\left[R_j\right]_i\left[B_{NZ}\right]_{ij}\,tJ_i\,d\zeta. \qquad (4.14)$$

The sum in equation (4.14) proceeds over the elements for each domain j, and the N- and R-arrays are defined in the element sense. Each [$\mathbf{B}_{NZ}$] array is a Boolean array to place the non-zero contributions of elements and sides of elements on the interface. Each [$\mathbf{B}_{NZ}$] array is of dimension 8 by ($2n_j$), where n_j is the number of nodes in sub-domain j. Array [$\mathbf{B}_{NZ}$] is formed as the product of two subsidiary arrays, [$\mathbf{B}_E$] and [$\mathbf{B}_G$], i.e., [$\mathbf{B}_{NZ}$] = [$\mathbf{B}_E$] [$\mathbf{B}_G$]. Array [$\mathbf{B}_E$], of dimension 8 by 8, is an element Boolean array that places the integration on the side of the element that is on the interface. Array [$\mathbf{B}_G$], of dimension 8 by ($2n_j$), is a sub-domain global Boolean array that extracts any element that is on the interface. The factor, J is the Jacobian determinate relating the local isoparametric coordinate of the side along the interface to the global physical coordinates, and t is the local element thickness. The integration variable, $d\zeta$, is a generic variable representing the local isoparametric coordinate of the element side that is on the interface.

The expression for the Jacobian determinate is derived by reasoning as follows. First, for an element side on the interface, the physical coordinates of the nodes defining the side are denoted by: (x_1,y_1) and (x_2,y_2). Second, the arc length differential, in two dimensions, is given by: $ds=\sqrt{dx^2+dy^2}$. Third, the relation between the global variables and the isoparametric variable is assumed locally linear, so the relationship between ds and $d\zeta$ is: $ds = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}\, d\zeta$. This relation between ds and $d\zeta$ is illustrated in Figure 4.3 for a representative example.

From an inspection of Figure 4.3, and by reference to the equation relating ds and $d\zeta$, it evident that the local Jacobian determinate is given by: $J_i = \frac{1}{2}\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$. Each [$\mathbf{N}_j$] array occurring in the integrand of equation (4.14) is a 2 by 8 array of shape functions arranged in the form given by equation (4.15):

$$\left[\mathbf{N}_j\right]=\begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}. \tag{4.15}$$

In equation (4.15), the shape functions N_1, N_2, and so on are bilinear functions of the isoparametric variables, ξ and η, as defined in equations (4.2) - (4.5). The traction, being derived from derivatives of bilinear shape functions, is constant over a given element. Thus, each [$\mathbf{R}_j$] array in the integrand of Equation (4.14) is a 2 by 8 array of constants, the values of which are chosen to give suitable tractions interpolated from the nodal values on the interface. The evaluation of the integrals in equation (4.14) was accomplished using two-point (second-order) Gauss quadrature.

4.2.1.3 Displacement-coupling matrices

Evaluation of the G-matrices was based on equation (2.19) of Chapter II, and that equation is reproduced here for reference as equation (4.16):

$$\left[\mathbf{G}_j\right]=\int\left[\mathbf{T}\right]^T\left[\mathbf{R}_j\right]ds. \tag{4.16}$$

In equation (4.16), [T] denotes the array of functions used to represent the geometrical definition of the interface between sub-domains 1 and 2. The [$\mathbf{R}_j$] array in equation (4.16) is the same array as is defined in equation (4.14). As was the case for the M-arrays, the G-arrays were also evaluated element-by-element with the integration expressed as in equation (4.17):

$$\left[\mathbf{G}_j\right]=\sum_{i=1}^{NEL}\int_{-1}^{+1}\left[\mathbf{T}\right]^T\left[\mathbf{R}_j\right]\left[\mathbf{B}_{NZ}\right]_{ij}\, tJ_i\, d\zeta. \tag{4.17}$$

The array, [T], of dimension 2 by ($2n_{bf}$), (where n_{bf} is the number of basis functions used in defining the geometrical interface) is arranged in the form given by equation (4.18):

$$\left[\mathbf{T}\right]=\begin{bmatrix} b_1 & 0 & b_2 & 0 & \cdots & b_{n_{bf}} & 0 \\ 0 & b_1 & 0 & b_2 & \cdots & 0 & b_{n_{bf}} \end{bmatrix}. \tag{4.18}$$

The functions $b_k(s(\zeta))$ in the T-array are the basis functions defined by equation (3.5) of Chapter III, and the other terms in the integrand of equation (4.17) are defined the same as in equation (4.14).

4.2.2 Algorithms for three-dimensional domains

4.2.2.1 Standard linear stiffness matrices

The elements for three-dimensional configurations in the present research are taken to be eight-node hexahedrons ('bricks') with trilinear, isoparametric shape functions as defined and discussed in References 29-32 and 49, for example. The node arrangement for this element is shown in Figure 4.4 and the shape function definitions corresponding to the arrangement of Figure 4.4 are given in equation (4.19) which are similar to those given by Cook et al.,[25]

$$N_j = \tfrac{1}{8}(1+\xi_j\xi)(1+\eta_j\eta)(1+\zeta_j\zeta), \quad (j=1,2,3,\ldots,8). \tag{4.19}$$

The constants, ξ_j, η_j, and ζ_j in equation (4.19) depend on the node index, j, and are defined by equations (4.20), (4.21), and (4.21):

$$\{\xi_j\}_{(j=1,2,3,\ \ .8)} = \{-1,-1,-1,-1,1,1,1,1\}^T \tag{4.20}$$

$$\{\eta_j\}_{(j=1,2,3,\ \ .8)} = \{-1,-1,1,1,-1,-1,1,1\}^T \tag{4.21}$$

$$\{\zeta_j\}_{(j=1,2,3,\ \ .8)} = \{1,-1,-1,1,1,-1,-1,1\}^T. \tag{4.22}$$

Evaluation of the stiffness matrix, **[K]**, for each domain proceeds by assembling results from element stiffness matrices similar to the two-dimensional case discussed in section 4.2.1.1. The element stiffness matrix, **[k$_e$]**, for this case is given by equation (4.23):

$$[\mathbf{k_e}]_i = \int_{-1}^{+1}\int_{-1}^{+1}\int_{-1}^{+1}[\mathbf{B}]_i^T[\mathbf{E}][\mathbf{B}]_i\, J d\xi d\eta d\zeta. \tag{4.23}$$

As in the two-dimensional case, the array **[B]** is written as a product of three subsidiary matrices and expresses the relation between strains and nodal displacements.

The array [E] is the material property matrix relating stress to strain, and J is the determinate of the Jacobian of the transformation between global physical coordinates and local element isoparametric coordinates.

The Jacobian matrix, [J], is defined similarly to equation (4.7) of the plane-stress case except now [J] is of dimension 3 by 3 and is given by equation (4.24):

$$[J]=[D_N]\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \\ x_4 & y_4 & z_4 \\ x_5 & y_5 & z_5 \\ x_6 & y_6 & z_6 \\ x_7 & y_7 & z_7 \\ x_8 & y_8 & z_8 \end{bmatrix}. \qquad (4.24)$$

The factor, $[D_N]$, is a 3 by 8 array of partial derivatives of the shape functions and is given by equation (4.25), wherein subscripts with a comma denote partial derivatives:

$$[D_N]=\begin{bmatrix} N_{1,\xi} & N_{2,\xi} & N_{3,\xi} & N_{4,\xi} & N_{5,\xi} & N_{6,\xi} & N_{7,\xi} & N_{8,\xi} \\ N_{1,\eta} & N_{2,\eta} & N_{3,\eta} & N_{4,\eta} & N_{5,\eta} & N_{6,\eta} & N_{7,\eta} & N_{8,\eta} \\ N_{1,\zeta} & N_{2,\zeta} & N_{3,\zeta} & N_{4,\zeta} & N_{5,\zeta} & N_{6,\zeta} & N_{7,\zeta} & N_{8,\zeta} \end{bmatrix}. \qquad (4.25)$$

The material property matrix, [E], is that of an isotropic, Hookean material in three dimensions and is given in equation (4.26):

$$[E]=\frac{E}{(1+\nu)(1-2\nu)}\begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}. \qquad (4.26)$$

As usual, E is the elastic modulus and ν is Poisson's ratio for the material.

The B-matrix is conveniently expressed as a product of three subsidiary matrices in the form $[B]=[H][G_{block}][D_{NE}]$. The H-matrix arises by expressing the strain vector in terms of the vector of linear displacement derivatives. The G_{block}-matrix expresses the relation between derivatives of displacement in global variables to derivatives of displacement in local isoparametric variables. The D_{NE}-matrix is an array of partial derivatives of shape functions. Thus, in matrix form, the strains are expressed in terms of the displacement-derivatives as equation (4.27):

$$\begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \varepsilon_z \\ \gamma_{xy} \\ \gamma_{yz} \\ \gamma_{zx} \end{Bmatrix}_{6\times1} = [H]_{6\times9} \begin{Bmatrix} u_{,x} \\ u_{,y} \\ u_{,z} \\ v_{,x} \\ v_{,y} \\ v_{,z} \\ w_{,x} \\ w_{,y} \\ w_{,z} \end{Bmatrix}_{9\times1} .$$

(4.27)

The H-array of equation (4.27) is written as:

$$[H]=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \qquad (4.28)$$

and represents the relation between strain and displacement derivatives with respect to physical coordinates. Next, the relationship between derivatives of displacement in global physical variables and derivatives in local isoparametric variables is expressed in equation (4.29):

$$\begin{Bmatrix} u_{,x} \\ u_{,y} \\ u_{,z} \\ v_{,x} \\ v_{,y} \\ v_{,z} \\ w_{,x} \\ w_{,y} \\ w_{,z} \end{Bmatrix}_{9\times 1} = \left[G_{block}\right]_{9\times 9} \begin{Bmatrix} u_{,\xi} \\ u_{,\eta} \\ u_{,\zeta} \\ v_{,\xi} \\ v_{,\eta} \\ v_{,\zeta} \\ w_{,\xi} \\ w_{,\eta} \\ w_{,\zeta} \end{Bmatrix}_{9\times 1}, \qquad (4.29)$$

where the G_{block}-array is a 9 by 9 matrix defined by

$$[G]_{block} = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma_{11} & \Gamma_{12} & \Gamma_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma_{21} & \Gamma_{22} & \Gamma_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & \Gamma_{31} & \Gamma_{32} & \Gamma_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ 0 & 0 & 0 & 0 & 0 & 0 & \Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\ 0 & 0 & 0 & 0 & 0 & 0 & \Gamma_{31} & \Gamma_{32} & \Gamma_{33} \end{bmatrix}. \qquad (4.30)$$

In equation (4.30), each 3 by 3 sub-matrix, $[\Gamma]$, is the matrix inverse of the Jacobian which was previously defined in equation (4.24); thus $[\Gamma]$is written as

$$[\Gamma] = \begin{bmatrix} \Gamma_{11} & \Gamma_{12} & \Gamma_{13} \\ \Gamma_{21} & \Gamma_{22} & \Gamma_{23} \\ \Gamma_{31} & \Gamma_{32} & \Gamma_{33} \end{bmatrix} = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}^{-1}. \qquad (4.31)$$

Lastly, matrix $[D_{NE}]$ is given by equation (4.32), a three-dimensional extension of the two-dimensional definition in equation (4.12):

$$[D_{NE}] = \begin{bmatrix} N_{1,\xi} & 0 & 0 & N_{2,\xi} & 0 & 0 & \cdots & 0 & 0 & N_{8,\xi} & 0 & 0 \\ N_{1,\eta} & 0 & 0 & N_{2,\eta} & 0 & 0 & \cdots & 0 & 0 & N_{8,\eta} & 0 & 0 \\ N_{1,\zeta} & 0 & 0 & N_{2,\zeta} & 0 & 0 & \cdots & 0 & 0 & N_{8,\zeta} & 0 & 0 \\ 0 & N_{1,\xi} & 0 & 0 & N_{2,\xi} & 0 & \cdots & N_{7,\xi} & 0 & 0 & N_{8,\xi} & 0 \\ 0 & N_{1,\eta} & 0 & 0 & N_{2,\eta} & 0 & \cdots & N_{7,\eta} & 0 & 0 & N_{8,\eta} & 0 \\ 0 & N_{1,\zeta} & 0 & 0 & N_{2,\zeta} & 0 & \cdots & N_{7,\zeta} & 0 & 0 & N_{8,\zeta} & 0 \\ 0 & 0 & N_{1,\xi} & 0 & 0 & N_{2,\xi} & \cdots & 0 & N_{7,\xi} & 0 & 0 & N_{8,\xi} \\ 0 & 0 & N_{1,\eta} & 0 & 0 & N_{2,\eta} & \cdots & 0 & N_{7,\eta} & 0 & 0 & N_{8,\eta} \\ 0 & 0 & N_{1,\zeta} & 0 & 0 & N_{2,\zeta} & \cdots & 0 & N_{7,\zeta} & 0 & 0 & N_{8,\zeta} \end{bmatrix}_{24\cdot 9} . \qquad (4.32)$$

4.2.2.2 Traction continuity matrix for three-dimensional elements

The general expression for the traction-continuity matrix for each sub-domain is given in equation (4.11), which is evaluated element-by-element with only those elements having a face on the interface producing non-zero contributions. Equation (4.11) is thus written for three-dimensional elements as:

$$\left[\mathbf{M}_j\right] = -\sum_{i=1}^{NEL} \int_{-1}^{+1} \int_{-1}^{+1} \left[\mathbf{N}_i\right]^T \left[\mathbf{R}_j\right]_i \left[\mathbf{B}_{NZ}\right]_i J_i \, d\alpha \, d\beta. \qquad (4.33)$$

The integration variables, α and β, are generic variables that represent local isoparametric variables for any element face on the interface. The sum goes over all elements of a fixed domain j, and each [**N**] and [**R**] array is defined in an element sense as for the two-dimensional case. Each [$\mathbf{B}_{NZ}$] array of dimension 24 by ($3n_j$) (where n_j is the number of nodes in sub-domain j) is a Boolean array that extracts the non-zero contributions of the elements and element faces that are on the interface.

The [$\mathbf{B}_{NZ}$] arrays are formed as the product of two subsidiary arrays, [$\mathbf{B}_E$] and [$\mathbf{B}_G$]. The [$\mathbf{B}_G$] arrays, each of dimension 24 by ($3n_j$), are sub-domain global Boolean arrays that extract the elements of the sub-domain that are on the interface. The [$\mathbf{B}_E$] arrays, each of dimension 24 by 24, are element-level Boolean arrays that extract the face of the element that is on the interface. The factors, J_i, are the Jacobian determinates relating the local isoparametric variables of the element-face on the interface to the global coordinate variables. The evaluation of the integrals in equation (4.33) was accomplished by second-order Gauss quadrature (two points per integration variable) as in the two-dimensional case previously discussed.

4.2.2.3 Displacement coupling matrices for three-dimensional elements

Evaluation of the G-matrices for three-dimensional configurations is based on equation (4.16) written for element-by-element evaluation as:

$$\left[\mathbf{G}_j\right] = \sum_{i=1}^{NEL} \int_{-1}^{+1} \int_{-1}^{+1} [\mathbf{T}]^T \left[\mathbf{R}_j\right]_i \left[\mathbf{B}_{NZ}\right]_i J_i \, d\alpha \, d\beta, \tag{4.34}$$

where α and β are generic variables of integration representing local iso-parametric variables for the element-face on the interface, and J_i is the determinate of the Jacobian matrix relating the isoparametric variables on the face of element 'i' to the global variables. The relationship between local integration variables, $d\alpha$ and $d\beta$, and interface parametric variables, ds and dt, is shown in Figure 4.5. The array, [T], is an array of basis functions used in defining the geometrical interface between the finite-element models of the three dimensional structure. This array, of dimension 3 by ($3n_{bf}$), where nbf is the number of basis functions used to define the interface, is given in equation (4.35):

$$[\mathrm{T}] = \begin{bmatrix} btp_1 & 0 & 0 & btp_2 & 0 & 0 & \cdots & btp_{n_{bf}} & 0 & 0 \\ 0 & btp_1 & 0 & 0 & btp_2 & 0 & \cdots & 0 & btp_{n_{bf}} & 0 \\ 0 & 0 & btp_1 & 0 & 0 & btp_2 & \cdots & 0 & 0 & btp_{n_{bf}} \end{bmatrix}_{3\times 3n_{bf}} . \tag{4.35}$$

The functions, $btp_k(s(\alpha,\beta),t(\alpha,\beta))$, in the T-array are the tensor-product basis functions defined by equation (3.16) of Chapter III, and the other terms in the integral of equation (4.34) are the same as defined in equation (4.33). Incidentally, the notation "btp" was chosen to indicate tensor-product basis functions.

4.3 Global assembly and solution

4.3.1 Global assembly

Numerical implementation of the procedures described in Sections 4.1 and 4.2 result in computer files containing the sub-matrices: [$\mathbf{K}_1$], [$\mathbf{K}_2$], [$\mathbf{M}_1$], [$\mathbf{M}_2$], [$\mathbf{G}_1$], and [$\mathbf{G}_2$]. These six files are assembled with the load and constraint vectors into a modified global linear system of algebraic equations as expressed in equation (4.1). The procedures used to create the sub-matrices, as implemented, result in a sparse global array that contains a significant number of zero rows and columns. Thus, as initially assembled, the generalized global stiffness matrix is singular and quite sparse.

In order to obtain a non-singular linear system, which could be solved by standard methods, a deflation and scaling procedure was implemented to modify the initial system. This process not only removes singular rows and columns but also additionally detects rows that are nearly redundant in a generalized vector sense. Such rows and corresponding columns are also deflated out of the system, and the resulting deflated system is scaled to have row-norms of similar magnitude. The procedure, as implemented, keeps track of indices that correspond to node locations in the finite-element model where displacements are to be computed. The method uses a pointer indexing system so that after the solution is obtained all results are relocated in their respective correct locations.

A master file, which is produced by the assembly pre-conditioning module, contains the deflated linear system coefficients, the corresponding modified right-hand-side vector, and the pointer vector containing integer index-recovery information. This master file is the file received by the solution module where the linear system is then solved for the unknown displacements.

4.3.2 Solution of the linear algebraic system

The generalized linear stiffness matrix, as received from the assembly module, is non-singular but it may not be positive definite. Such a linear algebraic system can be efficiently solved by the method of Gauss elimination and full pivoting. Although this method has been implemented in several standard packages over the years, the implementation described by Dongarra et al.[98] for the LINPACK system is robust, well tested, and widely available. Routines from the LINPACK system were therefore included as part of the solution module for the present research. For additional robustness and versatility, a user option was included to also allow solution by a standard singular value decomposition (SVD) method. In practice, if the deflated system is near singular, as evidenced by its condition number, then Gauss elimination with full pivoting may not give satisfactory results. In this case, the SVD approach may be used although it requires considerably more computation time than the Gauss elimination method.

After the solution to the modified system is obtained, by either the Gauss elimination method or the SVD method, the displacement values at the correct node locations are obtained by using the indexing information carried by the pointer vector. The resulting array of displacement values at the node locations is then used to obtain the stress field.

4.4 Stress recovery

4.4.1 Introduction

The displacement degrees-of-freedom obtained from a finite-element analysis are normally available at the node locations. Moreover, since many graphics packages (e.g., TECPLOT[99]) are based on having information at the node locations, and it is also useful to have stress values at the nodes. Unfortunately, from an accuracy standpoint, direct computation of stress at the node locations appears to be among the poorest of choices (e.g., Hinton and Campbell[100]). A method of obtaining accurate stress values at the node locations was therefore of primary importance.

4.4.2 Stress Recovery Procedure

The stress recovery methodology used in the present research is a generalized form of an approach developed for two-dimensional finite-element models discussed by Krishnamoorthy.[27] As implemented, the present method entailed three basic steps. First, the stresses were computed from constitutive equations and displacement derivatives at optimum sampling points for the element in use. Second, the resulting stress values were extrapolated to the node locations by means of isoparametric shape functions similar to those used for the original finite-element model. Third, multiple stress values, which arise at nodes having two or more elements in common, were averaged to obtain unique values at each node location.

4.4.2.1 Optimum stress sampling locations

The location of optimal stress sampling points for a given element type has been investigated by previous researchers. Representative examples of these investigations are discussions by Strang and Fix,[101] Zienkiewicz and Taylor,[90] and Barlow.[102] The best choice of location for the stress sampling points is related to the order of the Gauss quadrature rule (i.e., the number of Gauss points) used in computing the integrals for evaluating the entries in the generalized stiffness matrix. The second-order rule (two Gauss points per dimension) was used for the present research as recommended in the following quotation from Cook et al.:[25]

> For an isoparametric element based on an assumed displacement field, the best quadrature rule is usually the lowest-order rule that computes the volume correctly and does not produce numerical instability. ... For bilinear and eight-node plane elements, and for the eight-node linear solid element, an order 2 Gauss rule is favored (four and eight points for plane and solid elements, respectively).

The above recommendation by Cook is based on the idea of choosing the order of quadrature to have a correct assessment of the strain energy in the structure. The logic goes as follows. First, the element must satisfy the patch test (represent a constant stress state) and the elements considered herein satisfy this requirement (this is shown in detail in Chapter V). Second, the element must exhibit zero strain for rigid-body motion. Third, the elements must be compatible and exhibit invariance with respect to coordinate rotations. These requirements are inherently satisfied by isoparametric elements, which are the elements used herein. Fourth, the quadrature rule should integrate exactly the element volume for arbitrary element shape. The quadrature rule stated in the above quote meets this requirement, and is the reason for Cook's recommendation.

The choice of related stress-sampling points used in the present research was based on the locations recommended by Hinton and Campbell,[100] who found that "In analysis involving numerically integrated elements such as isoparametric elements, experience has shown that the integration points are the best stress sampling points." On the other hand, for four-node quadrilateral and eight-node brick elements, Cook et al.[25] states that the most accurate stress sampling points are the element centroids. The drawback is that centroids used as sampling points makes extrapolation to the node points crude because it is impossible to capture a within-element variation with only one point.

For the present research, second-order Gauss quadrature (two points per dimension) was used and the corresponding locations in the element were used as stress sampling points for extrapolation to the nodes to get compatibility with contour plotting software. The stress was also computed at the centroids and saved to a file as a check on accuracy using of the stresses sampled at the Gauss points. It is evident from the literature that the last word on stress recovery algorithms has not been written and there is further room for research in this topic.

4.4.2.2 Extrapolation to the node locations

For second-order (two point) Gauss quadrature and four-node quadrilateral or eight-node brick elements, as was used in this research, the total number of Gauss points and stress sampling points in an element was the same as the number of nodes, i.e. four or eight depending on the element type. In each case, the coordinates of the Gauss points in terms of isoparametric variables were $\pm\frac{1}{\sqrt{3}}$.

The spatial relationship between the element node-locations and the Gauss point locations for a four-node quadrilateral element is shown in Figure

4.6. By extension, a similar spatial relation holds for three-dimensional eight-node brick elements. The number of stress sampling points was the same as the number of nodes for the elements used in the present research, so the same functional form was used for extrapolation from the sampling points as was used for displacement interpolation within the element. The distinction being that different isoparametric variables were used. For the case of quadrilateral elements, the isoparametric variables based on node locations were denoted by (ξ, η) and the isoparametric variables based on Gauss points were denoted by $(\bar{\xi}, \bar{\eta})$. Now, since the Gauss points were located at $\pm\frac{1}{\sqrt{3}}$ in the variables (ξ, η) and at ±1 in the variables $(\bar{\xi}, \bar{\eta})$, the two sets of variables were related as in equations (4.36) and (4.37):

$$\bar{\xi} = \sqrt{3}\xi$$

(4.36)

and

$$\bar{\eta} = \sqrt{3}\eta.$$

(4.37)

Evaluation of stress within the element was accomplished by using a representation in terms of the values at the Gauss points and the element shape functions with $(\bar{\xi}, \bar{\eta})$ as arguments for the functions. In particular, stresses at the node locations were obtained by evaluating the element bilinear shape functions, $\{N_1, N_2, N_3, N_4\}$ at $\pm\sqrt{3}$. The values of the normal stress in the x-direction at the nodes are expressed in terms of the corresponding values at the sampling points by equation (4.38):

$$\begin{Bmatrix} \bar{\sigma}_x^{(1)} \\ \bar{\sigma}_x^{(2)} \\ \bar{\sigma}_x^{(3)} \\ \bar{\sigma}_x^{(4)} \end{Bmatrix} = \begin{bmatrix} N_1(-\sqrt{3},-\sqrt{3}) & N_2(-\sqrt{3},-\sqrt{3}) & N_3(-\sqrt{3},-\sqrt{3}) & N_4(-\sqrt{3},-\sqrt{3}) \\ N_1(\sqrt{3},-\sqrt{3}) & N_2(\sqrt{3},-\sqrt{3}) & N_3(\sqrt{3},-\sqrt{3}) & N_4(\sqrt{3},-\sqrt{3}) \\ N_1(\sqrt{3},\sqrt{3}) & N_2(\sqrt{3},\sqrt{3}) & N_3(\sqrt{3},\sqrt{3}) & N_4(\sqrt{3},\sqrt{3}) \\ N_1(-\sqrt{3},\sqrt{3}) & N_2(-\sqrt{3},\sqrt{3}) & N_3(-\sqrt{3},\sqrt{3}) & N_4(-\sqrt{3},\sqrt{3}) \end{bmatrix} \begin{Bmatrix} \sigma_x^{(1)} \\ \sigma_x^{(2)} \\ \sigma_x^{(3)} \\ \sigma_x^{(4)} \end{Bmatrix}, \quad (4.39)$$

where the values at the sampling points are $\{\sigma_x^{(1)}, \sigma_x^{(2)}, \sigma_x^{(3)}, \sigma_x^{(4)}\}^T$ and the corresponding values at the nodes are $\{\bar{\sigma}_x^{(1)}, \bar{\sigma}_x^{(2)}, \bar{\sigma}_x^{(3)}, \bar{\sigma}_x^{(4)}\}^T$.

Similar equations for σ_y and τ_{xy} were obtained based on the same approach, and completely analogous relationships were used for the eight-node, three-dimensional element. In the present work, the 4 by 4 matrix in equation (4.38) is called a bilinear extrapolation matrix, and the analogous 8 by 8 matrix for the brick element is called a trilinear extrapolation matrix.

The extrapolation procedure begins by first computing the appropriate bilinear or trilinear extrapolation array. Then, by means of equations similar to equation (4.38), stress values at the nodes were computed from corresponding values at the sampling points for each element in the model.

4.4.2.3 Averaging nodal extrapolated stress values

During the course of extrapolating stress values at the Gauss points to values at the element node locations, it was possible that more than one stress value might be obtained at a single node when two or more elements had a node in common. This possibility is illustrated in Figure 4.7 for both quadrilateral and brick elements. Examination of the figure shows that there are three cases for the nodes; shared, not shared, and nodes on the subdomain boundary. In the event of shared nodes, the multiple nodal stress values from the common elements are averaged to obtain a single stress value for each node location in the finite-element model. A straightforward arithmetic average of the extrapolated values, as was done in the present research, implicitly assumes equal contribution from each of the adjacent elements. A more refined and sophisticated averaging method would use some type of weighting procedure based upon a predefined criterion such as element area or volume.

Values of the stresses are not treated differently from interior nodes in the extrapolation process, i.e., no distinction is made between boundary and interior nodes in the extrapolation process. Since nodes on a boundary will, in general, be shared by fewer elements than are shared by interior nodes, the nodes on the boundary have stress values that are obtained with less information than is used for a typical interior node. This may sometimes lead to more inaccurate values at the boundary nodes, particularly if the element density is low in the extrapolation direction.

Values of stress at the interface nodes are treated in the same way as boundary nodes, i.e., the interface node values are obtained with information only from the associated subdomain. To do it this way, and thereby forego 'smoothing' between the subdomains, was a deliberate decision for the present research in order to be able to identify effects due to the interface. In practice, one would likely use information from all subdomains connected to the interface and subsequently apply a smoothing or averaging process.

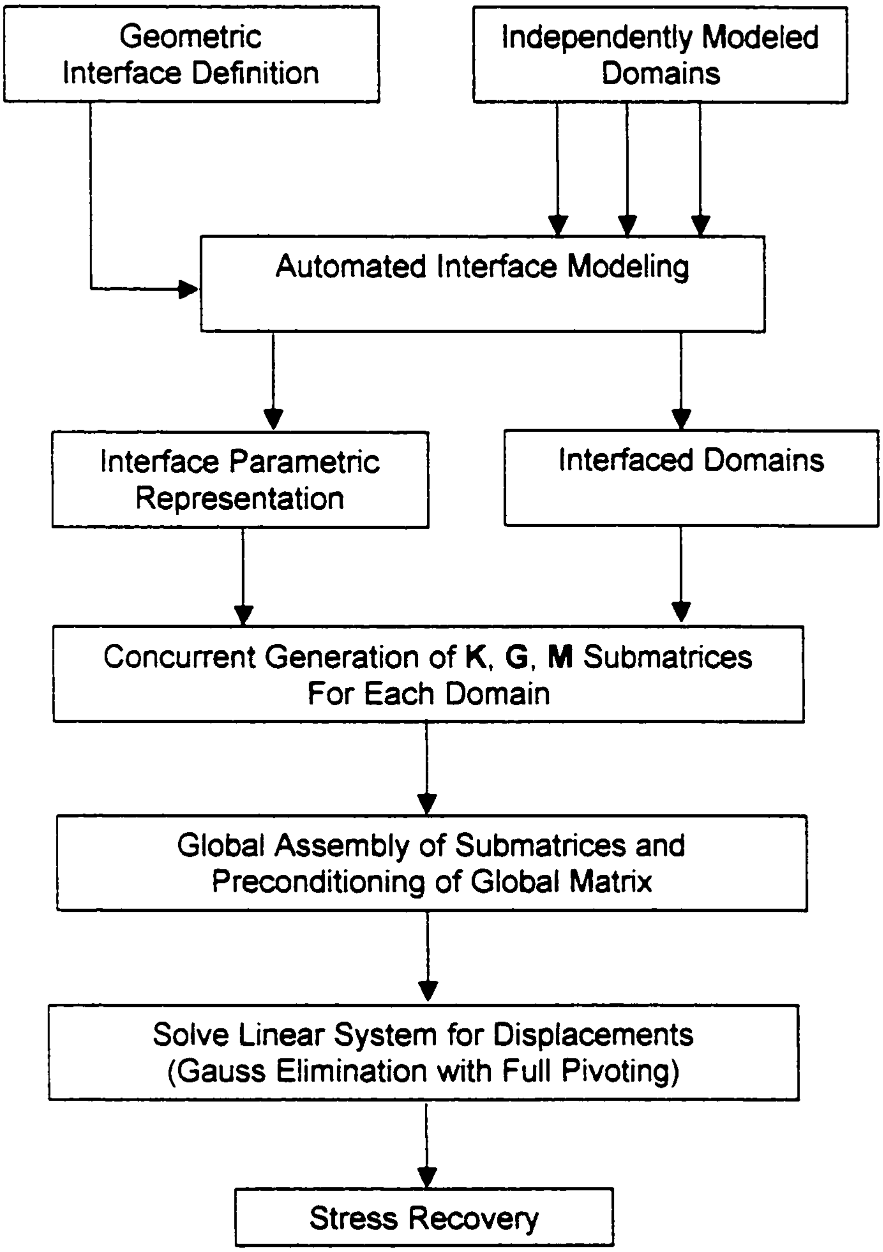

Geometric Interface Definition
Independently Modeled Domains
Automated Interface Modeling
Interface Parametric Representation
Interfaced Domains
Concurrent Generation of **K**, **G**, **M** Submatrices For Each Domain
Global Assembly of Submatrices and Preconditioning of Global Matrix
Solve Linear System for Displacements (Gauss Elimination with Full Pivoting)
Stress Recovery

Figure 4.1 Major steps in the solution process for the generalized global stiffness matrix.

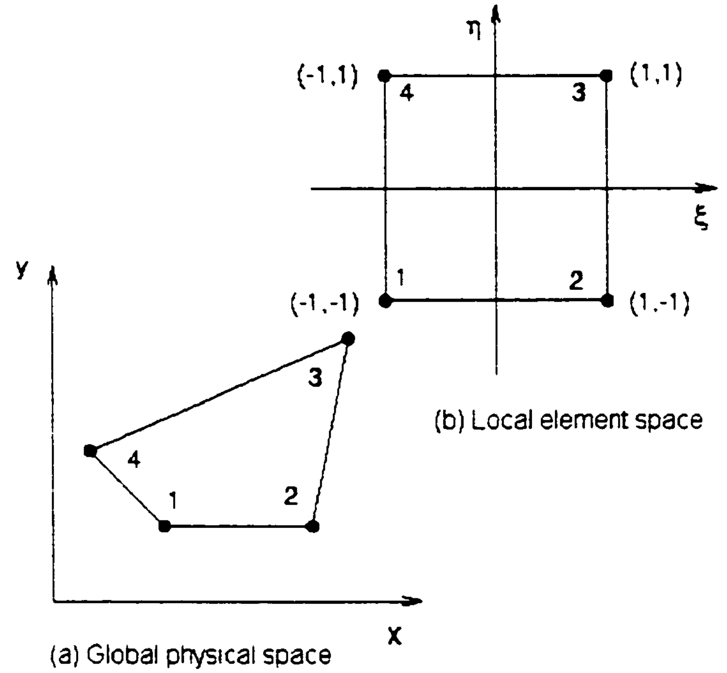


Figure 4.2 Four-node plane isoparametric quadrilateral element.

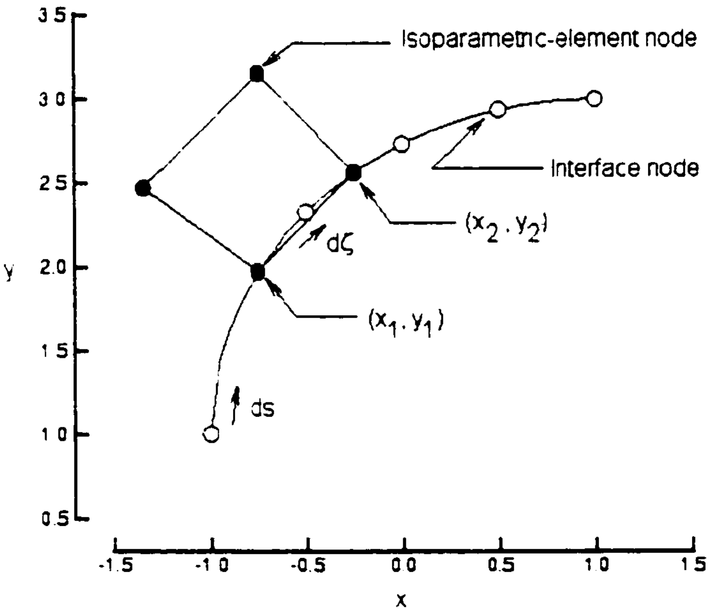


Figure 4.3 Relation between isoparametric and interface coordinates along an interface curve.

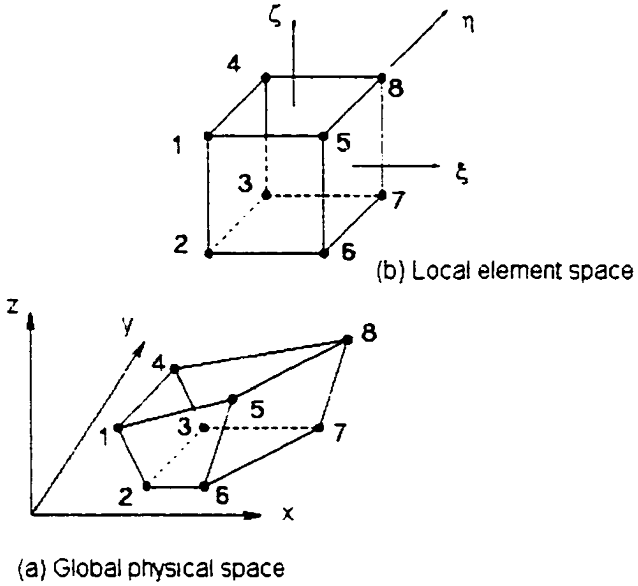


Figure 4.4 Eight-node isoparametric brick element.

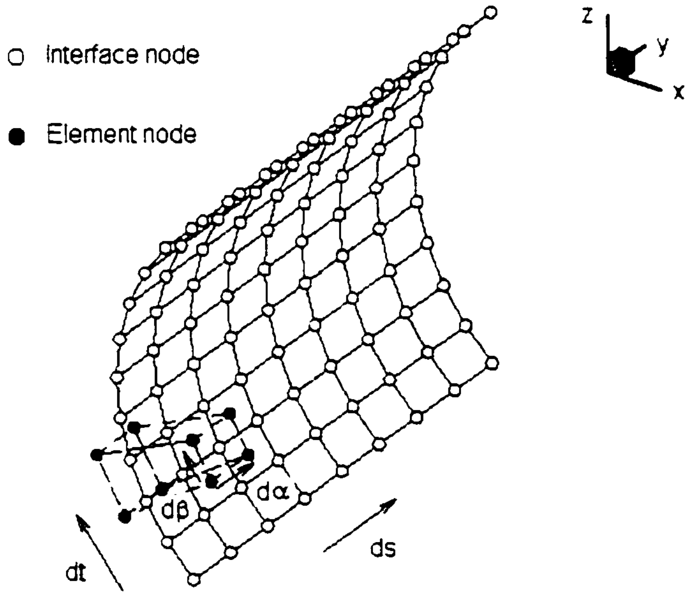
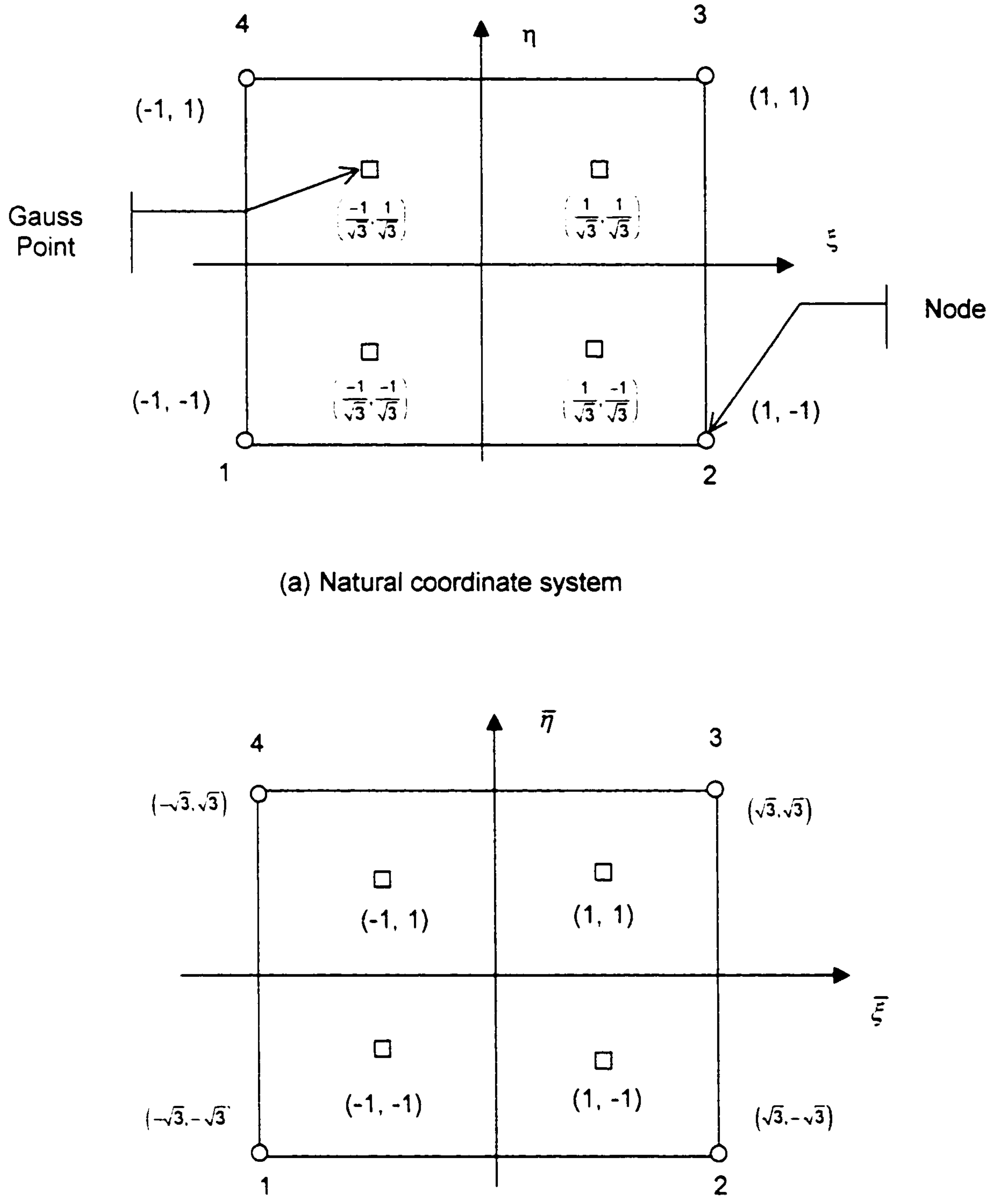


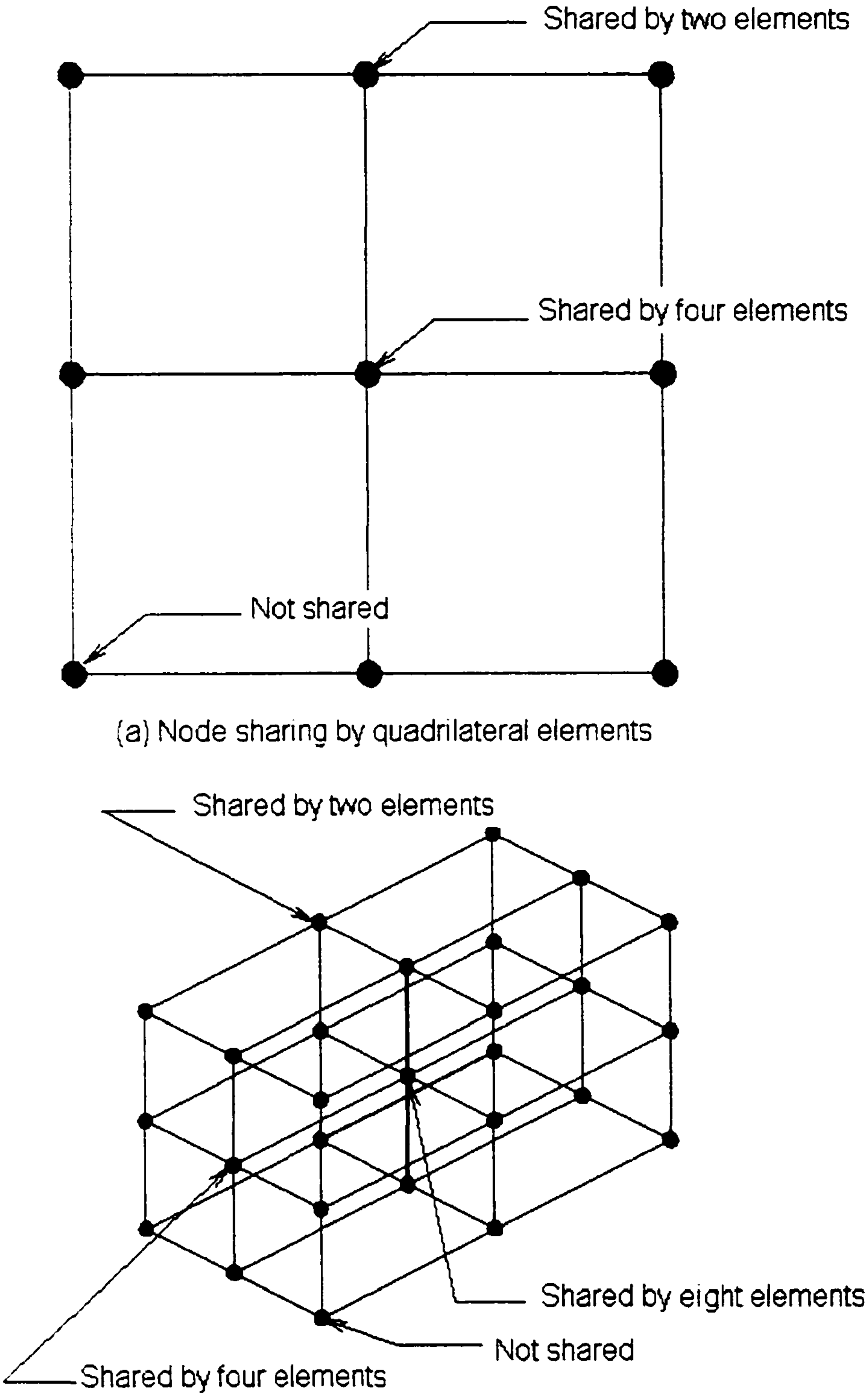
Figure 4.5 Relation between isoparametric and interface coordinates along an interface surface.


4
3
η
(-1, 1)
(1, 1)
Gauss Point
$\left(\frac{-1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right)$
$\left(\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right)$
ξ
Node
$\left(\frac{-1}{\sqrt{3}},\frac{-1}{\sqrt{3}}\right)$
$\left(\frac{1}{\sqrt{3}},\frac{-1}{\sqrt{3}}\right)$
(-1, -1)
(1, -1)
1
2
4
$\bar{\eta}$
3
$(-\sqrt{3},\sqrt{3})$
$(\sqrt{3},\sqrt{3})$
(-1, 1)
(1, 1)
$\bar{\xi}$
(-1, -1)
(1, -1)
$(-\sqrt{3},-\sqrt{3}$
$(\sqrt{3},-\sqrt{3})$
1
2

(a) Natural coordinate system

(b) Stress extrapolation coordinate system

Figure 4.6 Relationship between element nodes and Gauss sampling points for quadrilateral elements and two-point quadrature.



(a) Node sharing by quadrilateral elements

(b) Node sharing by brick elements

Figure 4.7 **Node sharing for quadrilateral and brick element types.**

CHAPTER V

APPLICATIONS AND RESULTS

5.1 Introduction

The results presented in this chapter include analyses of two-dimensional, three-dimensional, and curved membrane configurations. The two-dimensional results were obtained using a four-node quadrilateral element, whereas the three-dimensional results were obtained using an eight-node brick element. These two elements were introduced and briefly discussed in Chapter IV and are shown in Figures 4.2 and 4.4, respectively. Although these two elements tend to stiffen in bending, they are widely used and relatively straightforward to implement. The present research is devoted to interface methodology; therefore, use of sophisticated elements and detailed investigations of element properties *per se* are not justified as long as care is taken to distinguish between interface effects and element-related effects. It is prudent, however, to verify that the elements as implemented can represent constant strain states (i.e., pass a standard patch test).

5.2 MacNeal-Harder patch tests

More than a decade ago, MacNeal and Harder[103] proposed a set of standard problems to use for testing finite element behavior and accuracy. In the intervening years, this set of problems has been widely used to test existing and new elements. Several members of this set are known as patch tests, and accuracy on the patch tests is considered an important benchmark for an element type. In a general sense, a patch test compares the theoretical values of a standardized model to a finite element solution using the element being tested.

The patch test problem for two-dimensional plane stress quadrilateral elements is shown in Figure 5.1, and the corresponding specified displacements and theoretical solution is summarized in Table 5.1. To apply the patch test, values of the nodal displacements from the theoretical solution are specified on the element boundaries, displacements are computed at the interior nodes and strains and stresses are computed at each node and compared with the theoretical values.

Table 5.1 Patch test for four-node quadrilateral element

Specified displacement	Theoretical strain and stress
$u = 10^{-3}(x + \frac{1}{2}y)$	$\varepsilon_x = \varepsilon_y = \gamma_{xy} = 10^{-3}$
$v = 10^{-3}(y + \frac{1}{2}x)$	$\sigma_x = \sigma_y = 1333$ $\tau_{xy} = 400$

The patch test for the three-dimensional solid element is shown in Figure 5.2, and the corresponding boundary conditions and theoretical solution is summarized in Table 5.2.

Table 5.2 Patch test for eight-node brick element

Specified displacement	Theoretical strain and Stress
$u = 10^{-3}(x + \frac{1}{2}y + \frac{1}{2}z)$	$\varepsilon_x = \varepsilon_y = \varepsilon_z = \gamma_{xy} = \gamma_{yz} = \gamma_{zx} = 10^{-3}$
$v = 10^{-3}(\frac{1}{2}x + y + \frac{1}{2}z)$	$\sigma_x = \sigma_y = \sigma_z = 2000$
$w = 10^{-3}(\frac{1}{2}x + \frac{1}{2}y + z)$	$\tau_{xy} = \tau_{yz} = \tau_{zx} = 400$

It was found in the present research that the computed values were in complete agreement with the theoretical values for both the quadrilateral plane-stress element and the solid brick element.

5.3 Two-dimensional bar in tension

A basic configuration for benchmark investigation is the constant cross-section bar in tension with the material taken to be isotropic and linearly elastic. The analytical solution for this configuration with transverse roller boundary conditions at the attached end (i.e., boundary conditions needed to recover a uniaxial stress state) and imposed axial displacement at the free end of 0.05-inches is straightforward and widely known to be constant longitudinal normal stress, zero values for the other stress components, and linear displacement variation along the length of the bar. A rectangular two-dimensional plane stress configuration with the general layout and dimensions shown in Figure 5.3(a) was chosen for this investigation. The same figure also shows the finite-element model used for the computation. The bar was 5-inches long, 1-inch high and 0.1-inch thick. Longitudinal displacements of 0.05-inches were imposed on the right end-nodes. Material properties, similar to aluminum, were taken to be E = 10^7 psi and ν = 0.333. The finite element model, as shown in the figure, consisted of two subdomains; the left subdomain had 12 nodes and 5 elements whereas the right subdomain had 15 nodes and 8 elements. The interface geometry was a vertical line defined by the code-default minimum number of 5 interface nodes and located at x = 1.00-inch to the right of the fixed boundary. It is clear from the figure that the interior nodes of the subdomain models do not coincide at the interface.

Results from the plane stress computation are shown in Figures 5.3(b) and 5.3(c). These u and v displacement contours depict the plane stress solution for the prescribed bar response boundary conditions. It is clear from the figures that the displacement variations are linear in both directions and that the transverse displacement, v, is smaller than the u displacements by a factor of ν=0.333, in complete agreement with linear elasticity theory. Numerical values of the displacements and corresponding stresses agree nearly exactly with analytical values. These results verify the single domain and multiple domain software implementations for a uniaxial stress problem in two dimensions.

5.4 Three-dimensional bar in tension

A three-dimensional bar in tension was also chosen as a basic configuration for investigation to provide a benchmark test of correctness and accuracy for the algorithms and corresponding implementation of the three-dimensional solid element. The configuration chosen was a bar 10.0-inches in length with a square cross section of dimension 1.0-inch by 1.0-inch. The material properties were taken to be the same as for the two-dimensional bar: E = 10^7 psi, and ν = 0.333. The finite element model, shown as an exploded view in Figure 5.4(a), had a short domain with 50 nodes and 16 elements and a longer domain with 160 nodes and 81 elements. Roller boundary conditions were applied at the = 0 end with sufficient fixity to prevent rigid-body translation or rotation and not to generate any transverse normal stresses. The model had a planar interface between the domains, 1.00-inch from the fixed the fixed boundary in the positive x-direction, defined by a 4X4 array of 16 points. Longitudinal displacements (u-displacements) of 0.1-inch were applied on the x = 10.0-inch plane of the model. As in the two-dimensional case, this model should generate a uniaxial stress field and linear variations of longitudinal displacement.

A contour plot of computed u-displacement values is shown in Figure 5.4(b). From the figure, it is readily seen that the displacement contours were planar and the longitudinal displacement varied linearly along the length of the bar, as is known to be correct for a uniaxial stress solution for a uniform bar. Inspection of the file containing computed displacements showed the values were exact to eight significant digits throughout the bar. Additionally, the corresponding longitudinal stress value, as for the two-dimensional case, was determined to be essentially constant and of the expected magnitude. These results verify the single domain and multiple domain software implementation for a uniaxial stress problem in three dimensions.

5.5 Two-dimensional cantilever beam

A two-dimensional cantilever beam subject to transverse in-plane loading is a more stringent case to examine than a two-dimensional bar in tension since the stress state is not uniform along the beam length. The right-end boundary conditions for this case were implemented as an imposed transverse displacement of 0.05-inches for all nodes on the loaded end of the beam. The configuration, material properties, and model were the same as the two-dimensional bar-in-tension case previously discussed. These conditions simulate a beam solution. The computed transverse displacements and outer-fiber, longitudinal stresses along the outer surface fiber of the beam are shown on Figure 5.5 as filled circles. The corresponding results from strength-of-materials theory (solid line) and from a single-domain finite element analysis (open symbols) are also shown for comparison. The transverse displacement results are normalized by the tip deflection, the longitudinal position along the beam is normalized by the beam length, and the bending stress results are normalized by the maximum bending stress at the fixed end of the beam (for this case, σ_{ref} = 30 000 psi). The deflection results indicate the cubic behavior of the transverse displacement along the beam. The single domain finite element solution and the multiple domain finite element solution correlate exactly with the strength-of-materials beam solution for deflections. This is true even at the interface for the multiple domain solution.

The bending stresses plotted in Figure 5.5 show the strength-of -materials theory as a solid line. The finite element stresses for both the multi-domain and the single-domain models are evaluated at the Gauss points. These Gauss-point stresses are then extrapolated to the nodes by means of the procedure discussed in Chapter IV. The values obtained by extrapolating to the nodes are shown in Figure 5.5 as '+' symbols of the multi-domain method and open circles for the single-domain model. From the figure, it is clear that the single-domain and multi-domain values correlate well except at the interface. The finite-element stress values extrapolated to the nodes correlate well with the corresponding strength-of-materials values except near the ends of the beam. The longitudinal density of elements in the finite element models was not particularly refined, and the differences between the FEM results and strength-of-materials was attributed to the longitudinal mesh coarseness and the consequent lack of stress "information" available for the extrapolation procedure near the ends of the beam. The validity of this conclusion was investigated by extrapolating the Gauss-point stresses directly to the beam edge.

The Gauss-point stresses extrapolated to the beam edge are shown as filled triangles for the multi-domain model and as open squares for the single-domain model in Figure 5.5. Thus, results for a given element are associated with each pair of points (either open squares or filled triangles). The figure shows that there is good correlation between the single-domain and multi-domain values extrapolated to the beam edge.

A striking feature is that the values extrapolated for each pair of Gauss points from the same element "bracket" the strength-of-materials line. Evidently, if values extrapolated from the two Gauss points were averaged, the resulting mean values would fall very nearly on the strength-of-materials line. This result suggests that a possible way of obtaining more accurate stress, especially near the boundary of a domain, by direct extrapolation followed by averaging and taking the resulting value to be located at the midpoint between the locations of the extrapolation points.

A natural question that occurs is why not simply sample the stresses at the element centroids as advocated by some researchers for bilinear elements. One important disadvantage of this approach is that only a single stress value is obtained within an element. The only way that extrapolations to the boundary can then take place is to include a centroid stress-value from an adjacent interior element.

A variation on this theme is, in fact, practiced in the CFD arena. Flow quantities of interest are obtained at the cell centroids in the finite-volume approach and widely available post processor such as PLOT3D and its successor, FAST[110] provide the values at other locations via interpolation and extrapolation. The application of CFD post-processing software based on cell-centered data to analyze FEM results evaluated at element centroids is a promising approach.

Finally, part of the differences in stress results between multi-domain and the single-domain models at the interface are attributed to the fact that the interface theory is an integrated formulation (see equation (2.4) of chapter II) and stress is expected to be continuous across the interface in an integral sense, but not necessarily continuous in a pointwise sense.

5.6 Three-dimensional cantilever beam

The same finite-element model used for the three-dimensional bar was investigated using transverse tip loads to obtain a cantilever beam in a manner similar to the two-dimensional cantilever beam case.

Displacements and surface compressive stresses along the compression side are shown in Figure 5.6 together with corresponding strength-of-materials theory and single-domain finite element results. The displacement values are normalized by the tip value of v_{tip} = 0.1-inch and the stresses are normalized by the maximum bending stress at the fixed end of the beam (for this case, σ_{ref} = 15 000 psi). The points shown plotted in Figure 5.6 are average extrapolated values for a given x-plane, averaged over y and z. It is clear from the figure that there is excellent agreement between the displacements for all the methods, and that the overall trends for the stresses were similar for the various methods. As was the case for the two-dimensional beam, there was good agreement for the bending stress between the finite-element and strength-of-materials results except for differences at the two ends of the beam.

As with the two-dimensional case, the differences in nodal stress values at the ends was attributed to coarse mesh size in the longitudinal direction with resulting limited "information" for the stress recovery procedure as implemented. Values of stress at the Gauss points were also extrapolated directly to the beam surface for comparison. In this case, there were two extrapolated values obtained per x-location in an element, these values were averaged, and are the points shown in the figure. There were only slight differences between the multi-domain and single-domain results at the interface, and these were attributed to the formulation being an integral formulation wherein continuity is expected in an integral sense (equation (2.4) of Chapter II).

The behavior of the values obtained in this way was similar to that observed for the two-dimensional cantilever beam previously discussed. One difference between the two-dimensional and the three-dimensional results is a tendency for the directly extrapolated values to be larger that the strength-of-materials values over much of the beam length rather than consistently "bracketing" the strength-of-materials results as was true for the two-dimensional beam. The reason for this difference in behavior is not completely understood.

5.7 Flat plate with circular hole loaded in tension

A flat rectangular finite plate having a centrally located circular hole is a configuration of considerable interest because it exemplifies many of the characteristics that occur in a variety of practical applications. Civil engineering structures, such as bridges or building frames that are riveted or bolted together, embody repeated patterns of plates with holes. For many years, riveted steel plates were the standard construction method in shipbuilding. Although this procedure has been generally replaced by welding, the bulkheads and decks must still have hatch openings and these are plate-with-hole patterns. Today, metal aircraft are routinely constructed of sheet metal riveted together and supported by ribs and spars also riveted together. Welding and more exotic construction methods and materials are replacing this, but many aircraft substructures are still assembled by the method of rivets or screws. Additionally, the fuselages of aircraft have door and window openings, which are essentially plate-with-hole patterns. Because this configuration-type occurs so often in practice, it has been studied from early on in structural mechanics.

One of the first solutions for this configuration type was obtained in 1898 by G. Kirsch[104] for a doubly infinite two-dimensional plate having a circular hole and loaded in tension in one direction. The solution by Kirsch is now treated as a standard problem in elasticity texts (e.g., Timoshenko and Goodier[94]). Making use of a successive approximation procedure, Kirsch's results were extended by Howland[105] to allow for an infinite strip of finite width having a hole of significant diameter compared to the strip width. Howland's paper, in part motivated by the needs of the shipbuilding industry in the 1930's, presented an early definitive solution to the problem and tabulated stress concentration factors for selected values of the problem parameters. Since the appearance of Howland's work, several treatises have been published concerning stress concentration factors that include the plate with hole (e.g., Peterson[106] and Savin[107]). It is common practice nowadays for standard mechanical engineering design textbooks to have charts of stress concentration factors that include the plate with a circular hole (e.g., Spotts[108] and Shigley[109]). The results presented in the present research include both two-dimensional and a three-dimensional finite element solutions for this important case. Plane stress assumptions are imposed on the two-dimensional finite element models. The configurations selected for investigation are shown in Figures 5.7(a) and 5.21.

5.7.1 Reference solutions

For comparison, single-domain finite element reference solutions were generated for the two-dimensional plane stress case (see Figure 5.7(b)) and the three-dimensional elasticity case (Figure 5.21(c)). Modeling the plate with a hole using a single-domain model requires a transition from a circular topology near the hole to a rectangular topology near the plate boundary. While readily performed by many commercial pre-processing codes such as PATRAN, it is desirable to have regular-shaped elements near the hole. The finite element model herein is defined by the number of radial spokes of nodes from the hole, the number of rings of elements around the hole, the number of elements in the remaining x-direction rectangular region to the right of the hole, and the number of elements in the remaining y-direction rectangular region above the hole. These four numbers are used to define the mesh (e.g., see Figure 5.7(b)). In the three-dimensional models, the number of elements in the thickness direction (here taken as the z-direction) also needs to be specified. Solutions obtained using these single-domain finite element models serve as reference solutions for the multiple-domain cases.

In addition, the analytical solution of Howland[105] is used for comparison. Designers often work with stress concentration factors that are multipliers of the nominal stress across net section area. For the case considered (w/D = 5), Peterson[106] gives a stress concentration factor of K = 2.51. This stress concentration factor includes finite width effects and is based on nominal stress across net section area.

5.7.2 Two-dimensional plate with circular hole

The configuration dimensions and material properties used in the present investigation are shown and summarized in Figure 5.7. The plate was 16.0 inches long and 8.0 inches in width. The hole was centrally located and was 1.6 inches in diameter. The material properties were those of aluminum: Young's modulus E = 10^7 psi and Poisson's ratio $\nu = 0.333$. The plate was constrained to have displacement boundary conditions corresponding to an in-plane tension load and free to expand in the transverse direction. Due to symmetry, only one quarter of the plate was analyzed. For this loading, a stress gradient near the hole boundary along the horizontal symmetry plane is expected. The stress field away from the hole should become uniform. Results from three investigations are presented and discussed: (i) a mesh-refinement study, (ii) an interface location study, and (iii) an interface-shape study. The stress results presented for this case were normalized by the nominal stress over net section area, which for the present configuration and loading was $\sigma_{ref} = 156\ 250$ psi.

5.7.2.1 Mesh refinement

The finite element models for the mesh-refinement investigation are shown in Figure 5.8, which gives the models before and after remodeling. The finite element meshes of each domain were independently generated to conform to the natural topology of the respective subdomain. Around the hole, a circular topology was used and for the rest of the plate, an L-shaped rectangular domain was used. As seen in the figure, this resulted in a significant gap between the two models and thus presented an extreme test of the automatic remodeling methodology developed for this research. Results from the automatic remodeling phase are labeled as "interfaced domains" in the figure and demonstrate the ability of the methodology to perform extreme remodeling. The remodeling resulted in moving the nodes on the interface boundary of the L-shaped domain to the boundary of the circular domain. This automatic remodeling capability represents a unique new feature for interface technology. It permits the use of convenient modeling approaches for subregions and eliminates the need for exact boundary modeling along the interface.

The coarse model had 15 nodes and 8 elements in the circular domain and 29 nodes and 18 elements in the L-shaped domain. The medium mesh had 35 nodes and 24 elements in the circular domain and the same number of nodes and elements in the L-shaped domain as for the coarse model. The fine mesh had 63 nodes and 48 elements in the circular domain the same number of nodes and elements in the L-shaped domain as for the coarse model. The most refined case analyzed, designated the reference case, had 99 nodes and 80 elements in the circular domain and 96 nodes and 75 elements in the L-shaped domain.

For this study, the interface definition was taken as a set of points coincident with the outer boundary of the circular region. With this choice, elements in L-shaped domain near the interface became distorted during the re-modeling process. Since large stress-gradients are expected to occur near the hole but not away from the hole, less error due to element distortion was expected in the L-shaped domain that was expected in the circular domain nearer to the hole. It is more important to have regularly shaped elements near the region with the stress gradient.

Contours of field quantities computed in this investigation are shown in

Figures 5.9 and 5.10, which give v-displacement contours and longitudinal stress contours respectively for the coarse and refined meshes. Also shown are results for a single-domain finite-element computation for comparison. The single-domain model had mesh-resolution comparable to the reference two-domain interface case.

Examination of the displacement contours shows remarkable agreement between the single-domain results and the multiple-domain interface reference results. The results indicate no discontinuity in the displacement contours across the interface. The coarse-mesh results had minor differences compared to the single-domain, attributed to the mesh resolution and are not considered an effect of the interface methodology. For example, the slight gaps in the coarse-mesh contour plot indicate the geometry error associated with a coarse mesh of linear elements used to model a curved boundary.

The stress contours show an overall agreement between the multiple-domain interface reference case and the single-domain reference case except for some isolated discontinuities near the interface. Some of the discrepancy is due to the stress recovery procedure based on averaged nodal stresses. Examination of the contours in the vicinity of $y = 0$ showed that there was excellent agreement between these two cases; an important design result since the $y = 0$ line near the hole is where maximum stress occurs and therefore where failure is likely to occur.

More detailed results for the important $y = 0$ line are shown in Figure 5.11 which gives a line plot of stress concentration factor based on net section area. The plot shows results for the coarse, medium and fine meshes. Included on the plot for comparison are values from Howland's paper[105] and a single-point stress concentration factor from Peterson's treatise.[106] Examination of the figure indicates excellent agreement between the fine-mesh results, the single-domain results, and Howland's values. All of the results shown are in good agreement, and differences shown by the coarse and medium grids were attributed to mesh resolution and not to the interface methodology.

5.7.2.2 Effect of interface location

Finite elements for the interface location study are shown in Figure 5.12. In this study, the outer radius, R, of the circular domain was varied to examine the interface location effect on the plane stress solution. Three finite element models representing circular interfaces located at an R of 1.5-inches, 2.0-inches and 3.0-inches from the hole center are shown in Figure 5.12. All of the physical dimensions and material properties were kept the same as shown in Figure 5.7. As was the case for the mesh refinement study, the interface definition was taken to lie on the outer boundary of the circular domain, and this kept element distortion confined to the L-shaped domain. However, for small values of R, the interface boundary approaches the expected area of large stress gradient near the hole. The three finite element models for this interface location study each had 99 nodes and 80 elements in the circular domain and 96 nodes and 75 elements in the L-shaped domain. The spatial distribution of these nodes and elements changed for each model.

The v-displacement contours for the three cases are given in Figure 5.13, and an examination of the results showed that there were no significant differences due to the variation of interface location for this configuration. Contours of the corresponding longitudinal stress are shown in Figure 5.14. Examination of these contours revealed general overall agreement for the three models except for isolated discontinuities in the contours across the interface.

A line plot of normalized longitudinal stress is given in Figure 5.15 for the important $y = 0$ line. Included on the figure are results from Howland's previously cited analytical investigation, results from the single-domain finite element computation, and a single-point result on the hole boundary taken from Peterson's treatise previously cited. There is excellent agreement between all the results, and this observation fosters confidence in the present methodology as well as indicating a general insensitivity of the method to interface location (provided, of course, that the interface is located beyond the region of maximum stress gradient very near the hole).

5.7.2.3 Effect of interface shape

The previous two analysis studies used a circular interface coincident with the outer boundary of the circular subdomain and wherein elements of the L-shaped domain suffered distortion due to the remodeling process. In order to have further confidence in the methodology, it was important to investigate the effect of interface shape on computed results. An interface that conformed to the L-shape boundary was defined and the remodeling resulted in a slight distortion of the elements from each domain that were adjacent to the interface. Since the interface in this instance had a sharp corner, the effect of maintaining or removing the sharp corner was also investigated. The models used for this are shown in Figure 5.16 where the independent domain meshes are plotted together with the meshes produced by the automatic remodeling with and without retention of the node at the sharp corner.

Contours of computed field quantities from the analyses are given in Figures 5.17 and 5.18, which show v-displacements and longitudinal stress, respectively for the single-domain reference and the two interface cases. Examination of the displacement contours revealed only minor differences between the three results. The longitudinal stress contours were quite similar near the hole; however, isolated discontinuities at the interfaces occurred similar to those arising in the previous cases.

A plot of normalized longitudinal stress along the $y = 0$ line is given in Figure 5.19, which also shows Howland's and Peterson's results for comparison. It is evident from the figure that the results were relatively insensitive to the interface shape. For this case, there was a slight difference between values computed within each domain just at the interface. Such a discontinuity was not observed for the models with circular interfaces. Although the exact reason for this effect is not known for sure, it may have been due to the distortion of some elements in the critical circular region, a possibility to bear in mind when one models similar applications.

5.7.3 Three-dimensional plate with circular hole

5.7.3.1 Configuration details and problem definition

The configuration dimensions for this case are shown in Figure 5.20. Because of symmetry, it was only necessary to consider one quadrant of the full plate, as is shown in the figure. The configuration had material properties similar to the two dimensional case: $E = 10^7$ psi and $\nu = 0.333$. The plate was taken to be 1.00-inch thick and the hole was chosen to be 1.00-inch in radius. Since the full plate was 8.00-inches wide, the hole-diameter to plate-width ratio was therefore 0.25. The reference stress use to normalize the stress results presented was σ_{ref} = 166 667 psi.

For this case, the circular domain model had 64 nodes and 27 eight-node brick elements. The L-shaped domain had 160 nodes and 81 eight-node brick elements. The interface was defined by 66 points in an 11 by 6 array having the geometrical form of a cylindrical surface. The original subdomain models, interface surface, and subdomain models after automatic remodeling are shown in Figure 5.21 together with the single-domain reference model.

5.7.3.2 Results for tension loading

The model was subjected to displacement constraints that represented tension loading in a manner similar to the two-dimensional case. Displacement contours for the multi-domain and single-domain models are shown in Figure 5.22. These contours are shown in a two-dimensional view with the z-axis pointing out of the paper, but they are three-dimensional contours.

Because of the symmetry of the configuration and boundary conditions, it was expected that the three-dimensional displacement contours should exhibit negligible variation in the z-direction. An examination of the contours in Figure 5.22 reveals that this was indeed the case, as no discernible variation in this direction could be seen. The sharpness of the contour lines between the contour levels is an indication of the lack of variation in the depth, or z-direction. Examination of the actual computed values also confirmed this lack of z-variation.

The contours shown in this figure are quite similar to the two-dimensional model results shown in Figure 5.17.

Longitudinal stress contours for a similar view with the z-direction pointing out of the paper are given in Figure 5.23 for the single-domain reference model and the two-domain interfaced model. The general characteristics of these contour plots are similar to the two-dimensional results shown in Figure 5.18. There is some "fuzziness" in the contour lines for the stresses that indicate a slight variation in the stress contours in the z-direction. This slight variation was attributed to two sources. First, small variations in the displacements are amplified in the differentiation process to obtain the stress. Second, the stress recovery procedure is based on an extrapolation and nodal averaging technique that could introduce small variations.

A line plot of longitudinal stress along the y = 0 plane averaged in the z-direction is shown in Figure 5.24. The figure includes the multi-domain interfaced results, the single-domain reference results, and values from Howland's analytical solution of the semi-infinite two-dimensional flat plate. The overall agreement between the results shown is excellent in the same way as the two-dimensional results.

5.8 Curved membrane with circular hole

5.8.1 Background

A curved membrane with hole is a configuration of considerable interest to aerospace researchers because of its obvious similarity to a fuselage section with window or other opening. Although the four-node quadrilateral element has only plane stress capability and therefore does not support out-of-plane deformations, a curved-membrane panel under in-plane loading can be analyzed with the present methodology. This configuration also provided an opportunity to test the automated remodeling capability for an interface having out-of-plane curvature.

5.8.2 Configuration and problem definition

The layout and dimensions of the panel investigated is shown in Figure 5.25. The aspect ratio of the plot distorts the representation and is somewhat deceiving; in actuality, the panel was a circular section having a 20-inch radius of curvature. The panel arc subtended an angle of approximately 11.5-degrees, which made the arc length distance about 4.03-inches. The width of the panel was 8.0-inches and the hole was 1.6-inches in diameter. The material properties were the same as used for the previous configurations: $E = 10^7$ psi and $\nu = 0.333$. The panel was constrained to have displacement boundary conditions corresponding to a tension load in the longitudinal direction (y-direction as given in the figure). The reference stress used to normalize stress results for this case was σ_{ref} = 156 250 psi.

5.8.3 Finite-element models

The model used for the analysis is shown in Figure 5.26. The independently modeled subdomains are shown in Figure 5.26(a) and consist of a curved L-shaped subdomain and a curved circular subdomain. The interface was taken coincident with the outer ring of the circular domain and was defined by 11 points. The circular subdomain had 99 nodes and 80 elements, whereas the L-shaped subdomain had 121 nodes and 96 elements. The remodeled domains are shown in Figure 5.26(b), an examination of which showed that the remodeling gave results similar to the flat cases thus verifying the remodeling capability for interfaces having out-of-plane curvature.

5.8.4 Computed results

Longitudinal displacement contours are shown in Figure 5.27(a) and corresponding stress contours are given in Figure 5.27(b). The contours for the displacement and stress display characteristics similar to the two-dimensional flat plate results (cf. Figures 5.9 and 5.10).

Since this configuration was analyzed chiefly to investigate the automatic remodeling capability, previously published analytical or computational results were not sought in the literature for comparison. It was thought that the qualitative comparison between the flat-plate contour plots and the panel contour plots were sufficient to verify the methodology.

5.9 Two-dimensional plate with rectangular hole or re-entrant notch

The cases discussed so far have had either no geometrical variation (uniform bars and beams) or have had continuous geometrical transitions (two-dimensional plate and panel; three-dimensional plate). The present and subsequent cases have discontinuous geometrical variation and sharp re-entrant corners in the vicinity of the interface. That is, the interface boundary does not necessarily extend over the entire boundary on both subdomains models.

5.9.1 Configuration and problem definition

The two-dimensional configuration studied was a notched flat plate as depicted in Figure 5.28. Because of symmetry, the section shown also represents a quadrant from a plate with a rectangular hole. The plate was 8.00-inches long and 4.00-inches wide. The corner had a 1.50-inch square notch leaving a tab of dimensions 1.5-inch by 2.5-inch. The material properties were taken to be the same as the previously discussed cases. The plate was constrained to have displacement boundary conditions corresponding to a tension load in the longest direction (y-direction in the figure). The reference stress used to normalize stress results for this case was based on nominal stress over net section area and for this case was σ_{ref} = 200 000 psi.

5.9.2 Finite element model

The models are given in Figures 5.29(a) – 5.29(d). The single-domain reference model in Figure 5.29(a) had 325 nodes and 316 elements giving a reasonably high resolution model intended to capture effects in the corner region where high gradients in the computed quantities were expected. Two different interfaced model types were investigated: one with the edge nodes matching and one with the interface edge nodes not matching. For the model type with non-matching interface edge nodes, the interface node corresponding to the node on the larger subdomain was included for one calculation and not included for another calculation.

The model having the edge nodes match is shown in Figure 5.29(b). The other cases are shown in Figures 5.29(c) and 5.29(d). The smaller subdomain (tab) for each of these three models had 91 nodes and 72 equally sized elements. The large subdomain for the edge-node matched model had 238 nodes and 208 equally sized elements, giving a reasonably dense mesh in both subdomains. The large subdomain for the edge-node non-matched model had 182 nodes and 156 equally sized elements, again giving a fairly dense mesh.

5.9.3 Computational results

Plots of longitudinal displacement contours are given in Figure 5.30. Corresponding longitudinal stress contours are shown in Figure 5.31, and a line plot of longitudinal stress near the y = 1.5-inch line are shown in Figure 5.32. The displacement contours for the interfaced models show general characteristics similar to the contours for the single-domain reference model. However, there are noticeable differences in the vicinity of the corner and interface for the matching vs. non-matching models. The matching model is quite similar to the single-domain case, whereas the non-matching models do not agree with the single-domain results.

The contour plots of the longitudinal stress in Figure 5.31 show that the single-domain and matched edge-node have similar characteristics, but there are distinct differences between the single-domain and the non-matched edge-node cases.

Examination of the line plot of longitudinal stress given in Figure 5.32 shows excellent agreement between the single-domain results and the edge-node matched results except for slight excursions between the x = 3-inch and 4-inch positions. The results from the two multi-domain cases wherein the edge nodes were not matched showed relatively large excursions about the single-domain curve throughout the plotted range. Since the present methodology was developed based on the tacit assumption of matching edge-nodes, these results are not too surprising.

5.10 Three-dimensional bar with abrupt size reduction

5.10.1 Configuration and problem definition

The configuration investigated was a solid bar of rectangular section having an abrupt section reduction from a 2.0-inch by 2.0-inch size to a 1.0-inch by 1.0-inch size. Figure 5.33 shows one quadrant of the configuration, which was sufficient for analysis due to the geometrical symmetry and symmetry of the loading. The material properties of the bar were taken to be the same as the cases previously discussed, i.e., $E = 10^7$ psi and ν =0.333. The overall length of the bar was 10.0 inches; the larger-sectioned piece was 1.0 inch long and the smaller-sectioned piece was 9.0 inches long. The reference stress used to normalize stress results for this case was the nominal longitudinal stress in the long piece, which for this case was σ_{ref} = 100 000 psi.

5.10.2 Finite element models

The reference model was a single-domain model with 131 nodes and 52 elements as shown in Figure 5.34. The 9.00-inch part of the interfaced model had 81 nodes and 36 elements of the brick type. Two models for the short piece were investigated: one having nodes that matched the adjacent surface edges of the long piece and one that did not match the adjacent surface edges of the long piece. The short model that matched the surface edges had 72 nodes and 25 elements of the brick type, whereas the non-matching short part had 98 nodes and 36 elements (also of the brick type).

In the present three-dimensional case, there is a great deal of scope in how the interface itself can overlap the nodes from adjacent subdomains. For the present research, two interface geometries were defined: one with its outer edges matching corresponding mesh lines on each subdomain and one with its outer edges not coinciding with any mesh line of the subdomains. These two interfaces are shown in Figure 5.34 as the 0.5-inch by 0.5-inch interface and the 0.6-inch by 0.6-inch interface, respectively.

Possible variations between the interfaces and interface-node assignments on the larger subdomain are shown in Figures 5.35(a) – 5.35(c). The cases of non-coincident edges are given in Figures 5.35(a) and 5.35(b). In these figures, the small circles denote points defining the 0.6-inch by 0.6-inch interface. The large circles denote nodes assigned to the interface from the longer subdomain. The squares denote nodes assigned to the interface from the shorter subdomain.

In Figure 5.35(c) the same symbols denote the respective interface points and node assignments, except the interface is 0.5-inch by 0.5-inch and matches geometrically the end of the longer subdomain.

5.10.3 Computed results

Longitudinal displacement contours on a slice parallel to the x-z plane and at y=0 (the center plane of the configuration) is shown in Figure 5.36. The single-domain results are shown in Figure 5.36(a), the matched edge-node results are in Figure 5.36(b), the edge-node exterior to the longer subdomain face is in Figure 5.36(c), and the edge-node interior to the longer subdomain is in Figure 5.36(d). The non-coincident results are each similar to each other but differ from the single-domain results. On the other hand, the coincident-edge results are quite similar to the single-domain results.

Longitudinal stress contours in the y = 0 plane are shown in Figure 5.37. Figures 5.37(a)-5.37(d) show results from models corresponding to the displacement contours shown in Figure 5.36. The strength-of-materials theoretical stresses are shown in Figure 5.37(3) for comparison. Strength-of-materials results for a geometry having step size reduction in section are unrealistic and are typically supplemented by stress concentration factors for design work. For a geometry of this nature, the finite-element solutions are likely to be a better representation of reality. Each of the cases in Figure 5.37 captures the general nature of the stress in the longer subdomain and the expected stress reduction in the shorter piece. There are also large stress gradients near the sharp corner as evidenced by the "bunching" of contour lines in the vicinity of the corner.

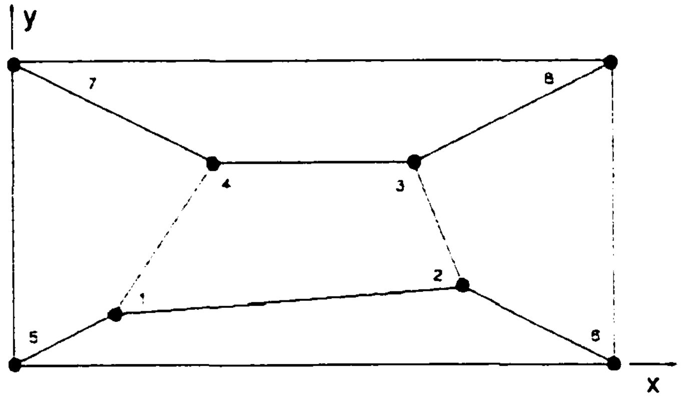


Figure 5.1 MacNeal-Harder two-dimensional plane-stress patch test layout.

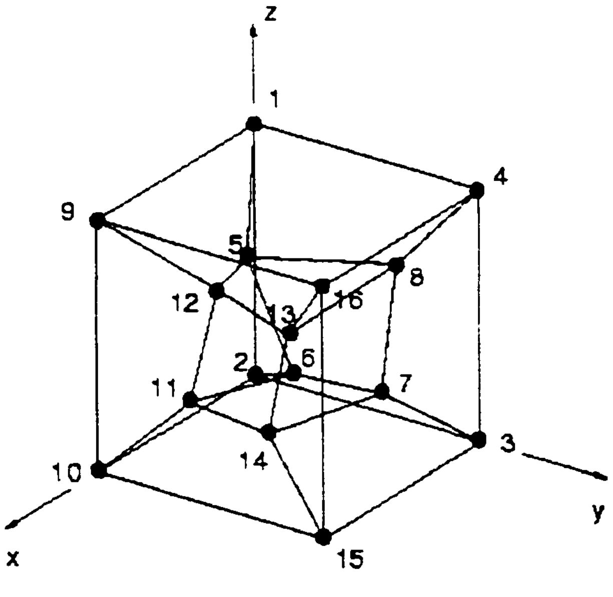
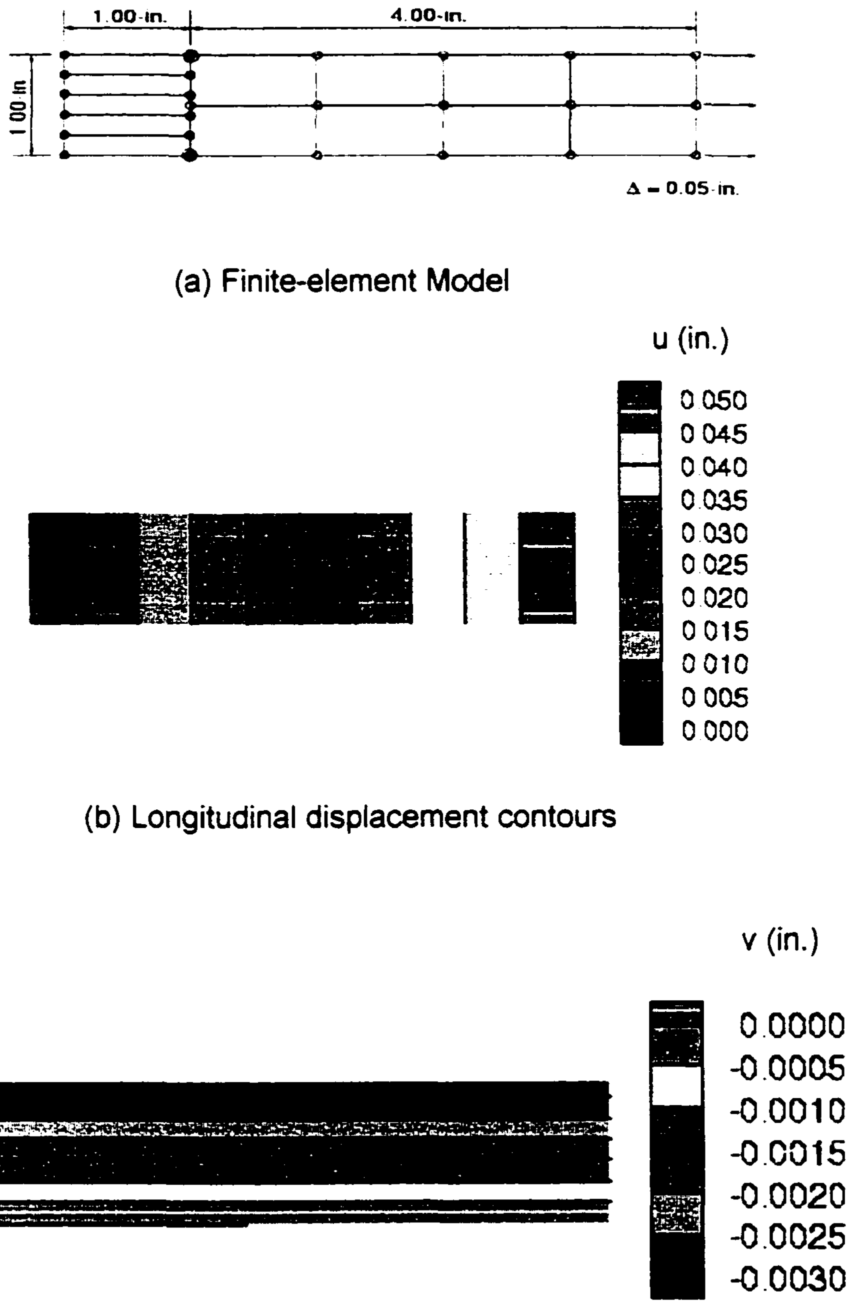


Figure 5.2 MacNeal-Harder three-dimensional patch test layout.

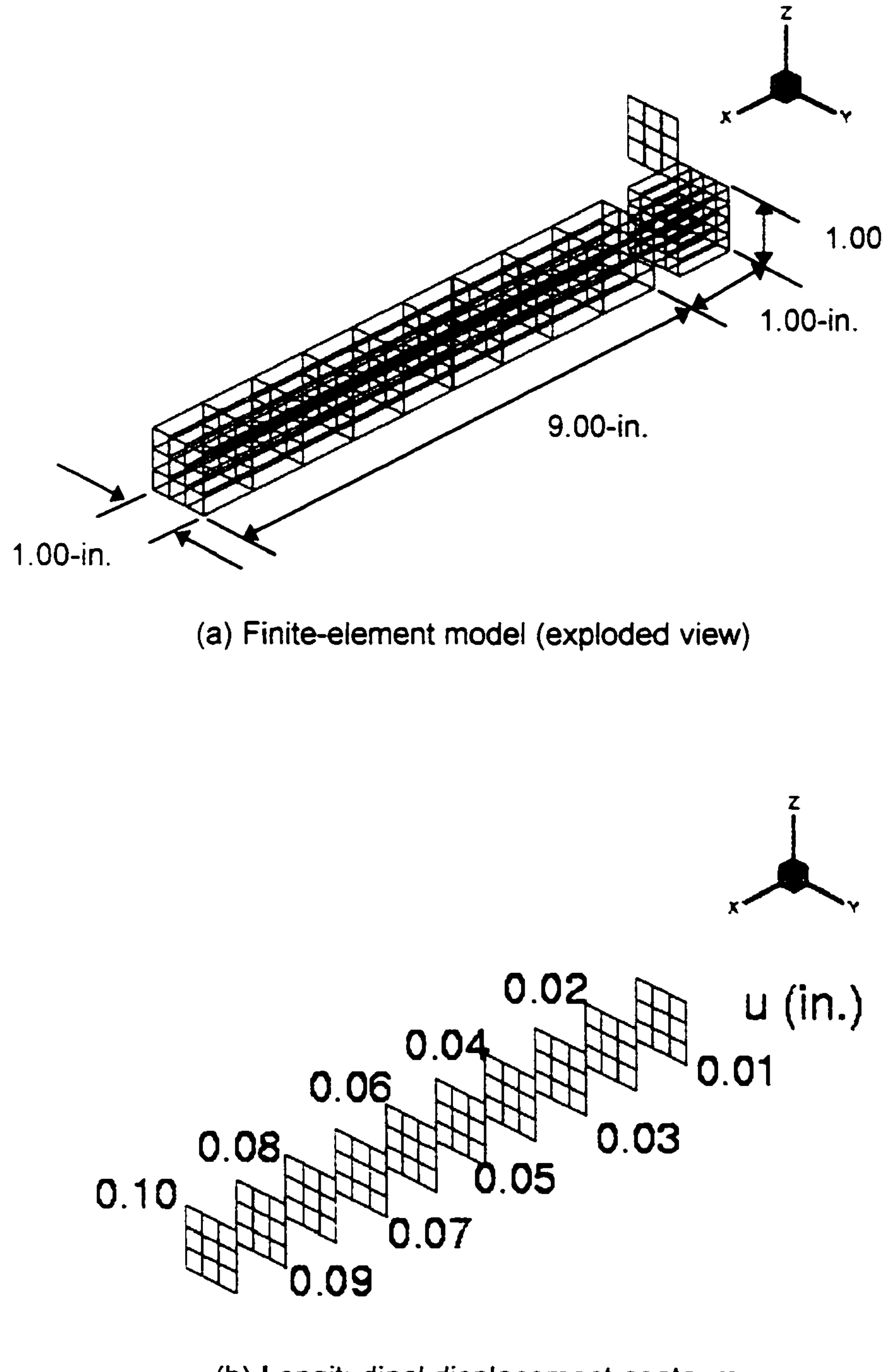


(a) Finite-element Model

(b) Longitudinal displacement contours

(c) Transverse displacement contours

Figure 5.3 Two-dimensional bar loaded in tension.



(a) Finite-element model (exploded view)

(b) Longitudinal displacement contours

Figure 5.4 Three-dimensional bar loaded in tension.

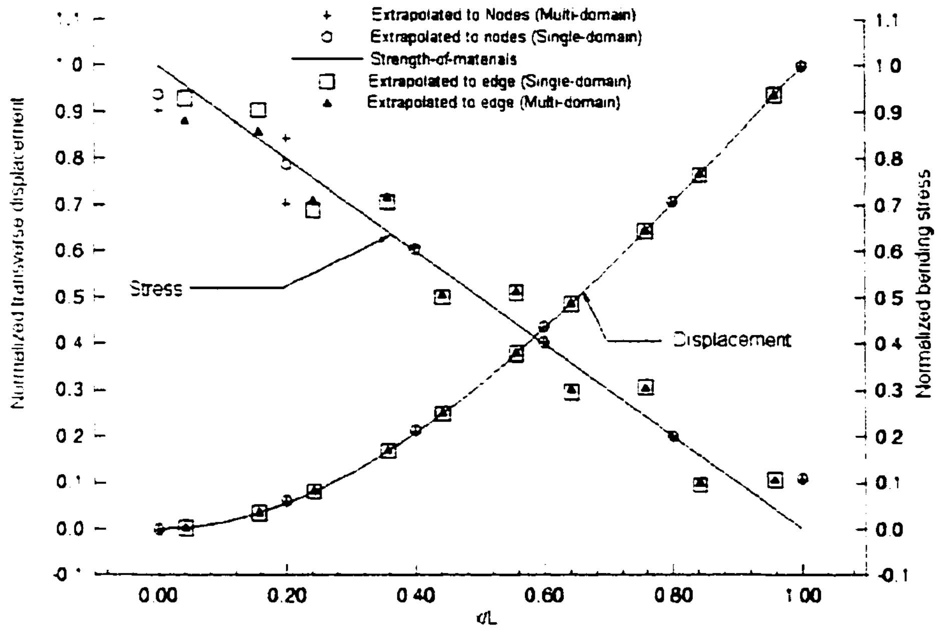


Figure 5.5 Transverse displacement and bending stress for two-dimensional cantilever beam.

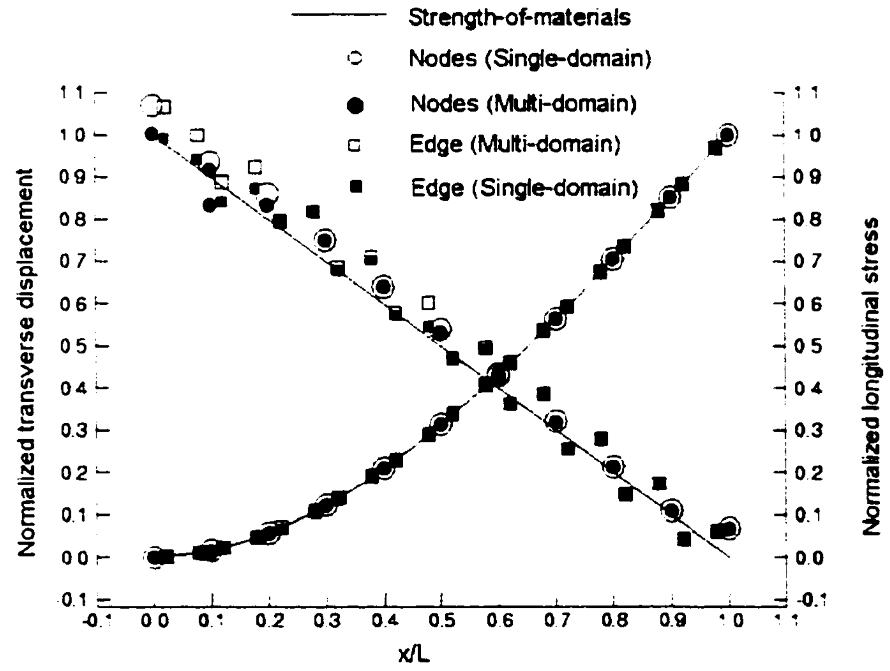
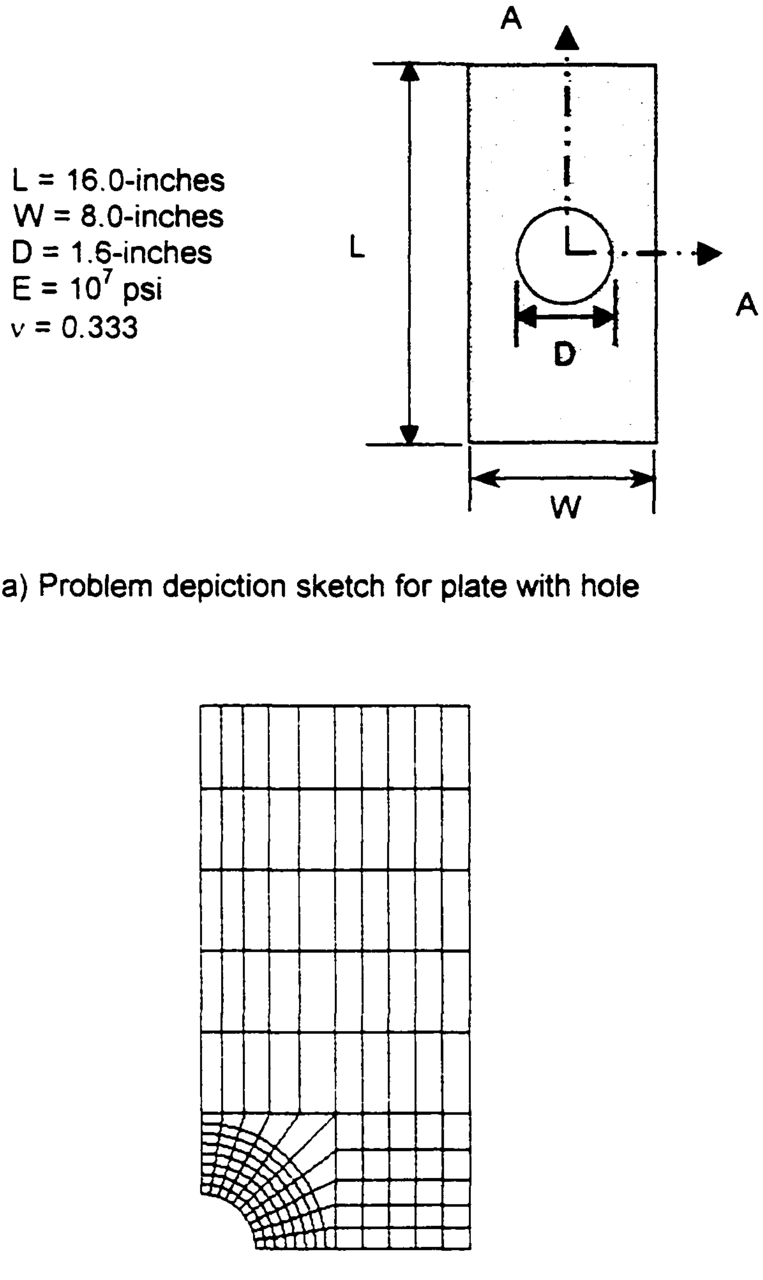


Figure 5.6 Transverse displacement and bending stress for three-dimensional cantilever beam.



(a) Problem depiction sketch for plate with hole

(b) Single-domain reference model

Figure 5.7 Configuration layout and reference finite element model for two-dimensional plate with hole.

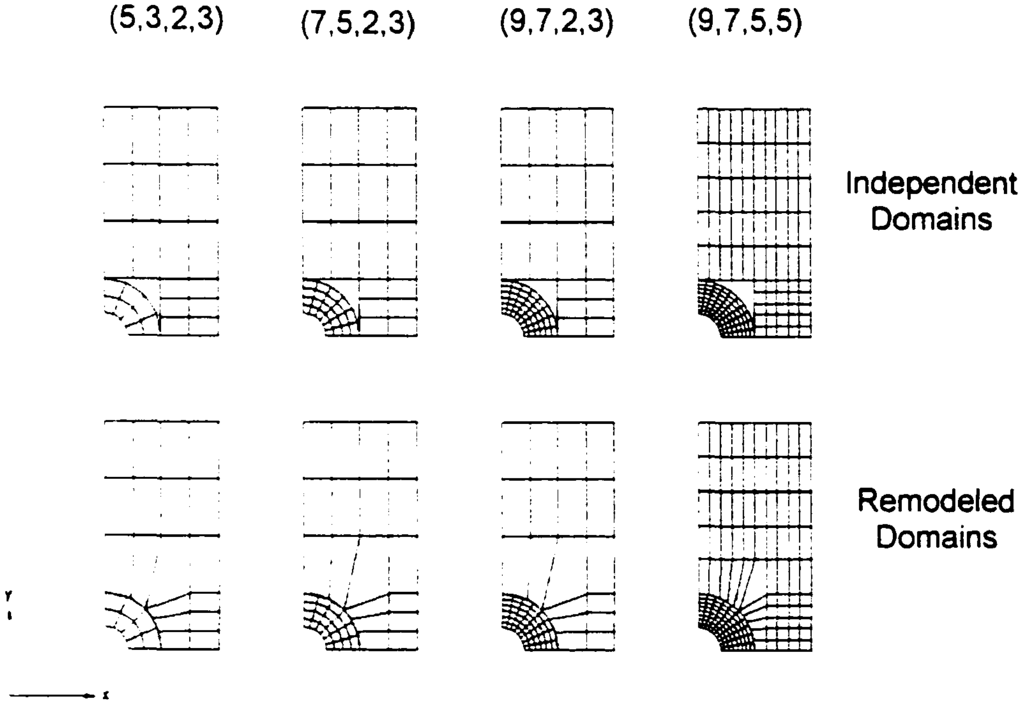


Figure 5.8 Mesh refinement for two-dimensional plate with hole.

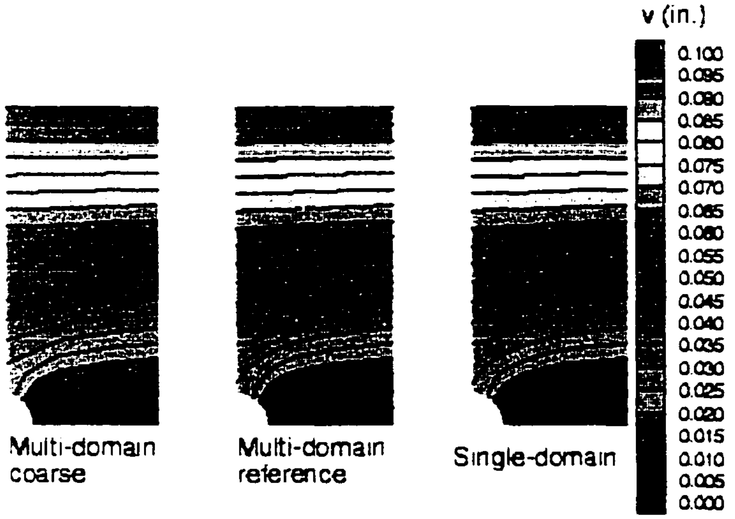


Figure 5.9 Longitudinal displacement contours for plate with hole.

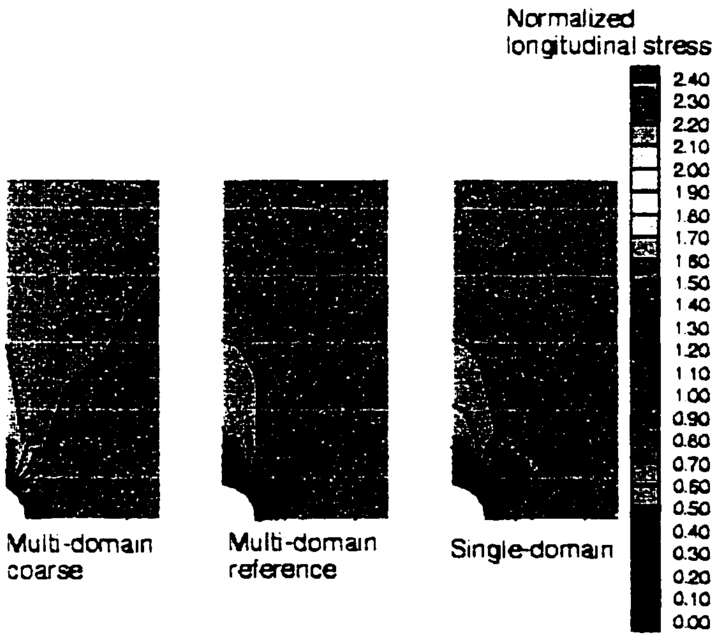


Figure 5.10 Normalized longitudinal stress contours for plate with hole.

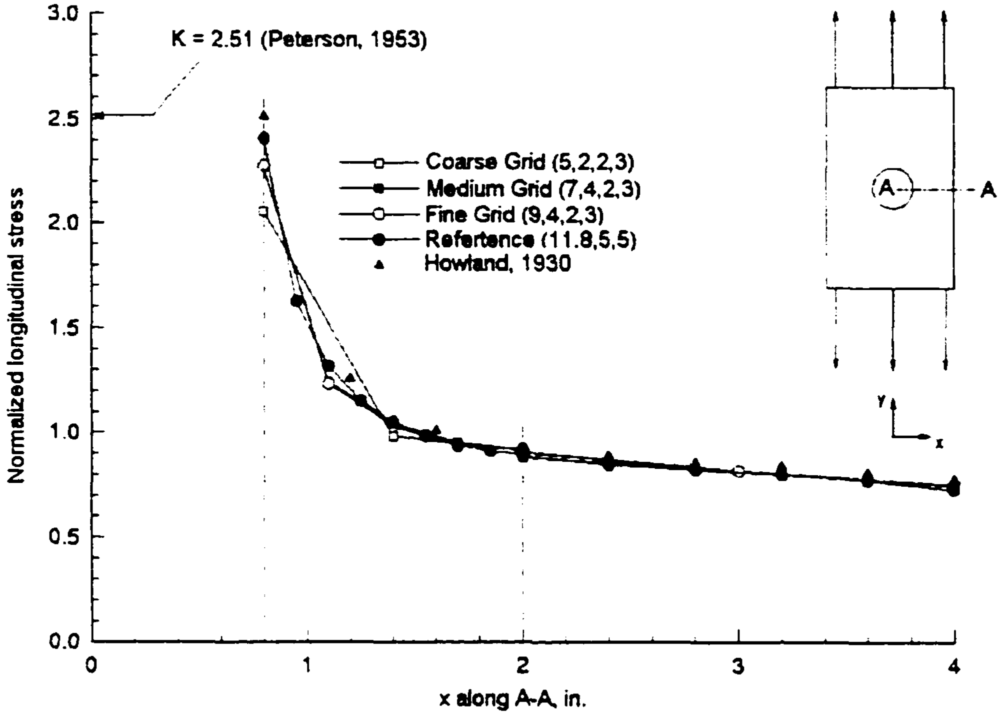


Figure 5.11 Longitudinal stress at mid-length for two-dimensional plate with hole.

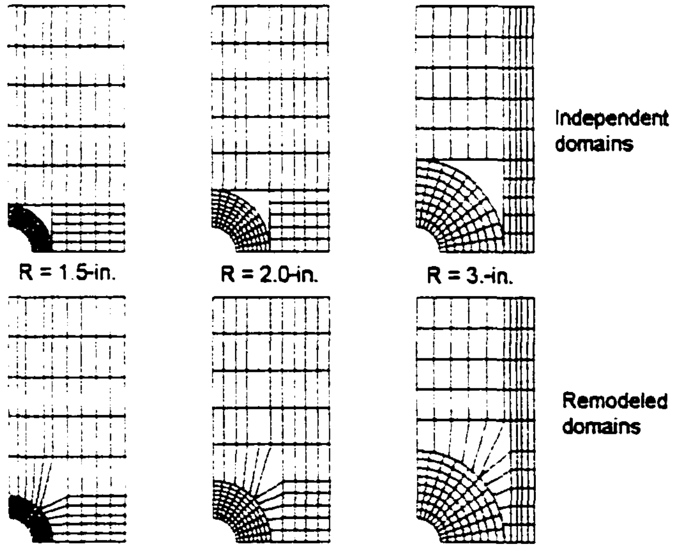


Figure 5.12 Multi-domain grids for different interface locations.

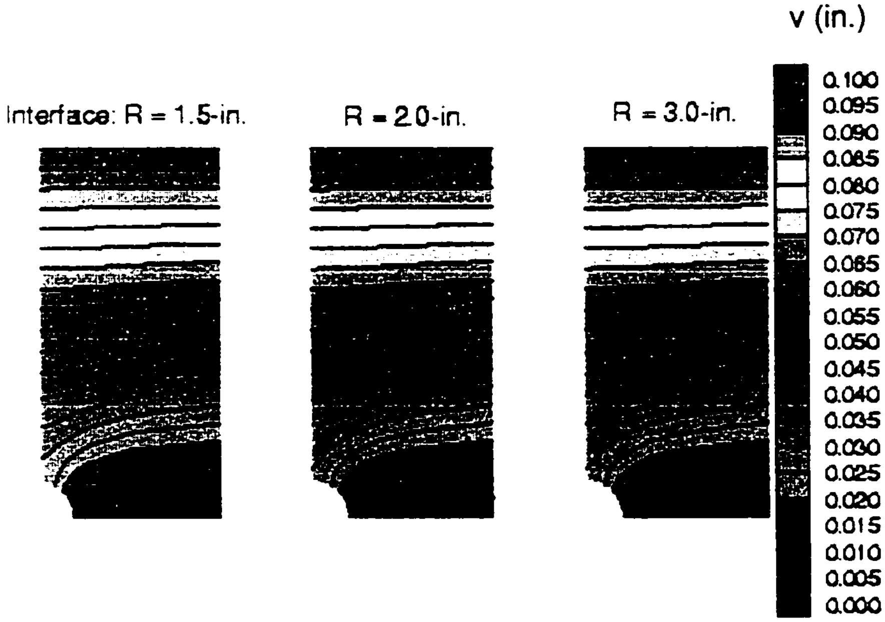


Figure 5.13 Longitudinal displacement contours showing insensitivity to interface location.

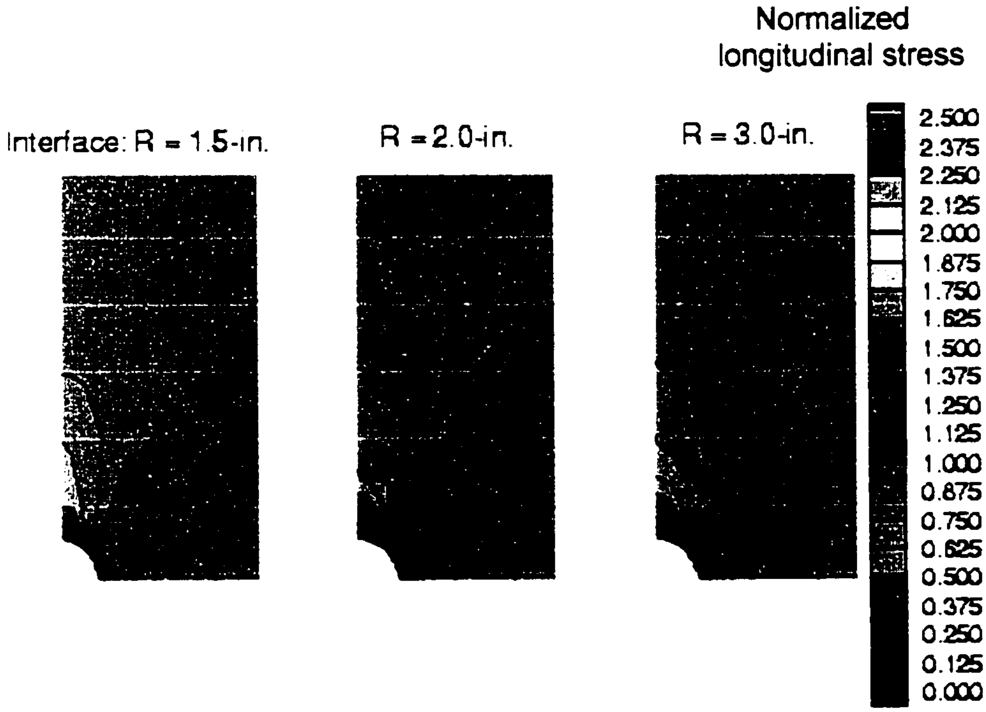


Figure 5.14 Normalized longitudinal stress contours showing insensitivity to interface location.

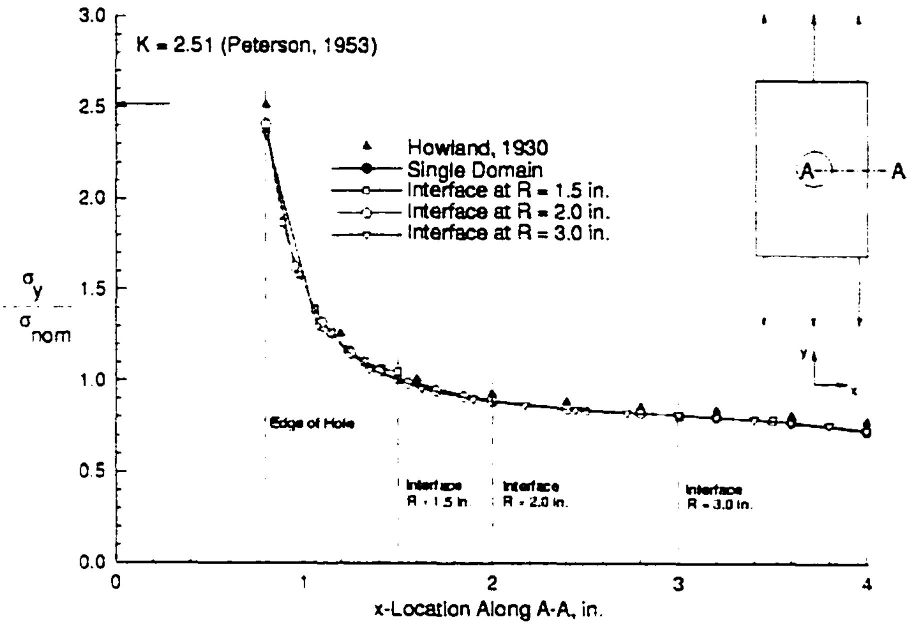
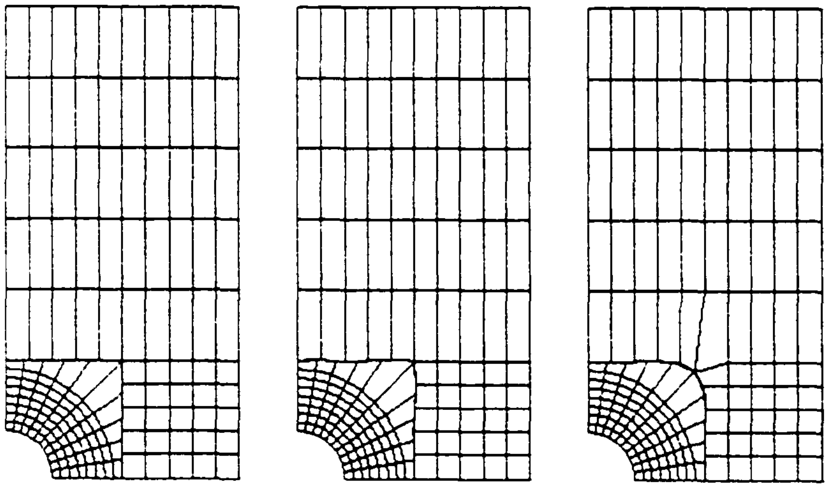


Figure 5.15 Longitudinal stress at mid-length showing insensitivity to interface location.



Independent Domains Interfaced with corner node Interfaced without corner node

Figure 5.16 Multi-domain meshes having L-shaped interfaces in finite-element model for two-dimensional plate with hole.

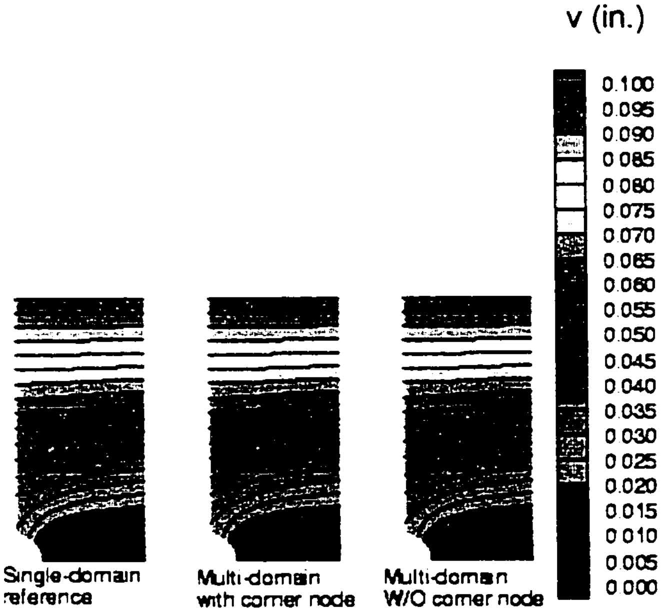


Figure 5.17 Longitudinal displacement contours for two-dimensional plate with hole showing insensitivity to interface shape.

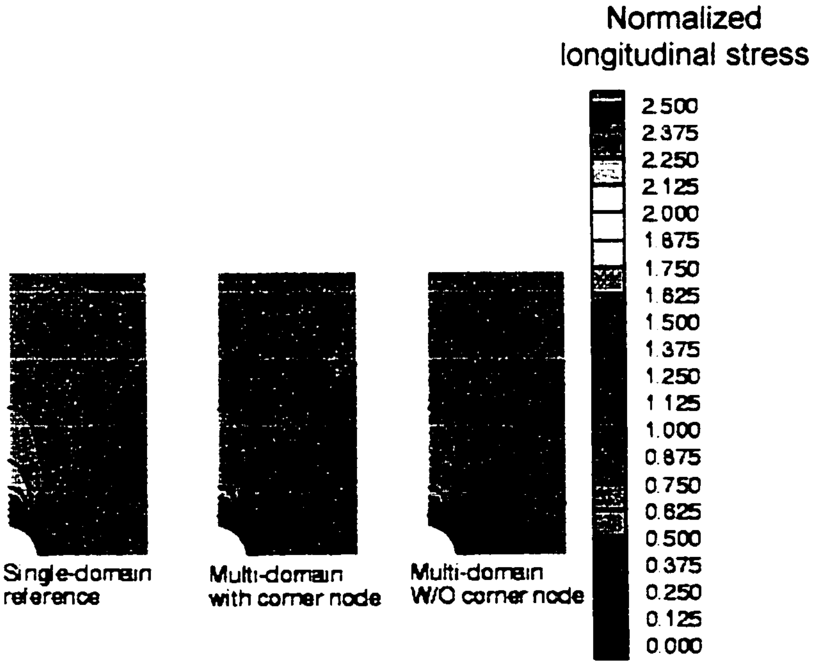


Figure 5.18 Normalized longitudinal stress contours for two-dimensional plate with hole showing negligible sensitivity to interface shape.

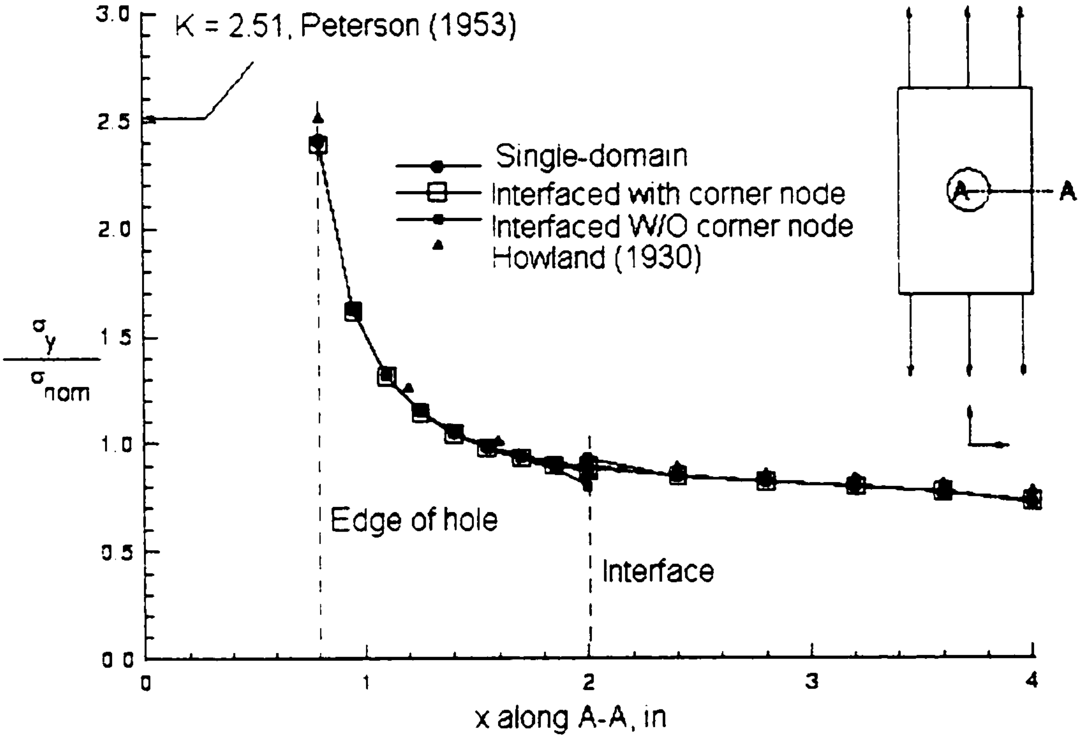


Figure 5.19 Longitudinal stress along y = 0 line for two-dimensional plate with hole showing insensitivity to interface shape.

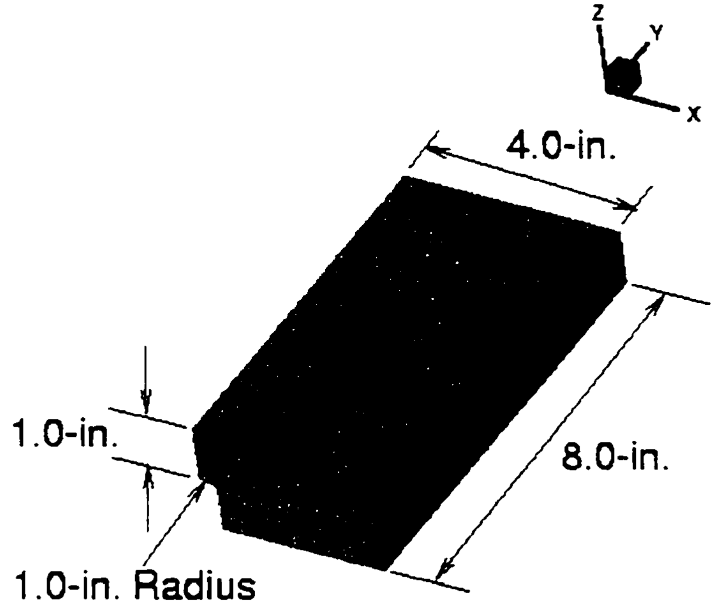


Figure 5.20 Configuration layout for solid plate with circular hole.

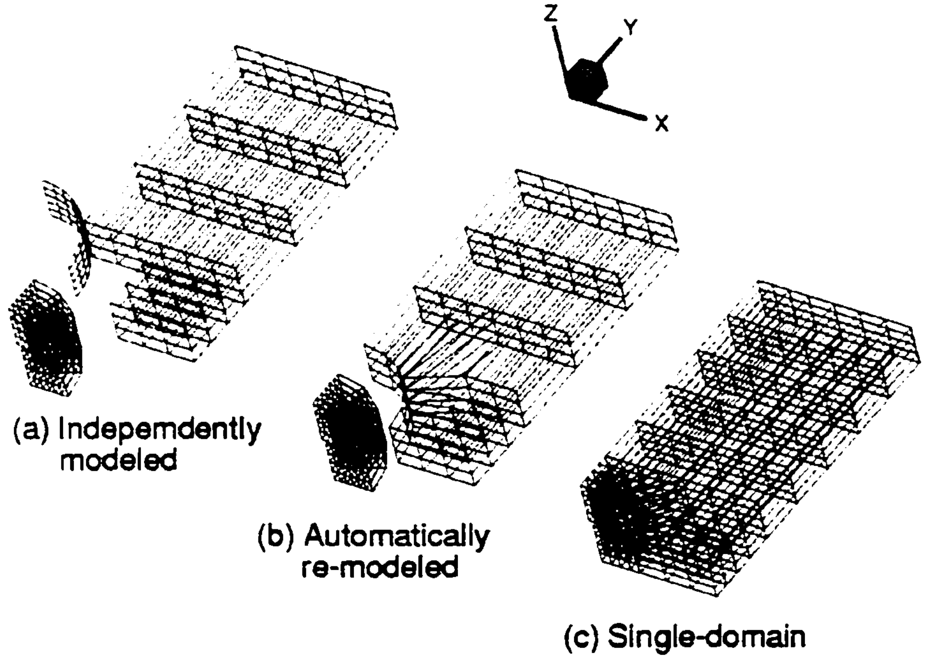


Figure 5.21 Finite element models of solid plate with circular hole.

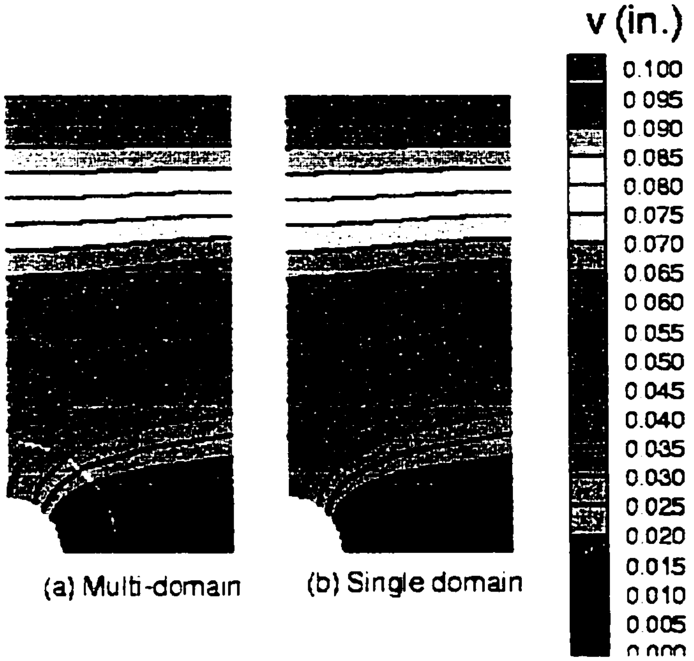


Figure 5.22 Longitudinal displacement contours for solid plate with circular hole showing negligible variation in depth direction.

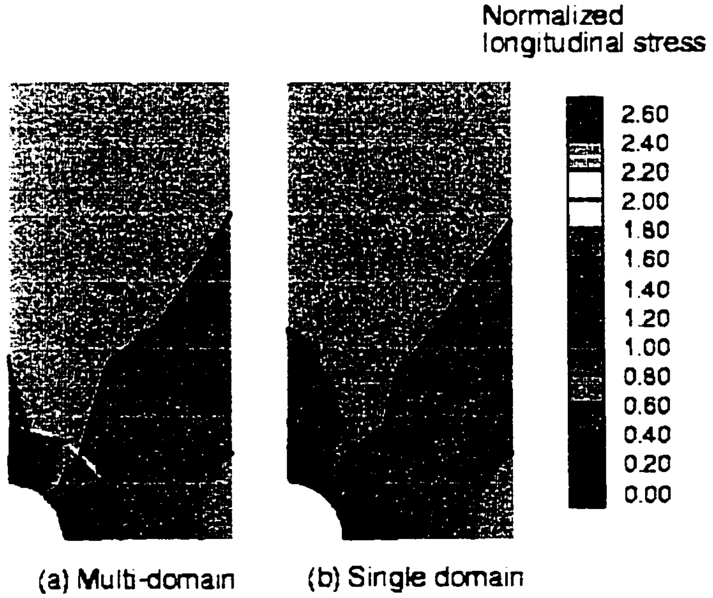


Figure 5.23 Longitudinal stress contours for solid plate with circular hole showing negligible variation in depth direction.

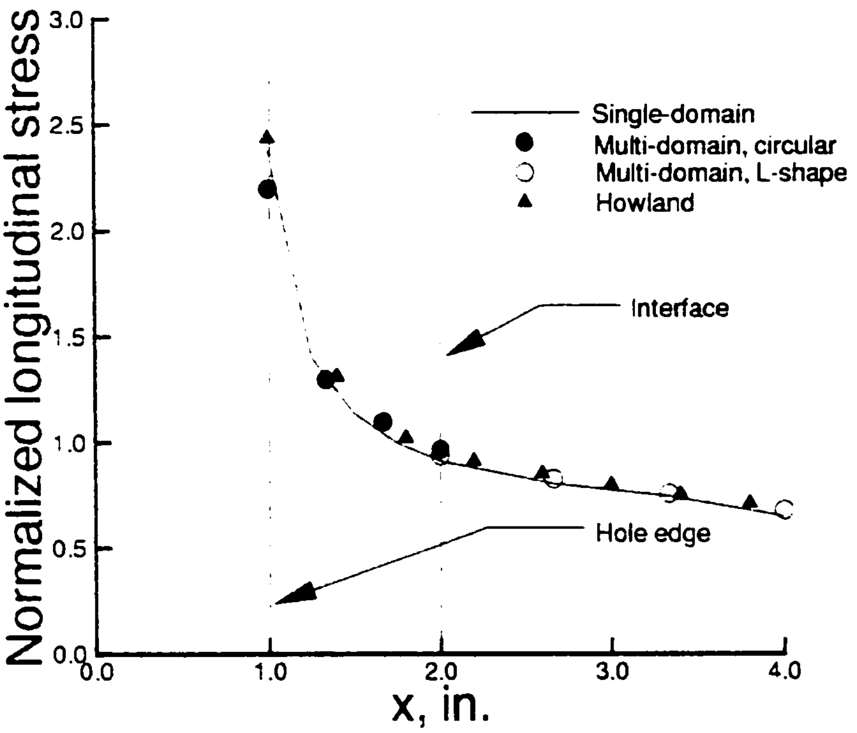


Figure 5.24 Longitudinal stress along y = 0, z = 0, line for solid plate with circular hole.

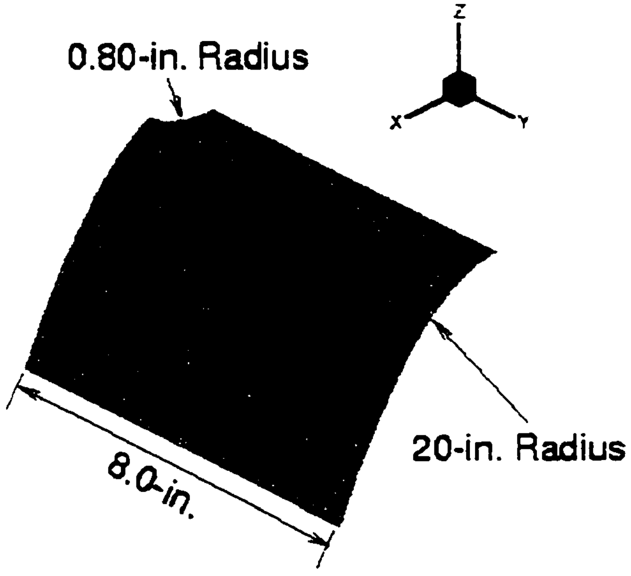


Figure 5.25 Configuration layout for curved membrane with circular hole.

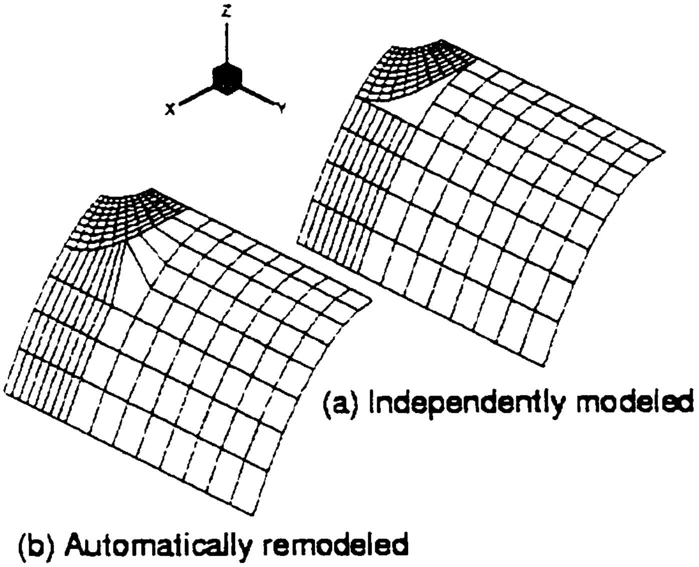
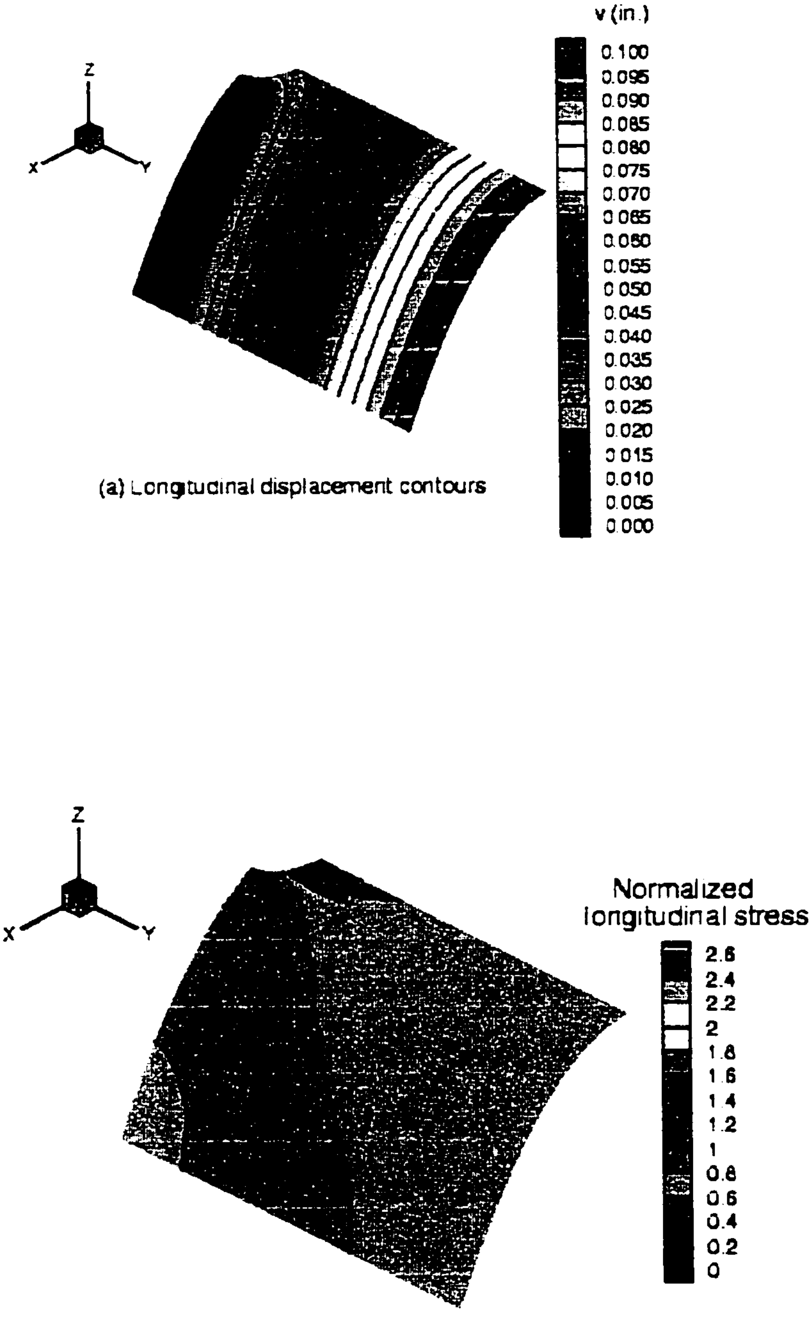


Figure 5.26 Multi-domain finite element model for curved membrane with circular hole.



(a) Longitudinal displacement contours

(b) Longitudinal stress contours

Figure 5.27 Displacement and stress contours for curved membrane with circular hole.

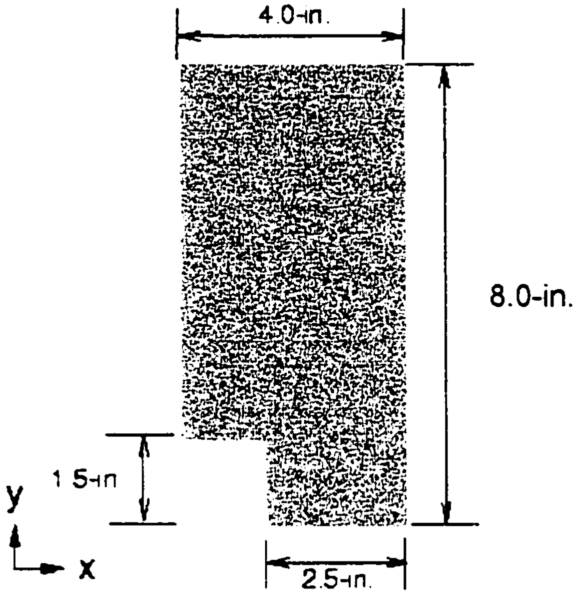


Figure 5.28 Configuration layout for two-dimensional notched plate.

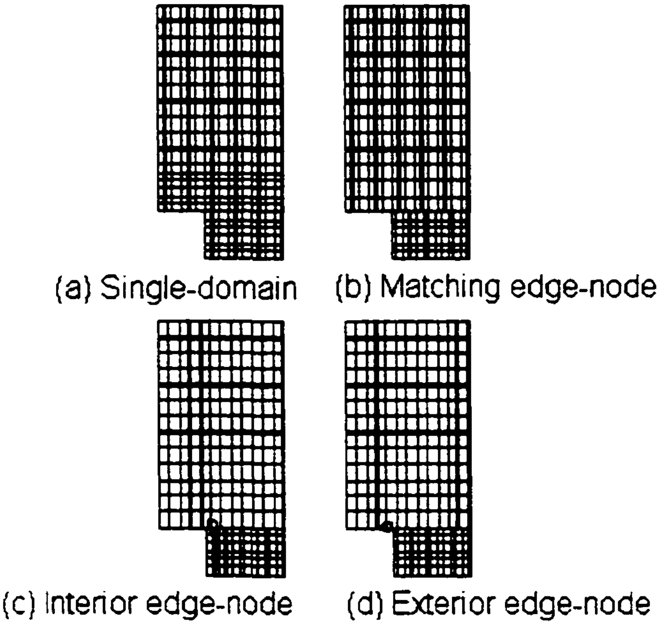


Figure 5.29 Finite element models for two-dimensional notched plate.

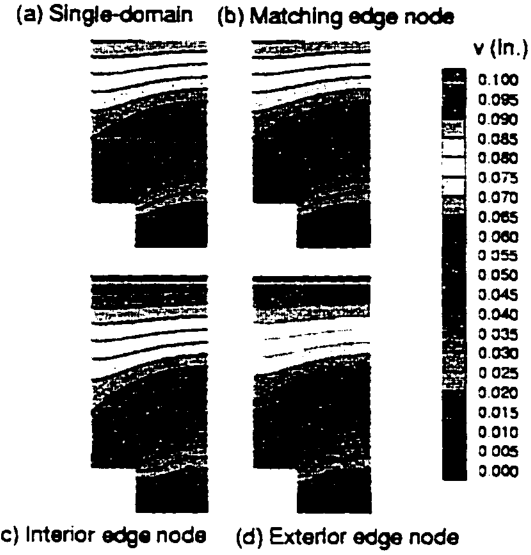


Figure 5.30 Longitudinal displacement contours for two-dimensional notched plate.

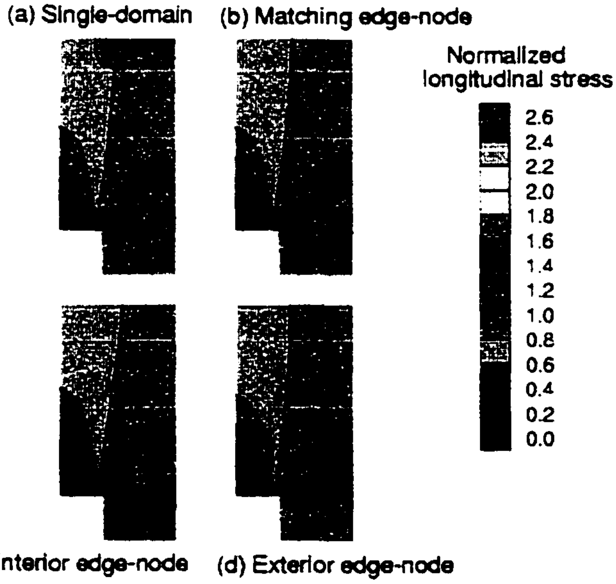


Figure 5.31 Longitudinal stress contours for notched plate.

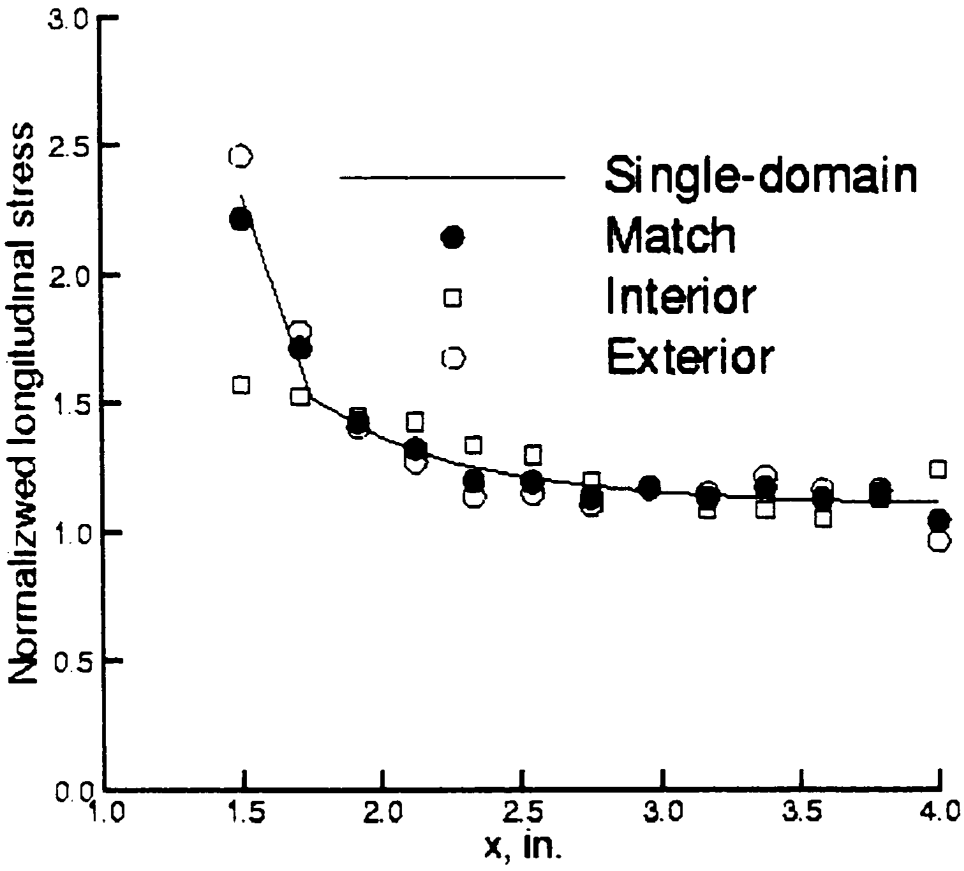


Figure 5.32 Longitudinal stress along notch line for notched plate.

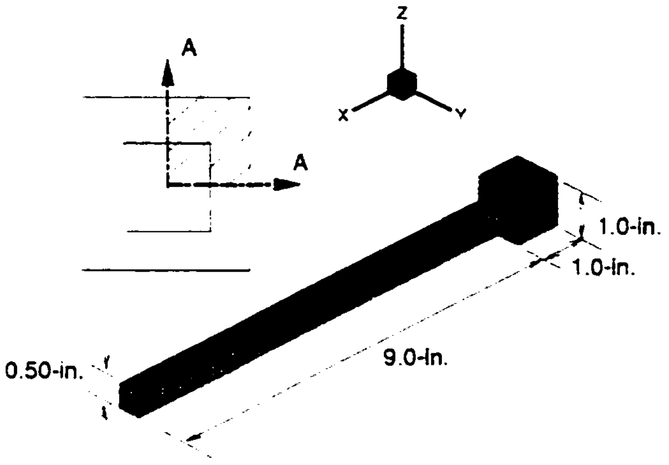


Figure 5.33 Configuration layout for solid bar with abrupt size reduction.

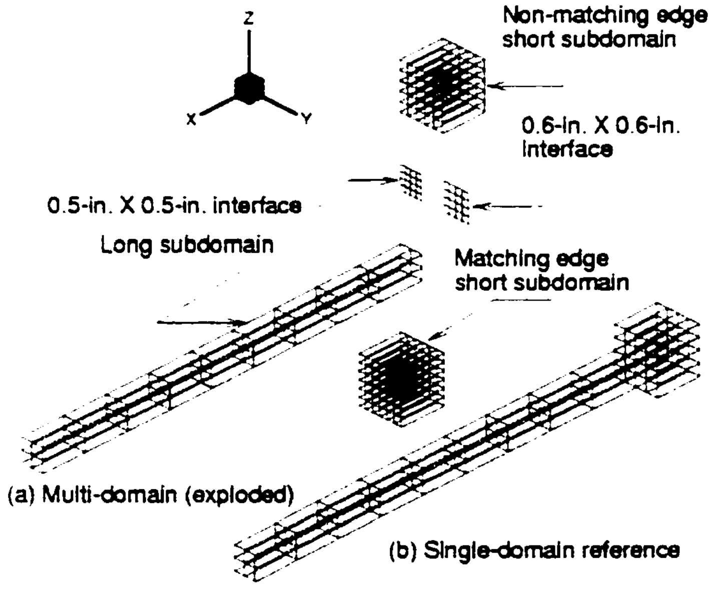
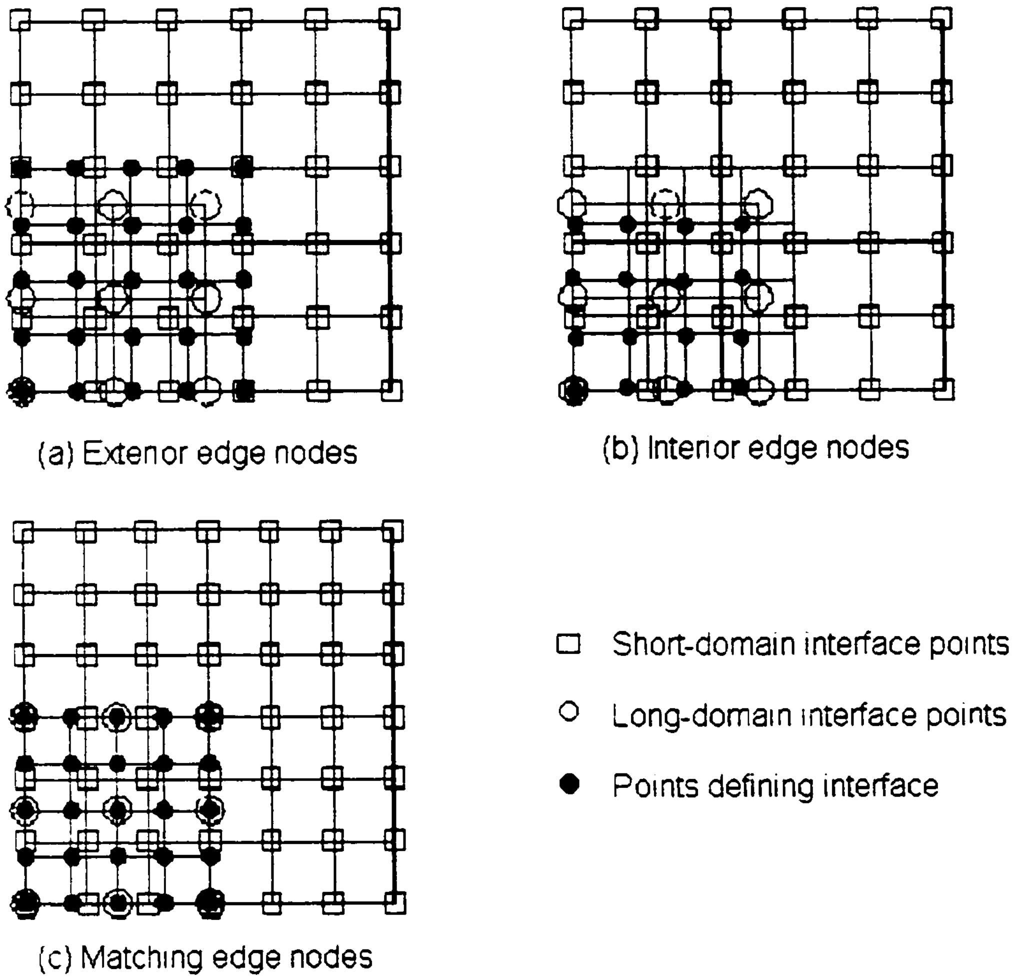

Z
X
Y
Non-matching edge short subdomain
0.6-in. X 0.6-in. Interface
0.5-in. X 0.5-in. interface
Long subdomain
Matching edge short subdomain
(a) Multi-domain (exploded)
(b) Single-domain reference

Figure 5.34 Finite element models and interfaces for solid bar with abrupt size reduction.



(a) Exterior edge nodes

(b) Interior edge nodes

(c) Matching edge nodes

□ Short-domain interface points
○ Long-domain interface points
● Points defining interface

Figure 5.35 Interface variations for multi-domain finite element models of solid bar having abrupt size reduction.

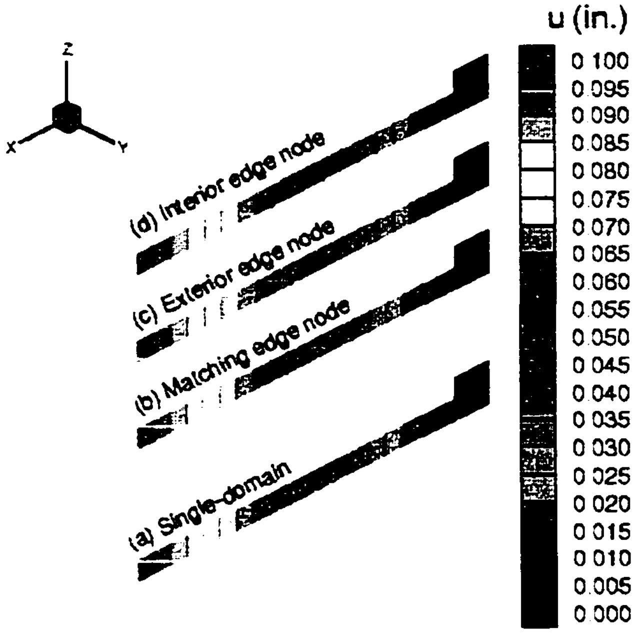


Figure 5.36 Longitudinal displacement contours on center plane of solid bar having abrupt size reduction.

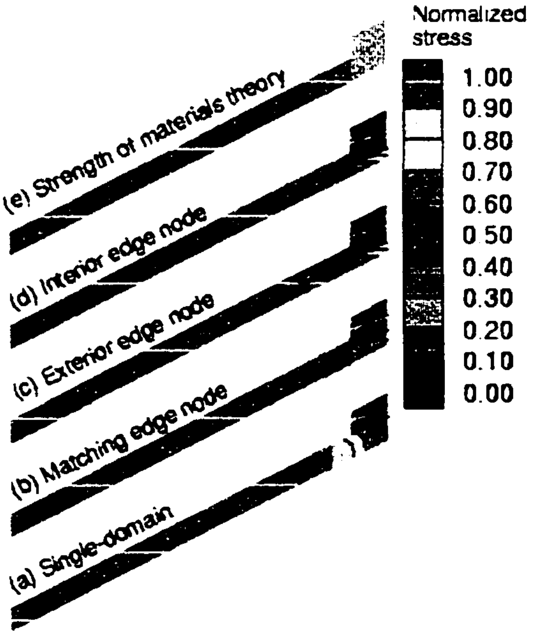


Figure 5.37 Longitudinal stress contours on center plane of solid bar having abrupt size reduction.

CHAPTER VI

CONCLUSIONS AND RECOMMENDATIONS

6.1 Summary

An overview of the background and objectives of the present research together with a historical review of related research was presented in Chapter I. The literature review included a discussion of approaches to interface-type investigations for both Computational Fluid Dynamics (CFD) and Computational Structural Mechanics (CSM) fields of research. The goals of the research were specified in section 1.3.

Chapter II discussed the formulation of previous interface theory based on a hybrid-stress variational principle and included the steps followed in the overall solution process. Chapter III presented a formulation and solution approach for the automatic remodeling of the independently modeled subdomains, a key part of the present research. Chapter III also included representative examples of independently modeled subdomains remodeled automatically using the present methodology. Chapter IV discussed the methodology for evaluating and assembling the generalized global stiffness matrix. Finally, Chapter V presented and interpreted representative finite element cases that demonstrated the methodology.

Cases examined included both two- and three-dimensional patch tests to validate the element implementation. Basic configurations studied were bars and cantilever beams in both two- and three-dimensions. Flat plates with circular holes, both membrane and solid, were analyzed and discussed. The plate with hole examples covered mesh refinement, interface location, and interface shape studies. A panel with out-of-plane curvature having a circular hole was investigated. The example investigations concluded with two- and three-dimensional configurations having discontinuous changes in section size and included coincident and non-coincident interface edge definitions.

6.2 Conclusions

The five research objectives defined in Chapter I, Section 1.3 have been achieved in the course of the investigations and discussed in the present document. In the applications cases presented, the methodology has been demonstrated and the results presented have shown good agreement with previously published results and benchmark finite element computations.

Specifically, the major conclusions of this research are as follows:

1. Methodology for automatically remodeling independently modeled subdomains, having non-coincident regions in the finite element models, has been developed and tested for a variety of cases in both two- and three dimensions. The remodeling methodology has been shown to be accurate and robust for every case considered in the present research. This automatic remodeling capability represents a unique new feature for interface technology. It permits the use of convenient modeling approaches for subregions and eliminates the need for exact boundary modeling along the interface.
2. Algorithms for evaluating displacement and traction constraint integrals have been developed, implemented and tested individually and as part of the combined methodology. The algorithms have been designed to be compatible with the general parametric representation used in the remodeling methodology, and the resulting implementation is both versatile and robust.
3. Linear solution algorithms have been implemented to address the problem of the inherent non-positive-definite character of the global generalized stiffness matrices. The implementation includes user-selectable options for solving the large linear systems based on condition number of the matrix for the system. This feature allows solution, if necessary, of near-singular systems. The trade-off in invoking the optional, sophisticated capability to solve the more intractable systems is additional computational time.
4. Representative classes of applications have been solved that gave good agreement with solutions from the literature and from single-domain reference

solutions. The results obtained and presented in Chapter V demonstrate the robustness, accuracy, and versatility of the overall methodology as implemented.

5. Overall, the work presented herein demonstrates an original and significant research contribution in the area of computational mechanics.

6.3 Recommendations for additional research

Every worthwhile research effort bears more fruit than the original questions answered when additional questions to be answered and new areas ripe for further investigation are identified.

6.3.1 Element type

The four-node quadrilateral elements and the eight-node brick elements used to implement the theory are widely available, straightforward to implement, and give generally good results except for configurations and loading resulting in significant bending. Adding additional, more sophisticated element types to the code capability is an area for additional work.

The cubic spline elements used to define the interfaces provided excellent results for the implementation effort invested. There are situations, however, where cubic splines can result in spurious curvature in the interpolation curves or surfaces. In these cases, a satisfactory possible solution is the use of splines in tension. These spline types have the additional satisfying feature of mimicking elastic behavior; an interesting property for interface interpolation functions that are intended to be used in structural applications to possess .

6.3.2 Stress Recovery

The stress recovery methods implemented for the present research possess several desirable features. One of these being the intuitively natural way that the element shape functions are used to extrapolate the stresses from the sampling points to the nodes. A second desirable feature is in the coding simplicity of arithmetic averaging for node stresses where more that one element shares a node. One limitation of the stress recovery, as implemented and for the meshes considered in the present research, was revealed in the result from the cantilever beam cases. The stresses extrapolated to the end nodes for these cases were inaccurate because insufficient element density was modeled near the beam ends. This also affected the continuity of the stresses across the interfaces.

6.3.3 Global assembly

The most memory-intensive module in the set of codes is the module that assembles and preconditions the generalized global stiffness matrix. One reason for this is due to the approach chosen for the integral evaluations of the M- and G-matrices. These matrices have non-zero contributions only over a limited number of element edges or surfaces, but the current methodology performs integration over all the elements because it was simpler in the bookkeeping. The many zero entries generated by this procedure were later deflated out in the preconditioning phase. This approach is computationally inefficient and requires unnecessary temporary memory. A clear area for improvement is to make this module more efficient by using a more sophisticated and robust evaluation method.

6.3.4 Generality of interface definition

The two-dimensional methodology, based on one-dimensional interfaces, possesses considerable generality largely due to the more limited geometrical possibilities permitted in two-dimensions. The present approach, however, does not allow for interfaces skewed with respect to the element edges, for example the case of a panel intersecting another panel at an arbitrary orientation. This is an important feature that would enlarge the potential application areas.

For the three-dimensional methodology based on two-dimensional interfaces, there is even more scope for worthwhile increased capability due in large part to the richer variety of geometrical possibilities offered in more dimensions. The present implementation is restricted to subdomain surfaces and related interface definitions which have their respective edges parallel. The capability to include arbitrary relative orientations would enhance the usefulness and robustness of the methodology.

An even more interesting area for potential imprcvement was highlighted during the analysis of the bar having discontinuous area change at the interface. It was observed that there was a possible ambiguity of which interface node to include if the interface edges did not coincide. Computed results for these situations revealed a less than ideal attempt by the methodology to somehow negotiate the discontinuity in cross sectional area. Clearly, there is a need for additional research into this; a generalized type of interpolation likely will need to be developed to fully solve this problem.

An even more innovative area of potential research related to the core of the interface concept would be to see if any of the ideas that have been widely employed in the CFD area could be used in CSM applications. In particular, the Chimera idea of employing automated grid modification and overlapped meshes in two and three dimensions is a fascinating possibility.

REFERENCES

1. C. V. Clerk and R. Muller, 'GIFTS-100: Graphics Oriented Interactive Finite Element Time Sharing System,' *Handbook of Finite Elements,* C. A. Brebbia (ed.), CML Publications, England, (1981).

2. O. C. Zienkiewicz, 'The Finite Element Method: From Intuition to Generality,' *Applied Mechanics Reviews*, 249-256 (1970).

3. O. C. Zienkiewicz, 'The Generalized Finite Element Method - State of the Art and Future Directions,' *ASME Joumal of Applied Mechanics*, **50**, 1210-1217 (1983).

4. K. K. Gupta and J. L. Meek, 'A Brief History of the Beginning of the Finite Element Method,' *International Joumal for Numerical Methods in Engineering*, **39**, 3761-3774 (1996).

5. John Robinson, *Early FEM Pioneers*, Robinson and Associates, Bridestowe, U.K. (1985).

6. R. H. Gallagher, 'Thirty Years of Finite Element Analysis - Are There Issues Yet to be Resolved?', *Finite Elements in Analysis and Design*, **6**, 1-8 (1989).

7. A. K. Noor, 'Books and Monographs on Finite Element Technology,' *Finite Elements in Analysis and Design*, **1**, 101-112 (1985).

8. A. K Noor, 'Bibliography of Books and Monographs on Finite Element Technology,' *Applied Mechanics Reviews*, **44**, 307-317 (1991).

9. Howard Eves, *An Introduction to the History of Mathematics*, 3rd ed., Holt, Rinehart and Winston, New York, (1969).

10. Harold C. Martin and Graham F. Carey, *Introduction to Finite Element Analysis: Theory and Application*, McGraw-Hill, New York, (1973).

11. Irving H. Shames and Clive L. Dym, *Energy and Finite Element Methods in Structural Mechanics*, Hemisphere Publishing, New York, (1985).

12. John William Strutt (Baron Rayleigh), *The Theory of Sound, Vol. 1*, Dover Reprint of 1894 ed., Dover Publications, New York, 83-113 (1945).

13. Walter Ritz, 'Uber eine neue Methode zur Losung gewisser Variationsprobleme der mathematischen Physik,' *J. fur die reine und angewandte Mathematik*, **135**, 1-61 (1909).

14. R. Courant, 'Variational Methods for the Solution of Problems of Equilibrium and Vibrations,' *Bull. Am. Math. Soc.*, **49**, 1-23 (1943).

15. Gene Barron, 'History and Development,' *What Every Engineer Should Know About Finite Element Analysis*, John R. Brauer (ed.), Marcel Dekker, Inc., (1993).

16. G. Polya, 'Sur une Interpretation de la Methode des Differences Fines Qui Peut Fournir des Bornes Superieures ou Inferieures,' *C. R. Acad. Sci.*, **235**, 995-999 (1952).

17. J. Hersch, 'Equations Differentielles et Fonctions de Cellues,' *C. R. Acad. Sci.*, **240**, 1602-1604 (1955).

18. H. F. Weinberger, 'Upper and Lower Bounds for Eigenvalues by Finite Difference Methods,' *Comm. Pure Appl. Math.*, **9**, 613-623 (1956).

19. F. Edward McLean, *Supersonic Cruise Technology*, NASA SP-472, (1985).

20. M. J. Turner, R. W. Clough, H. C. Martin, and L. J. Topp, 'Stiffness and Deflection Analysis of Complex Structures,' *J. of the Aeronautical Sciences*, **23**, 805-854 (1956).

21. R. W. Clough, 'The Finite Element Method in Plane Stress Analysis,' *Proc. of 2nd Conference on Electronic Computation*, ASCE, 345-378 (1960).

22. J. Greenstadt, 'On the Reduction of Continuous Problems to Discrete Form,' *IBM J. Res. Dev.*, **3**, 355-363 (1959).

23. R. H. MacNeal and C. W. McCormick, 'The NASTRAN Computer Program for Structural Analysis,' *Computers and Structures*, **1**, 389-412 (1971).

24. O. C. Zienkiewicz and Y. K. Cheung, *The Finite Element Method in Structural and Continuum Mechanics*, McGraw-Hill, (1967).

25. R. D. Cook, Concepts and Applications of Finite Element Analysis, John Wiley, (1974).

26. Klaus-Jurgen Bathe, *Finite Element Procedures*, Prentice-Hall, Englewood Cliffs, NJ, (1996).

27. C. S. Krishnamoorthy, *Finite Element Analysis: Theory and Programming*, 2nd ed., Tata/McGraw-Hill, New Delhi, (1995).

28. Kenneth H. Huebner and Earl A. Thornton, *The Finite Element Method for Engineers*, 2nd. ed., McGraw-Hill, New York, (1993).

29. J. N. Reddy, *An Introduction to the Finite Element Method*, 2nd ed., McGraw-Hill, New York, (1993).

30. H. Kardestuncer and D. H. Norrie (eds.), *Finite Element Handbook*, McGraw-Hill, New York, (1987).

31. Norman F. Knight, Jr., 'Finite Element Techniques for Nonlinear Postbuckling and Collapse of Elastic Structures,' *Structural Dynamical Systems, Computational Techniques and Optimization*, Gordon and Breach, New York, 1-95 (1997).

32. N. F. Knight, Jr., 'Intelligent Design Evolution and Analysis System,' *Virginia Consortium of Engineering and Science Universities: Spring Conference*, (1997).

33. J. B Ransom, Global/Local Stress Analysis of Composite Structures, NASA TM-101640, (1989).

34. J. B. Ransom and N. F. Knight, Jr., 'Global/Local Stress Analysis of Composite Panels,' *Computers and Structures*, **37**, no. 4, 375-395 (1990).

35. N. F. Knight, Jr., Jonathan B. Ransom, O. Hayden Griffin, Jr. and Danniella M. Thompson, 'Global/Local Methods Research using a Common Structural Analysis Framework,' *Finite Elements in Analysis and Design*, **9**, 91-112 (1991).

36. Jonathan B. Ransom, *Global/Local Analysis of Composite Structures*, M.S. Thesis, Department of Mechanical Engineering and Mechanics, Old Dominion University, (1989).

37. T. Krishnamurthy and I. S. Raju, 'Coupling Finite and Boundary Element Methods for Two-Dimensional Potential Problems', *33rd SDM, AIAA Paper No. 92-2240*, (1992).

38. Itio Hirai, 'An Efficient Zooming Method for Finite Element Analysis,' *Int. J. num. Meth. Engng.*, **20**, 1671-1683 (1984).

39. Itio Hirai, Youshiro Uchiyama, Yoji Mizuta, and Walter D. Pilkney, 'An Exact Zooming Method,' *Finite Elements in Analysis and Design*, **1**, 61-69 (1985).

40. J. S. Przemieniecki, 'Matrix Structural Analysis of Substructures,' *AIAA Journal,* **1**, no. 1, 138-147 (1963).

41. I. C. Taig, 'Automated Stress Analysis using Substructures', *Proc. of the Matrix Methods in Struc. Mech. Conf.,* Wright-Patterson AFB, 255-274 (1966).

42. James C. Cavendish and William J. Gordon, 'Substructured Macro Elements Based on Locally Blended Interpolation,' *Int. J. num. Meth. Engng.*, **11**, 1405-1421 (1977).

43. Hibbitt, Karlsson, and Sorensen, ABAQUS / Standard User's Manual, Volume I, Version 5.5, (1995).

44. I. J. Somervaille, 'A Technique for Mesh Grading Applied to Conforming Plate Bending Finite Elements,' *Int. J. num. Meth. Engng.*, **6**, 310-312 (1973).

45. A. K. Gupta, 'A Finite Element for Transition From a Fine to a Coarse Mesh', *Int. J. num. Meth. Engng.*, **12**, 35-45 (1978).

46. J. M. Housner and M. A. Aminpour, 'Multiple Methods Integration for Structural Mechanics Analysis and Design,' *First NASA Advanced Composites Technology Conference,* NASA CP 3104, Part 2, 875-889 (1991).

47. J. M. Housner, M. A. Aminpour, and S. L. McCleary, 'Some Recent Developments in Computational Structural Mechanics,' *Proceedings of the International Conference on Computational Engineering Sciences*, Editors: S. N. Athuri, B. E. Beslcos, R. Jones, and G. Yagawa, August 12-16, Melbourne, Australia, 376-381 (1991).

48. Yvon Maday, Cathy Mavriplis and Anthony Patera, Nonconforming Mortar Element Methods: Application to Spectral Discretizations, NASA CR-181729 (1988).

49. Gary L. Giles and Keith R. Norwood, 'Coupled Equivalent Plate Finite Element Formulation in Multiple-Methods Structural Applications,' *Journal of Aircraft*, **31**, no. 5, 1189-1196 (1994).

50. A. K. Rao, I. S. Raju, and A. V. Krishnamurthy, 'A powerful Hybrid Method in Finite Element Analysis,' *Int. J. num. Meth. Engng.*, **3**, 389-403 (1971).

51. Charbel Farhat and Francois-Xavier Roux, 'A Method of Finite Element Tearing and Interconnecting and Its Parallel Solution Algorithm,' *Int. J. num. Meth. Engng.*, **32**, 1205-1227 (1991).

52. Charbel Farhat and M. Geradin, 'Using a Reduced Number of Lagrange Multipliers for Assembling Parallel Incomplete Field Finite Element Approximations,' *Computer Meth. in Appl. Mech. and Engng.*, **97**, 333-354 (1992).

53. M. A. Aminpour, J. B. Ransom and S. L. McCleary, 'Coupled Analysis of Independently Modeled Finite Element Subdomains,' *AIAA/ASME/ASCE/AHS/ASC 33rd Struc., Struc. Dynamics, and Materials Conf.*, Dallas, AIAA Paper No. 92-2235, (1992).

54. Mohammad A. Aminpour, Jonathan B. Ransom and Susan L. McCleary, 'A Coupled Analysis Method for Structures with Independently Modeled Finite Element Subdomains,' *Int. J. num. Meth. Engng.*, **38**, 3695-3718 (1995).

55. Jonathan B. Ransom, Susan L. McCleary and Mohammad A. Aminpour, 'A New Interface Element for Connecting Independently Modeled Substructures,' *AIAA/ASME/ASCE/AHS/ASC 34th Struc., Struc. Dynamics, and Materials Conf.*, La Jolla, CA, AIAA Paper No. 93-1503 (1993).

56. C. G. Davila, J. B. Ransom and M. A. Aminpour, Cross-Surface Interface Element for Coupling Built-up Structural Subdomains, NASA TM-109125 (1994).

57. Mohammad A. Aminpour and Thiagarajan Krishnamurthy, 'A Two-Dimensional Interface Element for Multi-Domain Analysis of Independently Modeled Three-Dimensional Finite Element Meshes,' *AIAA/ASME/ASCE/AHS/ASC 38th Struc., Struc. Dynamics, and Materials Conf. and Exhibit and AIAA/ASME/AHS Adaptive Struc. Forum*, Kissimmee, FL, AIAA Paper No. 97-1297 (1997).

58. Mohammad A. Aminpour and Thiagarajan Krishnamurthy, and Tushar D. Fadale, 'Coupling of Independently Modeled Three-Dimensional Finite Element Meshes with Arbitrary Shape Interface Boundaries,' *AIAA/ASME/ASCE/AHS/ASC 39th Struc., Struc. Dynamics, and Materials Conf.*, Long Beach, CA, AIAA Paper No. 98-2060 (1998).

59. Mohammad A. Aminpour, Susan L. McCleary and Jonathan B. Ransom, 'A Global/Local Analysis Method for Treating Details in Structural Design,' *Third NASA Advanced Composites Tech. Conf.*, Long Beach, CA, (1992).

60. Mohammad A. Aminpour, Thiagarajan Krishnamurthy, Susan L. McCleary and Majdi A. Baddourah, 'Application of a New Interface Element to the Global/Local Analysis of a Boeing Composite Crown Panel,' *4th NASA/DoD Advanced Composites Tech. Conf.*, Salt Lake City, UT (1993).

61. Jerrold M. Housner, Mohammad A. Aminpour, Carlos G. Davila, John E. Schiermeier, W. Jefferson Stroud, Jonathan B. Ransom and Ronnie E. Gillian, 'An Interface Element for Global/Local and Substructuring Analysis,' *MSC World Users' Conf.*, Los Angles, CA (1995).

62. J. B. Ransom, 'Interface Technology for Geometrically Nonlinear Analysis of Multiple Connected Subdomains,' *AIAA/ASME/ASCE/AHS/ASC 38th Struc., Struc. Dynamics, and Materials Conf. and Exhibit and AIAA/ASME/AHS Adaptive Struc. Forum*, Kissimmee, FL, AIAA Paper No. 97-1298 (1997).

63. Kamran Fouladi-Semnani, *Navier-Stokes Simulations of Flows About Complex Configurations using Domain Decomposition Techniques*, Ph. D. Dissertation, Department of Mechanical Engineering and Mechanics, Old Dominion University, (1990).

64. John P. Steinbrenner, John R. Chawner, and Chris L. Fouts, The GRIDGEN 3D Multiple Block Grid Generation System, V. I and II, WRDC-TR-3022, (July 1990).

65. Abdullah Arabshahi and David L. Whitfield, 'A Multiblock Approach to Solving the Three-Dimensional Unsteady Euler Equations About a Wing-Pylon-Store Configuration,' *AIAA Atmospheric Flight Mechanics Conf.*, Boston, AIAA Paper No. 89-3401 CP, (1989).

66. Brian A. Nishida, Ronald G. Langhi, and Daniel P. Benze, 'A Multiblock/Multigrid Euler Analysis of a Propfan Transport With Wing-Mounted Nacelles Including Slipstream Effects,' *AIAA 29th Aerospace Sciences Meeting*, Reno, AIAA Paper No. 91-0706, (1991).

67. Achi Brandt, 'Multi-Level Adaptive Solutions to Boundary-Value Problems', *Mathematics of Computation*, **31**, 333-390 (1977).

68. J. Z. Zhu and A. W. Craig, 'Finite Element Multigrid Algorithms and Their Application to Engineering Problems', *Proceedings of the NUMETA '85 Conference*, University College of Swansea, 927-932 (1985).

69. T. L. Holst, K. L. Gundy, J. Flores, and N. M. Chaderjian, 'Numerical Solution of Transonic Wing Flows Using an Euler/Navier-Stokes Zonal Approach,' *AIAA 18th Fluid Dynamics, Plasmadynamics and Lasers Conf.*, Cincinnati, AIAA Paper No. 85-1640, (1985).

70. Terry L. Holst, Scott D. Thomas, Unver Kaynak, Karen L. Gundy, Jolen Flores, and Neal M. Chaderjian, *Computational Aspects of Zonal Algorithms for Solving the Compressible Navier-Stokes Equations in Three-Dimensions*, NASA TM-86774, (1985).

71. B. S. Baldwin and H. Lomax, 'Thin Layer Approximation and Algebraic Model for Separated Flows,' AIAA Paper No. 78-257, (June 1978).

72. R. Beam and R. F. Warming, 'An Implicit Finite Difference Algorithm for Hyperbolic Systems in Conservation-Law-Form,' *Journal of Computational Physics*, **22**, 87-110 (1976).

73. E. H. Atta, 'Component-Adaptive Grid Interfacing,' AIAA Paper No. 81-0382, (Jan. 1981).

74. E. H. Atta and J. Vadyak, 'A Grid Interfacing Zonal Algorithm for Three-Dimensional Transonic Flows About Aircraft Configurations,' *AIAA/ASME 3rd Joint Thermophysics, Fluids, Plasma and Heat Transfer Conf.*, AIAA Paper No. 82-1017, (1982).

75. J. L. Steger, F. C. Dougherty, and J. A. Benek, 'A Chimera Grid Scheme,' *ASME Symposium on Advances in Grid Generation*, FED, **5**, Houston, 1-11 (1983).

76. Peter Daves, ed., *The American Heritage Dictionary of the English Language*, Dell, New York, 126 (1974).

77. J. A. Benek, J. L. Steger, and F. C. Dougherty, 'A Flexible Grid Embedding Technique with Application,' AIAA Paper No. 83-1944, (July 1983).

78. J. A. Benek, T. L. Donegan, and N. E. Suhs, 'Extended Chimera Grid Embedding Scheme with Application to Viscous Flows,' AIAA Paper No. 87-1126, (June 1987).

79. J. A. Benek, J. L. Steger, F. C. Dougherty, and P. G. Buning, *Chimera: A Grid-Embedding Technique*, AEDC TR-85-64, Arnold Engineering Development Center Report, (April 1986).

80. O. Baysal, K. Fouladi, and V. P. Lessard, 'Multigrid and Upwind Viscous Flow Solver on Three-Dimensional Overlapped and Embedded Grids,' *AIAA Journal*, **29**, no. 6, 903-910 (1991).

81. Victor Robert Lessard, *Domain Decomposition for Multigrid Finite Volume Flow Solvers*, M. S. Thesis, Department of Mechanical Engineering and Mechanics, Old Dominion University, (1989).

82. James C. Newman, III and O. Baysal, 'Transonic Solutions of a Wing/Pylon/Finned-Store Using Hybrid Domain Decomposition,' *AIAA Atmospheric Flight Mechanics Conf.*, Hilton Head, AIAA Paper No. 92-4571, (1992).

83. O. Baysal, K. Fouladi, R. W. Leung, and J. S. Sheftic, 'Interference Flows Past Cylinder-Fin-Sting-Cavity Assemblies,' *Journal of Aircraft*, **29**, no. 2, 194-202 (1992).

84. Oktay Baysal and Guan-Wei Yen, 'Kinematic Domain Decomposition for Boundary-Motion-Induced Flow Simulations,' *5th SIAM Conf. on Domain Decomposition Methods for Partial Differential Equations, Norfolk*, (1991).

85. Oktay Baysal and Guan-Wei Yen, 'Kinematic Domain Decomposition to Simulate Flows Past Moving Objects,' *29th Atmospheric Sciences Meeting*, Reno, AIAA Paper No. 91-0725, (1991).

86. Mohamed E. Eleshaky and Oktay Baysal, 'Aerodynamic Shape Optimization via Sensitivity Analysis on Decomposed Computational Domains,' *4th AIAA/NASA/OAI/ Multidisciplinary Analysis and optimization Conf.*, Cleveland, AIAA Paper No. 92-4698 CP, (1992).

87. Steven J. Parks, Pieter G. Buning, Joseph L. Steger, and William M. Chan, 'Collar Grids for Intersecting Geometric Components Within the Chimera Overlapped Grid Scheme,' AIAA Paper No. 91-1587 CP, (1991).

88. Robert L. Meakin, 'Moving Body Overset Grid Methods for Complete Aircraft Tiltrotor Simulation,' AIAA Paper No. 93-3350 CP, (1993).

89. O. Baysal, K. P. Singh, and G. W. Yen, 'Dynamic CFD Methods for Prescribed and Aerodynamically-Determined Relative-Moving Multibody Problems,' First AFSOR Conference on Dynamic Motion CFD, 31-34, (Eds. Sakell, Knight), New Brunswick, NJ, (June 1996).

90. G. W. Yen and O. Baysal, 'Computing Unsteady High Speed Flows Past an Oscillating Cylinder Near a Vertical Wall,' *Journal of Spacecraft and Rockets*, **31**, no. 4, 630-635, (1994).

91. G. W. Yen and O. Baysal, 'Effects of Efficiency Techniques on Accuracy of Dynamic Overlapped Grids for Unsteady Flows,' *Journal of Fluids Engineering*, **119**, no. 3, 577-583 (1997).

92. O. Baysal and X-B. Luo, 'Dynamic Unstructured Method for Relative Motion of Multi-Body Configuration at Hypersonic Speeds,' *Journal of Aircraft*, **36**, no. 4, 725-729 (1999).

93. O. C. Zienkiewicz and R. L. Taylor, The Finite Element Method, 4th ed. Vol. 1 – Basic Formulations and Linear Problems, McGraw-Hill, New York, (1989).

94 S. P. Timoshenko and J. N. Goodier, *Theory of Elasticity*, 3rd ed., McGraw-Hill, New York, (1987).

95. Glenn James and Robert C. James (eds.), *Mathematics Dictionary*, D. Van Nostrand, Princeton, (1966).

96. I. D. Faux and M. J. Pratt, *Computational Geometry for Design and Manufacture*, Halsted Press, New York, 163 (1979).

97. A. C. H. Chan, 'The Magic of Gauss Quadrature,' *Engineering Computation*, 8, 189-190 (1991).

98. J. J. Dongarra, C. B. Moler, J. P. Bunch and G. W. Stewart, *LINPACK User's Guide*, SIAM, Philadelphia, (1979).

99. Amtec Engineering, Inc., *Tecplot User's Manual, Version 7*, NASA/LaRC Central Scientific Computing Complex Document G-25, (1996).

100. E. Hinton and J. S. Campbell, 'Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method,' *International Journal for Numerical Methods in Engineering*, **8**, 461-480 (1974).

101. Gilbert Strang and George J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, NJ, 181-192 (1973).

102. John Barlow, 'Optimal Stress Locations in Finite Element Models,' *International Journal for Numerical Methods in Engineering*, **10**, 243-251 (1976).

103. Richard H. MacNeal and Robert L. Harder, 'A Proposed Standard Set of Problems to Test Finite Element Accuracy', *Finite Elements in Analysis and Design,* 1, 2-20 (1985).

104. G. Kirsch, *Fachzeitschrift Verein Deutscher Ingenieur,* **1**, Dusseldorf, (1898).

105. R. C. J. Howland, 'On the Stresses in the Neighborhood of a Circular Hole in a Strip under Tension', *Philosophical Transactions*, R. Soc. London, **229**, 49-84 (1930).

106. R. E. Peterson, *Stress Concentration Design Factors*, John Wiley and Sons, New York, (1953).

107. G. N. Savin, *Stress Concentration around Holes*, New York, (1961).

108. M. F. Spotts, *Design of Machine Elements*, 4th ed., Prentice-Hall, Englewood Cliffs, (1971).

109. Joseph Edward Shigley, Mechanical Engineering Design, 2nd. ed., McGraw-Hill, New York, (1972).

110. Pamela P. Walatla, Jean Clucas, R. Kevin McCabe, and Rick Potter, http://science.nas.nasa.gov/Software/FAST, NASA/AMES Research Center Website, (23 February 2000).

VITA

Ollie J. Rose
602 Scott Street
Goldsboro, NC 27534
orose@exchange.moc.edu

Degrees:

B.A. Physics, East Carolina University, 1969
M.S. Physics, East Carolina University, 1975
M.A. Applied Mathematics, University of Maryland, 1976
Ph.D. Engineering Mechanics, Old Dominion University, 2000

Professional Experience:

Approximately twenty years experience in computational mechanics (CFD and FEM), wind tunnel testing, and supervision of technical professionals. Approximately five years of teaching technical subjects at undergraduate college level. Currently assistant professor of science and mathematics engaged in developing a new applied physics and pre-engineering program at an independent undergraduate college.

Affiliations:

American Society of Mechanical Engineers, American Association of Physics Teachers, North Carolina Academy of Science, PE License (Virginia)

Honors:

Sigma Pi Sigma (national physics honor society), ASEE/NASA Summer Fellow