

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI[®]

The Variable Element Topology Finite Element Method

By

Philip Michael Gullett

B.S. (University of California, Davis) 1991
M.S. (University of California, Davis) 1997

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Civil and Environmental Engineering

in the

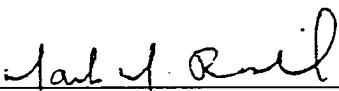
OFFICE OF GRADUATE STUDIES

of the

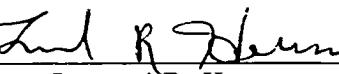
UNIVERSITY of CALIFORNIA

Davis

Approved:



Mark M. Rashid, Chair



Leonard R. Herrmann



John B. Bolander

Committee in Charge

2001

UMI Number: 3026648



UMI Microform 3026648

Copyright 2001 by Bell & Howell Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

ABSTRACT

The Variable Element Topology Finite Element Method

Philip M. Gullett

This dissertation introduces a new innovation in the finite element method, called the Variable-Element-Topology Finite Element Method (VETFEM), in the context of linear elastic solid mechanics. The VETFEM provides a means of generating element based, nodal polynomial shape functions without imposing the geometric and topological restrictions that are characteristic of the conventional finite element method. The VETFEM shape functions are numerically determined in the physical element domain using a series of constraints and minimizations that generate functions suitable for use as an approximation basis in the Galerkin discretization of the weak form. Shape functions generated by this approach share many of the desirable qualities of conventional finite element shape functions such as compact support and the Kronecker delta property. In addition, VETFEM elements are completely compatible with conventional isoparametric elements, and a computational mesh may consist of both types of elements in any convenient pattern to form a finite element model. Because VETFEM shape functions may contain slight discontinuities on element edges, the VETFEM is generally nonconforming. However, convergence of the method is achieved by the completeness of the approximation space and the satisfaction of the Céa conditions. The VETFEM was incorporated into a finite element code, and the numerical examples presented here demonstrate that VETFEM solutions compare favorably to those of conventional finite elements.

Acknowledgements

A guy like me doesn't finish something like this without owing some folks a great deal of thanks. First among them are wife Wendy and our children. Especially Wendy, because she knew what being married to a graduate student meant: five long years of low income, student housing, etc. Through it all, she supported my efforts and encouraged me to get this done, and I'm happily indebted to her.

Next in line is my advisor and mentor, Professor Mark Rashid. I'm not sure how I managed to dupe such an intelligent guy into taking me on as a student. But from the beginning he exhibited a confidence in me that made me think I could actually accomplish this work. And it was with his guidance, support, and constant encouragement that I've actually completed the goal. I am very fortunate to have been one of his students.

I also am grateful to Professors Bolander and Herrmann for serving on my qualifying examination and thesis committees and to Ritwick Roy and Stefano Berton, officemates with whom I passed many stimulating and enjoyable hours.

Finally, I thank God who, in His grace, orchestrated this whole thing for me. Soli Deo Gloria.

*To my wife
and my children*

Contents

Abstract	ii
Dedication	iv
1 Introduction	1
1.1 The finite element method	2
1.2 Meshless methods	4
1.3 The VETFEM	8
1.4 Outline	9
2 Overview of Galerkin methods	10
2.1 Governing equations	10
2.1.1 Strong form	11
2.1.2 Weak (variational) form	13
2.1.3 Weak form properties	15
2.2 Galerkin method	15
2.2.1 Convergence	17
2.2.2 The finite element method	20
2.2.3 Meshless methods	25
2.2.4 The VETFEM	30

3 VETFEM	32
3.1 VETFEM shape functions	32
3.2 Polynomial shape functions	34
3.3 Convergence	36
3.3.1 Completeness	38
3.3.2 Consistency	38
3.4 Shape function construction	43
3.4.1 Kronecker delta constraints	44
3.4.2 Consistency constraints	45
3.4.3 Constraint application	47
3.4.4 Compatibility optimization	48
3.4.5 Smoothness optimization	50
3.5 Numerical integration rule	52
3.5.1 Consistency constraints and numerical integration	56
3.6 Outline of the algorithm	56
3.7 Example shape functions	58
3.7.1 Example 1: square element with 4 nodes	59
3.7.2 Example 2: square element with 7 nodes	63
3.7.3 Example 3: concave element	70
3.8 Closing remarks	77
4 VETFEM convergence	78
4.1 Nonconforming methods	78
4.2 Convergence analysis of the VETFEM	82
4.2.1 The patch test	89
4.2.2 The generalized patch test	96

5 Numerical Results	99
5.1 Cantilever beam	99
5.2 Slotted plate	102
5.3 Plate with a circular hole	105
5.3.1 Mesh data	115
6 Conclusion	123
A Reference Information	126
A.1 Notation	126
A.2 Symbols	128
A.3 Definitions	129
A.4 Theorems and identities	129
B VETFEM Software	131
B.1 General Gaussian elimination	131

List of Figures

1.1	Image of a section from human femur	4
1.2	Image of a section from human vertebral body	5
1.3	Trabecular bone section with finite element discretization	6
1.4	A VETFEM element in \mathbb{R}^3	9
2.1	An elastic body with prescribed displacements and tractions in \mathbb{R}^3	11
2.2	An example computational model for the FEM method	20
2.3	An example of an inappropriate FEM subdomain configuration	21
2.4	A quadrilateral reference element	23
2.5	Non-coplanar nodes	25
2.6	An example computational model for meshless methods	26
3.1	A typical VETFEM element in \mathbb{R}^2	35
3.2	Pascal's triangle	36
3.3	Square element with four nodes	59
3.4	Shape functions for square element with four nodes	61
3.5	Shape function properties for square element with four nodes	62
3.6	Square element with seven nodes	63
3.7	Shape functions 1 – 4 for square element with seven nodes	67
3.8	Shape functions 4 – 7 for square element with seven nodes	68
3.9	Shape function properties for square element with seven nodes	69

3.10 Concave element with five nodes	70
3.11 Shape function ϕ_3 is nearly zero outside the shaded region	71
3.12 Shape functions 1 – 2 for concave element	73
3.13 Shape function 3 for concave element	74
3.14 Shape functions 4 – 5 for concave element	75
3.15 Local basis function properties	76
4.1 Schematic of the mesh parameter h	85
4.2 A patch test including essential and natural boundary conditions	90
4.3 Pairwise cancellation of shape function contributions to the bilinear form .	94
5.1 Analysis of a cantilever beam	100
5.2 Cantilever beam deformed meshes	101
5.3 Analysis of a slotted plate	102
5.4 Meshes used in slotted plate analysis	104
5.5 Analysis of a strip plate with circular hole	105
5.6 Solution spaces of self-similar and non-self-similar function sequences . . .	106
5.7 Plot of Displacement vs. Element size	110
5.8 Log Plot of Displacement vs. Element size	111
5.9 Plot of Displacement vs. Element size	112
5.10 Log Plot of First Principal vs. Element size	113
5.11 Log Plot of Displacement vs. Element size	114
5.12 Example meshes from sequence 1	116
5.13 Example meshes from sequence 2	118
5.14 Example meshes from sequence 3	120
5.15 Example meshes from sequence 4	122

List of Tables

3.1	Nodal coordinates for square element with four nodes	60
3.2	Shape function coefficient values for a square element with four nodes	60
3.3	Nodal coordinates for square element with seven nodes	64
3.4	VETFEM stages and the number of bound and free coefficients	64
3.5	Shape function coefficient values for a square element with seven nodes	66
3.6	Nodal coordinates locations for concave element	72
3.7	Shape function coefficient values for a concave element	72
4.1	Example nodal values of displacement from a patch test	92
5.1	Comparison of VETFEM and FEM displacement solution for slotted plate .	103
5.2	Page references for mesh data	107
5.3	Mesh sequence 1 data	115
5.4	Mesh sequence 2 data	117
5.5	Mesh sequence 3 data	119
5.6	Mesh sequence 4 data	121
A.1	Definitions of symbols and data sets	128

Chapter 1

Introduction

This dissertation describes the development and implementation of a new method for generating approximate solutions to second order elliptic boundary value problems of solid mechanics. The technique, here called the Variable-Element-Topology Finite Element Method (VETFEM), is similar to the conventional finite element method (FEM) in that it relies on a mesh structure to construct approximating spaces, but it differs from the FEM by including a numerical procedure that generates polynomial shape functions for elements of arbitrary polygon geometry. The benefit of this approach is that it significantly simplifies mesh generation which is currently one of the single biggest hurdles in finite element analysis of complex processes today.

This chapter provides an introduction to the prominent numerical technologies for generating approximate solutions to problems of solid mechanics including the finite element method and the more recently developed meshless procedures. First, Section 1.1 reviews the finite element method, which is currently the most powerful and common procedure for generating solutions. Next, Section 1.2 briefly discusses some of the newer, so-called meshless approaches. Section 1.3 introduces the VETFEM concept, and the remainder of this dissertation is outlined in Section 1.4.

1.1 The finite element method

The finite element method (FEM) was developed by engineers in the mid-1950s [74] and since that time has become the primary means of generating approximate solutions to boundary value problems in solid mechanics. In the opening of the *Handbook of Numerical Analysis* [21], Tinsley Oden writes,

Finite elements; perhaps no other family of approximation methods has had a greater impact on the theory and practice of numerical methods during the twentieth century

The FEM was developed through physical insight and reasoning related to developing approximate solutions to problems of stress analysis. After its initial variational formulation, commonly identified with a 1960 publication by Clough [22], the method expanded rapidly over the succeeding decade to include mixed methods, hybrid methods and Hermite approximations [34, 61, 60]. Its appeal is multifaceted. The procedure itself is highly intuitive, and represents material properties, complex geometric domains and boundary conditions with precision. In addition, the method is mathematically rigorous and computationally convenient. These are among the reasons that now, forty years after its inception, research and application of the finite element method continues to expand vigorously.

Finite element solutions to boundary value problems rely on a division of the problem domain into a set or mesh of subdomains called elements. The element shapes are representable in an alternative coordinate system as simple shapes, typically n -simplices (triangular or tetrahedral) or n -rectangles (quadrilateral or hexahedral). The elements are then used to construct local polynomial basis functions from which a global approximate solution may be constructed. This element-based approach provides the FEM with several important features. Because elements may have essentially arbitrary shapes, complex domain geometries may be precisely followed. In addition, the compact support provided by the locally defined basis functions yields a near orthogonal basis for approximations. This leads to discrete equations that are well suited for powerful direct (frontal [39], skyline, and

SuperLU [25]) and iterative (conjugate gradient [46, 71], multigrid [33, 76, 3]) sparse matrix solvers.

One of the practical difficulties with the finite element framework is construction of a suitable mesh. Mesh generation involves defining a set of nodes and then forming elements that describe a domain accurately, subject to various size and shape criteria. While this process seems quite simple (it is not difficult to imagine an object, such as a coffee mug, subdivided into elements), creation of algorithms to fully automate the process has proved difficult. In fact, Blacker has dubbed automated hexahedral element meshing the “*Holy Grail*” of meshing research [10]. Significant progress has been made in automatic hexahedral element meshing technology, and several strategies have been devised; however, there is, at this time, no all-hex method that will robustly mesh an arbitrary volume [58].

The difficulty of mesh generation has made application of the FEM inefficient for problems that are characterized by extremely complex geometry, large mesh distortions, and/or evolving topologies. Problems in the latter categories typically involve dynamic fracture, penetration, or very large deformations and find vital application in the defense, aerospace, automobile, and nuclear industries. Examples include simulation of metal drawing and forming processes, shape optimization, failure simulations, and advanced materials development. In the case of large deformations, the severe mesh distortions can lead to a breakdown in the numerical accuracy of the approximation. For problems of changing topology, the domain must be remeshed incrementally in order for the discrete model to accurately reflect these changes. Because significant intervention by the analyst is generally required, mesh generation becomes the costliest, most error-prone, and most time consuming part of the analysis. The costs associated with mesh generation often makes simulation more expensive than trial and error testing of actual components.

In areas such as biomechanics, for example, extreme geometric complexity (see Figures 1.1 and 1.2) necessitates fully automatic mesh generation. Current meshing techniques include direct translation of the voxels (volumetric pixels) from a three-dimensional digital

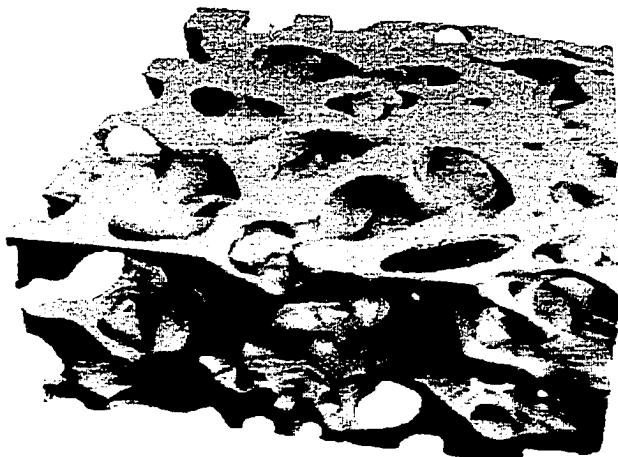


Figure 1.1: 4 x 4 x 1.6 mm section of trabecular bone from a human femur imaged at 20 micron resolution (Reprinted from [1], with permission)

image into hexedral elements (see Figure 1.3). The resulting lack of conformity of the finite element mesh to the smooth boundaries induces large local stress errors in the discrete approximation [14, 31].

1.2 Meshless methods

The difficulty of application of the FEM to some of the aforementioned classes of problems is the primary reason for the recent research efforts focused on techniques that do not rely on a conventional mesh. These so-called meshless methods construct approximations with reference to nodes only, although many of these methods require definition of a background mesh to support numerical quadrature. The fact that these methods do not require explicit connectivity between nodes, or definition of element subdomains, is seen as a significant advantage for problems involving moving discontinuities and large distortions. Although there are many meshless or mesh-free approaches, for example the generalized finite difference techniques [47], only those that utilize the Galerkin framework are considered in this dissertation. For more comprehensive surveys refer to Belytshko et al. [6] and Duarte [27]. Many of these methods may be grouped loosely into three categories: kernel methods, mov-

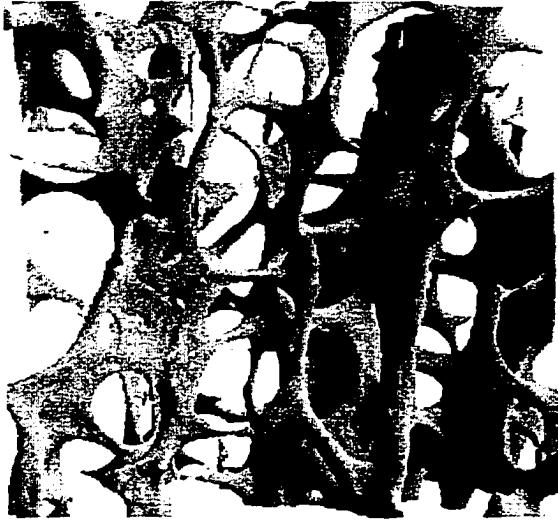


Figure 1.2: $4 \times 4 \times 1.8$ mm section from a human vertebral body (Reprinted from [1], with permission)

ing least-squares methods, and partition-of-unity methods. These groupings relate more to the historical development of the methods than to fundamental differences, but will be retained here. All meshless methods construct approximations from a set of discrete nodal parameters and nodal weight or window functions that provide small supports.

The earliest method is the Smoothed Particle Hydrodynamics (SPH), which was developed by Lucy [48] and Gingold and Monaghan [32], in the context of astrophysics. It was recognized as a kernel method by Monaghan [53], and applied to structural mechanics problems by Libersky [45]. The SPH is based on kernel interpolation theory and is typically implemented as a collocation method, i.e. the discrete equations arise by forcing the residual to vanish at discrete locations, which in this method are called particles. In this approach, the SPH does not involve an independent distribution of integration points. Instead, the velocity field information and the material-state information are collocated at the nodes. This method initially suffered from computational instabilities [50] and lacked linear consistency (see Section 2.2.1). These problems were particularly significant for solving problems of solid mechanics. However, improvements in stability and consistency have been achieved

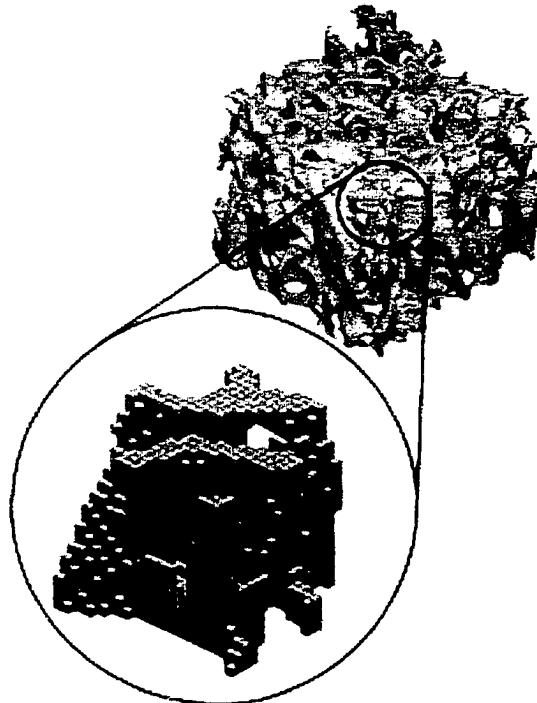


Figure 1.3: A rendering of a 5.1 mm cube of trabecular bone at $20\mu\text{m}$ resolution, and high-resolution finite element model with 60μ elements (Reprinted from [55], with permission from Elsevier Science)

[63, 40, 70, 29] , and the method has found particular applications in problems such as impact and fragmentation [15, 45, 62]. Although the SPH is most commonly employed in collocation methods, it has been implemented in a Galerkin discretization [70].

The Reproducing Kernel Particle Method (RPKM) is related to the SPH in that it is based on kernel interpolation theory. This method, however, adds a correction term to the kernel approximation that enables the method to reproduce polynomial solutions exactly, and significantly improves solution accuracy [18]. In contrast to the SPH, the RPKM is typically implemented in the Galerkin framework, and has been applied to problems including large deformation of rubber [17], contact, [19], and metal forming [16].

Moving least-squares (MLS) techniques are based on interpolation procedures that were originally developed for surface and curve fitting. The first application to problems of solid

mechanics was in the diffuse element method of Nayroles [54]. This approach was modified by Belytschko et. al. [7] in the Element-Free Galerkin (EFG) method. The EFG improved the accuracy of the MLS technique by adding terms to the interpolant derivatives that were omitted by Nayroles, and by employing Lagrange multipliers to enforce essential boundary conditions. In the MLS methods, approximation spaces emerge from a weighted least-squares minimization procedure. The MLS methods are consistent and stable methods for applications in solid mechanics, but are more computationally expensive than SPH. Belytschko [6] has also shown that, in most cases, the EFG method is equivalent to the RKPM.

Each of the aforementioned methods were developed from ideas connected with interpolation theory. These theories are used to provide a mechanism by which approximation bases can be generated from a distribution of particles without reference to a structured domain subdivision. Although the interpolation theories employed are different, the basis function generated in this way are clearly related. That is, the RKPM framework can be used to recover the SPH and EFG basis functions, and thus can be viewed as the more general approach that is independent of the interpolation theory. Meshless approaches were further generalized through the works of Duarte and Oden with the HP Clouds Method [28] and Babuska and Melenk with the Partition of Unity Finite Element Method [51]. These methods provide two different, but general, approximate displacement representations within which EFG and RKPM are instances of partition-of-unity approximations.

In the research context, meshless methods have been successfully applied to analysis of shells, large deformations, contact and turbulence. However, some of the inherent drawbacks in meshless methods have prevented them from effectively competing with the FEM in industrial applications. In particular, meshless approximating functions in general do not possess the Kronecker delta property, and consequently, the approximation does not interpolate nodal values of the solution, i.e., $u^h(\mathbf{x}_i) \neq u_i$, where $u^h(\mathbf{x}_i)$ and u_i are the approximate solution and nodal parameter at node i . This renders representation and influence of the

boundary vague and complicates imposition of essential boundary conditions. Strategies for imposition of essential boundary conditions include collocation, Lagrange multipliers, and coupling with finite elements. In contrast, because finite elements possess the Kroncker delta property, essential boundary conditions are enforced by setting nodal values equal to known values of the displacement; a second drawback is the lack of explicit connectivity among the nodes often leads to a large bandwidth in the global system of equations. Finally, the computations associated with construction of approximates are considerably greater than for the FEM due to required nodal searches, solution of a system of equations at every node, and more extensive numerical quadrature.

1.3 The VETFEM

This dissertation proposes a new computational procedure called the Variable-Element-Topology Finite Element Method. The VETFEM relies on a mesh structure of nodes and elements in order to construct an approximation space. The approximation space is spanned by basis functions that are formed from the union of element-based polynomial shape functions and that possess the Kronecker delta property and provide compact support. The VETFEM differs from conventional finite element methods in that shape functions are not predefined for specific parent element topologies. Instead, the method utilizes a systematic, constrained minimization procedure to generate polynomial interpolants which are defined completely in terms of the physical element geometry. This procedure is general and applies to any n -polygon region. In the VETFEM setting then, elements may have any number of nodes and facets. Figure 1.4 provides an example of a VETFEM element in three-dimensions. This enhancement provides great flexibility in element topology because it imposes virtually no restrictions on specification of nodal connectivities, while retaining the powerful features of the finite element method. VETFEM basis functions may exhibit slight discontinuities on element edges making it a nonconforming method. Nonconforming basis functions are commonly used in FEM applications, but in comparison to conforming

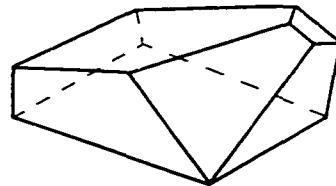


Figure 1.4: A VETFEM element in \mathbb{R}^3

methods, their development requires that more careful attention be paid to convergence. The VETFEM imposes a set of constraints on the shape functions that ensures generation of a convergent sequence of approximations.

1.4 Outline

Chapter 2 develops the strong and weak forms of the boundary value problem describing displacement based linear elasto-statics from the principals of conservation of momentum, reviews the Galerkin strategy for generating approximate solutions. It also examines the FEM, EFG and VETFEM methods as techniques for constructing approximation spaces. Chapter 3 provides a detailed introduction to the Variable-Element-Topology Finite Element Method. Chapter 4 discusses the convergence of this nonconforming method. Chapter 5 presents quantitative and qualitative results. Conclusions and future work are stated in Chapter 6.

Chapter 2

Overview of Galerkin methods

This chapter reviews a few solution strategies for the Galerkin discretization of the equations of linear elasticity. The strong and weak forms of the boundary value problem (BVP) are developed, followed by a presentation of the Galerkin discretization of the weak form. From this point, the finite element method, the Element-Free Galerkin method, and the Variable-Element-Topology Finite Element Method are introduced as strategies for generating suitable Galerkin approximation spaces. Emphasis is placed on how each method constructs basis functions, and the how these choices impact the functionality and utility of the method. Notations adopted here are used throughout this dissertation and are summarized in Appendix A.

2.1 Governing equations

In this section, the BVP that establishes the relationship between applied loads and the resulting deformations in a solid elastic body is developed in its weak and strong forms. The partial differential equations (PDEs) emerge from the governing principal of the balance of linear momentum, using the kinematic assumption of small strains, together with an elastic constitutive model. The boundary conditions are then added to complete the specification of the BVP.

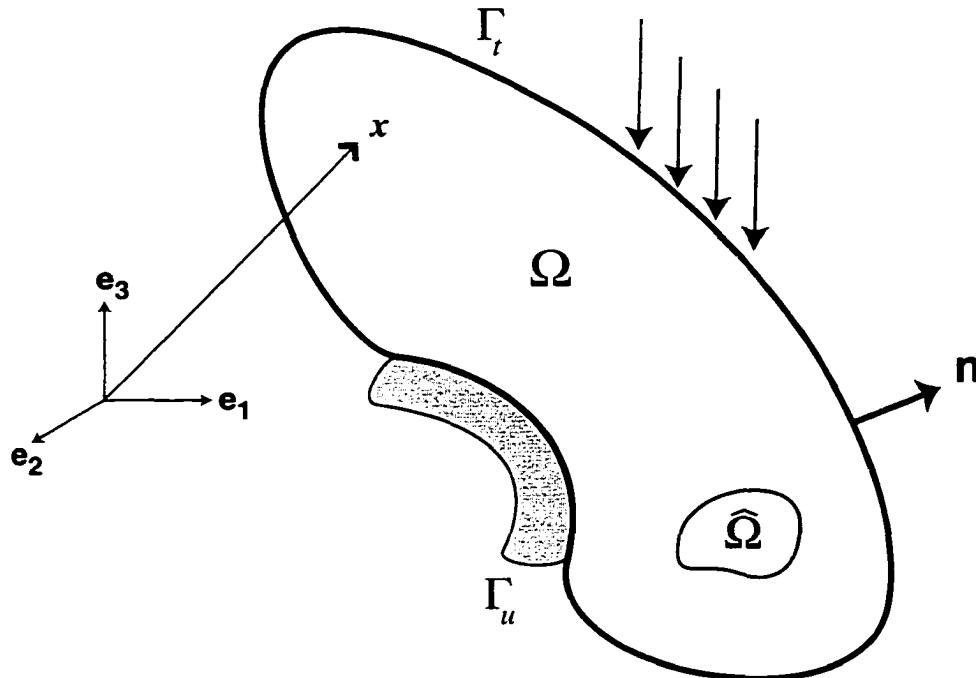


Figure 2.1: An elastic body with prescribed displacements and tractions in \mathbb{R}^3

2.1.1 Strong form

Consider the elastic body Ω shown in Figure 2.1. The positions of points in the body are denoted by the position vector x , and time is denoted by t . The boundary of Ω is denoted $\partial\Omega$ and is made up of sets $\partial\Omega_u$ and $\partial\Omega_t$ such that $\partial\Omega = \partial\Omega_u \cup \partial\Omega_t$. The body is subjected to prescribed displacements \bar{u} on $\partial\Omega_u$ and tractions \bar{t} on $\partial\Omega_t$. Note that these boundary conditions are considered for simplicity.

We seek to find the displacement $\mathbf{u}(x, t)$ of points in the body due to the external loading and constraints. The governing principle for this problem is the balance of linear momentum, which requires that momentum be conserved for every subregion of the body. Applying this balance to an arbitrary subregion $\hat{\Omega}$, that is bounded by the surface $\partial\hat{\Omega}$ yields,

$$\int_{\partial\hat{\Omega}} \hat{\mathbf{t}} \, ds + \int_{\hat{\Omega}} \mathbf{b} \, dA = \int_{\hat{\Omega}} \rho \ddot{\mathbf{u}} \, dA. \quad (2.1)$$

In equation (2.1), \mathbf{b} is the body force density, and $\hat{\mathbf{t}}$ is the traction per unit area acting on $\partial\hat{\Omega}$. Introducing the Cauchy stress tensor $\boldsymbol{\sigma} = \hat{\mathbf{t}}\hat{\mathbf{n}}$ (symmetric by virtue of angular momentum conservation), where $\hat{\mathbf{n}}$ is the normal to the differential area $d\hat{\Omega}$, the momentum balance becomes

$$\int_{\partial\hat{\Omega}} \boldsymbol{\sigma}\hat{\mathbf{n}} \, ds + \int_{\hat{\Omega}} \mathbf{b} \, dA = \int_{\hat{\Omega}} \rho\ddot{\mathbf{u}} \, dA. \quad (2.2)$$

We now re-express the force balance by using the divergence theorem to convert the boundary integral into a domain integral and collecting the terms:

$$\int_{\hat{\Omega}} (\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} - \rho\ddot{\mathbf{u}}) \, dA = \mathbf{0}, \quad (2.3)$$

where $\nabla \cdot$ is the divergence operator. Because the reference volume is arbitrary, and assuming that the integrand is sufficiently smooth, we conclude that it must vanish everywhere in the domain. This leads to the following statement of equilibrium

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \rho\ddot{\mathbf{u}} \quad (2.4a)$$

$$\sigma_{ij,j} + b_i = \rho\ddot{u}_i \quad (2.4b)$$

Under the assumption of small strains and displacements, the kinematic relations are

$$\boldsymbol{\epsilon} = \boldsymbol{\epsilon}(\mathbf{u}) = \nabla_s \mathbf{u} \quad (2.5)$$

where ∇_s is the symmetric gradient operator. The constitutive relation for linear elasticity is given by generalized Hooke's law:

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\epsilon} \quad (2.6a)$$

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl} \quad (2.6b)$$

where \mathbf{C} , the material moduli tensor, is a linear operator. For the special case of isotropic material properties, equation (2.6) reduces to

$$\boldsymbol{\sigma} = \lambda \text{tr} \boldsymbol{\epsilon} \, \mathbf{I} + 2\mu \boldsymbol{\epsilon} \quad (2.7a)$$

$$\sigma_{ij} = \lambda \epsilon_{kk} \delta_{ij} + 2\mu \epsilon_{ij} \quad (2.7b)$$

where λ and μ are the Lamé constants. By substituting the kinematic and constitutive relations into equation (2.4), we obtain Navier's equations which express the equilibrium in terms of displacements:

$$\rho \ddot{\mathbf{u}} - \nabla \cdot (\mathbf{C} : \boldsymbol{\epsilon}) = \mathbf{b} \quad (2.8)$$

or

$$\rho \ddot{\mathbf{u}} - (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) - \mu \nabla^2 \mathbf{u} = \mathbf{b}. \quad (2.9)$$

In equation (2.9), ∇^2 is the Laplacian. The BVP is completed by specification of the PDE along with initial and boundary conditions:

$$\rho \ddot{\mathbf{u}} - (\lambda + \mu) \nabla (\nabla \cdot \mathbf{u}) - \mu \nabla^2 \mathbf{u} = \mathbf{b} \quad (2.10a)$$

$$\boldsymbol{\sigma} \cdot \mathbf{n} = \bar{\mathbf{t}} \text{ on } \partial\Omega_t \quad (2.10b)$$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \partial\Omega_u \quad (2.10c)$$

The displacement solution, $\mathbf{u}(\mathbf{x})$, is the twice differentiable field satisfying both the PDE and boundary conditions of equations (2.10). We note that proving the existence of solutions to equation (2.10) is not trivial even for the case of homogeneous Dirichlet boundary conditions because equation (2.9) contains combinations of partial derivatives of the displacement. A complete proof ultimately relies on Korn's inequalities [35]. For the case of homogenous Neumann boundary conditions in which $\boldsymbol{\sigma}\mathbf{n} = \mathbf{0}$ on $\partial\Omega$, the nullspace (or kernel) of (2.10) is nontrivial and includes the entire space of rigid body displacements and rotations. For a detailed examination see [21, page 35f].

2.1.2 Weak (variational) form

Galerkin based approximate solutions are constructed to satisfy the weak or variational form of the BVP (2.10). In this section we develop the variational boundary value problem (VBVP) associated with equations (2.10). The variational form is weak, in the sense that it demands less smoothness of the displacement field than the strong form does. The weak form is developed by first specifying the space of test functions that satisfy the homogeneous

form of the essential boundary conditions,

$$\mathcal{V} = \{\mathbf{v} \in H^1(\Omega) : \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_u\}. \quad (2.11)$$

Here H^1 is a standard Sobolev space composed of functions with square-integrable first derivatives on Ω (see Appendix A.1 for a more precise definition). Now, confining attention to quasi-static deformations, we multiply equation (2.8) by any function $\mathbf{v} \in \mathcal{V}$ and integrate over the domain:

$$-\int_{\Omega} (\nabla \cdot \boldsymbol{\sigma}) \cdot \mathbf{v} \, dA = \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dA. \quad (2.12)$$

This form of equilibrium still involves two derivatives on the displacement field. We may shift one of the derivatives from the stress term to the test function by applying Green's theorem (A.2). This yields

$$\int_{\Omega} \boldsymbol{\sigma} : \nabla \mathbf{v} \, dA - \int_{\partial\Omega} (\boldsymbol{\sigma} \mathbf{n} \cdot \mathbf{v}) \, ds = \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dA, \quad (2.13)$$

$$\int_{\Omega} \boldsymbol{\sigma} : \nabla \mathbf{v} \, dA = \int_{\partial\Omega_t} \bar{\mathbf{t}} \cdot \mathbf{v} \, ds + \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dA. \quad (2.14)$$

Defining the space of admissible displacement fields for the variational form by

$$\mathcal{S} = \{\mathbf{u} \in H^1(\Omega) : \mathbf{u} = \bar{\mathbf{u}}, \mathbf{x} \in \partial\Omega_u\}, \quad (2.15)$$

we now state the weak formulation of (2.10): Find $\mathbf{u} \in \mathcal{S}$ such that

$$a(\mathbf{u}, \mathbf{v}) = f(\mathbf{v}) \text{ for all } \mathbf{v} \in \mathcal{V} \quad (2.16)$$

where $a(\cdot, \cdot) : \mathcal{V} \times \mathcal{S} \rightarrow \mathbb{R}$ is the bilinear form

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \boldsymbol{\sigma} : \nabla \mathbf{v} \, dA \quad (2.17)$$

and $f(\cdot) : \mathcal{V} \rightarrow \mathbb{R}$ is the linear form

$$f(\mathbf{v}) = \int_{\partial\Omega_t} \bar{\mathbf{t}} \cdot \mathbf{v} \, ds + \int_{\Omega} \mathbf{b} \cdot \mathbf{v} \, dA. \quad (2.18)$$

Solutions satisfying equation (2.16) are only required to be $H^1(\Omega)$ in the weak form, while in the strong form solutions must be from $H^2(\Omega)$. This fact makes the variational form more convenient for constructing approximate solutions, because it embraces a larger space of functions.

2.1.3 Weak form properties

Because subsequent discussions regarding the convergence of approximate methods rely on properties of the weak form, it is important to briefly discuss the uniqueness of solutions for the VBVP. For simplicity, consider the case of homogenous Dirichlet boundary conditions in which the solution and test spaces are equivalent. Uniqueness of problems of the form (2.16) is established through the Lax-Milgram lemma (Appendix A.3) which states that a unique solution exists, provided that the data f is continuous and that bilinear form $a(\cdot, \cdot)$ is continuous and V -elliptic [21, page 29]. An operator is said to be continuous on the space V if

$$|a(u, v)| \leq c_1 \|u\|_V \|v\|_V \quad \forall u, v \in V, \quad (2.19)$$

where $|\cdot|$ is the absolute value, and $\|\cdot\|_V$ is a norm on the space V . The bilinear form is V -elliptic (or coercive or positive definite) if

$$a(u, v) \geq c_2 \|v\|_V^2 \quad \forall v \in V. \quad (2.20)$$

For the variational elasticity problem, continuity may be demonstrated in a straightforward manner, for example see [65, pages 319–320]. Ellipticity, on the other hand is difficult to prove. As in the strong form, the equations depend on certain combinations of the derivatives of the displacement, and the proof relies on Korn’s inequalities. For more details see [68, page 73], [21, page 44f], and [35].

2.2 Galerkin method

The Galerkin method constructs approximations to the weak form, equation (2.16) by defining finite dimensional subspaces of the solution space, and solving the VBVP on these subspaces. In this section, we examine the Galerkin method in its general form, which is sometimes called the Petrov-Galerkin method. We begin by defining the finite dimensional spaces

$$\mathcal{S}_h \subset \mathcal{S}.$$

The space \mathcal{S}_h is spanned by N linearly independent functions $\{\phi_1, \phi_2, \dots, \phi_N\}$ that satisfy the homogeneous form of the essential boundary conditions, i.e. $\phi_i \in \mathcal{V}$, plus a function \bar{u}_h that satisfies the particular form of the essential BCs. Here, h is a parameter that indicates how close \mathcal{S}_h is to \mathcal{S} , and as $h \rightarrow 0$, $\mathcal{S}_h \rightarrow \mathcal{S}$. In addition, we define the spaces

$$\mathcal{V}_h \subset \mathcal{V}$$

spanned by N linearly independent functions $\{\psi_1, \psi_2, \dots, \psi_N\}$. By defining these finite dimensional spaces along with their basis functions, we now express the VBVP, equation (2.16), over these finite dimensional spaces: Find $\mathbf{u}_h \in \mathcal{S}_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) = f(\mathbf{v}_h) \text{ for all } \mathbf{v}_h \in \mathcal{V}_h. \quad (2.21)$$

Because \mathcal{S}_h and \mathcal{V}_h are spanned by the functions ϕ_i and ψ_j , we express them as linear combinations of the basis functions,

$$\mathbf{u}_h = \sum_{i=1}^N u_i \phi_i + \bar{u}_h, \quad (2.22)$$

$$\mathbf{v}_h = \sum_{j=1}^N v_j \psi_j. \quad (2.23)$$

Substituting these expressions for \mathbf{u}_h and \mathbf{v}_h into equation (2.21), and requiring the result to hold for any choice of $\{\mathbf{v}_1, \dots, \mathbf{v}_n\}$ leads to a discrete form of the VBVP which may be expressed as a system of linear equations,

$$\mathbf{K}\mathbf{u} = \mathbf{F}, \quad (2.24)$$

in which

$$K_{ij} = a(\phi_i, \psi_j) \text{ and } F_j = f(\psi_j). \quad (2.25)$$

The Ritz-Galerkin form is obtained by setting $\psi_i = \phi_i$. As will be seen later, the Ritz-Galerkin approximation is optimal in the sense that the approximation error is minimized with respect to the energy norm (see Section 2.2.1). Equation (2.24) is specialized to the

Navier equations, by replacing the general bilinear form of equation (2.21) with equation (2.17), and the linear form with equation (2.18) to obtain

$$\mathbf{K} = \int_{\Omega} \mathbf{B}_i^T \mathbf{D} \mathbf{B}_j \, dA, \quad (2.26a)$$

$$\mathbf{F}_i = \int_{\partial\Omega_i} \phi_i \bar{\mathbf{t}} \, ds + \int_{\Omega} \phi_i \mathbf{b} \, dA, \quad (2.26b)$$

where for plane strain problems

$$\mathbf{B}_i = \begin{bmatrix} \phi_{i,1} & 0 \\ 0 & \phi_{i,2} \\ \phi_{i,2} & \phi_{i,1} \end{bmatrix} \quad \text{and} \quad \mathbf{D} = \begin{bmatrix} \mu + \lambda & \mu & 0 \\ \mu & \mu + \lambda & 0 \\ 0 & 0 & \mu \end{bmatrix}.$$

2.2.1 Convergence

In order to arrive at results concerning convergence of Galerkin approximations, we first highlight ideas related to existence and uniqueness of solutions, error, and error measures.

Existence and uniqueness of the Ritz-Galerkin approximation are automatic, provided that existence and uniqueness have been established for the underlying VBVP. The argument for this is simple: if the bilinear form is continuous and elliptic on the entire, infinite dimensional space \mathcal{S} then it clearly is continuous and elliptic on any finite dimensional subspace $\mathcal{S}_h \subset \mathcal{S}$. Therefore, existence and uniqueness of the discrete form is inferred directly from the Lax-Milgram lemma [57].

The approximation error, \mathbf{e} , is defined as the difference between the approximate solution and the exact solution:

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_h.$$

The natural error measure is given by the energy norm as:

$$\|\cdot\| = \sqrt{a(\cdot, \cdot)}.$$

With this measure, the Ritz-Galerkin approximation may be shown to minimize the error in the energy norm, that is

$$\|\mathbf{u} - \mathbf{u}_h\| \leq \|\mathbf{u} - \mathbf{v}_h\| \quad \forall \mathbf{v}_h \in \mathcal{V}_h,$$

and further that the error is orthogonal to the approximation space. This property allows the Ritz-Galerkin solution to be interpreted as a projection of the true solution onto the approximation space.

With these definitions, we are prepared to consider convergence of the Galerkin approximation. A sequence of approximations, parameterized by h , is said to converge to the exact solution \mathbf{u} if

$$\lim_{h \rightarrow 0} \|\mathbf{u} - \mathbf{u}_h\| = 0. \quad (2.27)$$

In order to establish convergence for a particular family of basis functions we rely on Céa's lemma which provides the fundamental error estimate

$$\|\mathbf{u} - \mathbf{u}_h\| \leq C \inf_{v_h \in V_h} \|\mathbf{u} - v_h\|, \quad (2.28)$$

where C is a constant independent of the approximation space V_h . From Céa's lemma, it is clear that a sufficient condition for convergence is that $V_h \rightarrow \mathcal{V}$. That is, the Galerkin approximation will converge to the true solution provided that the sequence of spaces V_h fills *completely* the solution space \mathcal{V} as $h \rightarrow 0$. In which case, by equation (2.28) we have

$$\lim_{h \rightarrow 0} \inf_{v_h \in V_h} \|\mathbf{u} - v_h\| = 0. \quad (2.29)$$

The question of convergence now becomes one of defining a sequence of functions which fill the solution space. The intuitive solution is that because any smooth function \mathbf{u} can be expanded in a Taylor series, it can be approximated locally by polynomials. Therefore, one simple requirement is that approximation spaces contain certain polynomial solutions. In particular, for an elliptic BVP of order $2m$, the approximation space must contain all solutions of order m . In the context of displacement-based solid mechanics, this requires that an arbitrary linear displacement field be representable in the approximation space. This requirement is often referred to as the “approximability” condition in mathematical publications while in engineering literature it is often termed “completeness” or “consistency”.

The completeness of a basis can be assessed through the reproducing conditions. The reproducing conditions assert that an approximation space is complete through order m if it can exactly reproduce an arbitrary polynomial, P , of order m when nodal parameters are consistent with the nodal values of P . For example, in two-dimensions, an approximation is linearly complete if it satisfies the equality

$$u_h(\mathbf{x}) = \sum_{i=1}^N u_i \phi_i(\mathbf{x}) = a_0 + a_1 x + a_2 y, \quad (2.30)$$

for any constant values a_j , when the parameters u_i are given by

$$u_i = a_0 + a_1 x_i + a_2 y_i. \quad (2.31)$$

For second order problems the reproducing conditions are often tested for specific linear fields. The two common cases are

$$\sum \phi_i(\mathbf{x}) = 1 \quad (2.32)$$

$$\sum \mathbf{x}_i \phi_i(\mathbf{x}) = \mathbf{x}. \quad (2.33)$$

A more complete discussion of completeness is found in [5]. Provided the reproducing conditions are satisfied, the Galerkin approximation will converge.

In closing this section, we distinguish between two categories of basis functions: interpolates and approximants. Interpolates refer to a basis whose functions interpolate nodal data, i.e. $\mathbf{u}_h(\mathbf{x}_i) = \mathbf{u}_i$. These are functions which possess the Kronecker delta property $\phi_i(\mathbf{x}_j) = \delta_{ij}$. Finite elements fall into this category. Approximants are those methods for which the basis functions do not pass through the data, i.e. $\mathbf{u}_h(\mathbf{x}_i) \neq \mathbf{u}_i$. Meshless methods generally fall into this category. Despite whether a basis is composed of interpolants or approximants, it must satisfy the reproducing conditions in order to converge.

In the remaining sections, we examine common computational procedures used to generate the approximation spaces used in the Galerkin method. First the FEM method is reviewed, then as an example of meshless methods, we briefly present the Element-Free Galerkin methods, and finally we introduce the VETFEM.

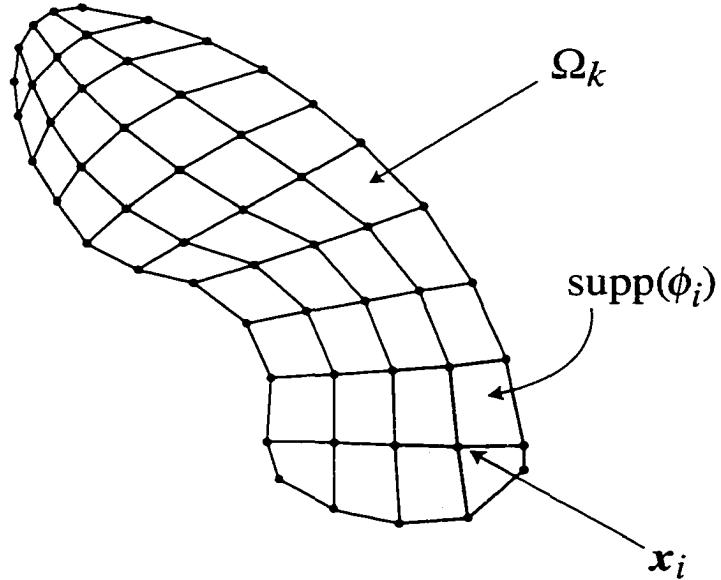


Figure 2.2: An example computational model for the FEM method

2.2.2 The finite element method

The finite element method (FEM) is the most common method for constructing approximation spaces. The FEM procedure is straightforward, and can be viewed abstractly as a three step procedure: first, a partition or mesh of the domain is created, then nodes are specified, and finally local basis functions, called shape functions, are defined.

Domain partition. A partition or mesh of the problem domain $\bar{\Omega}$ (the closure of Ω) is established by dividing it into a set of subdomains $\Omega_1, \Omega_2, \Omega_3, \dots, \Omega_M$ (see Figure 2.2). The elements are constructed with the following properties (see [56, 20]):

1. Each element consists of a nonempty interior, denoted Ω_k , and a piecewise smooth boundary $\partial\Omega_k$
2. The problem domain, Ω , is equal to the union of all the elements: $\bar{\Omega} = \bigcup_{k=1}^{LM} \bar{\Omega}_k$
3. The element interiors do not overlap: $\Omega_i \cap \Omega_j = \emptyset$, for $i \neq j$.

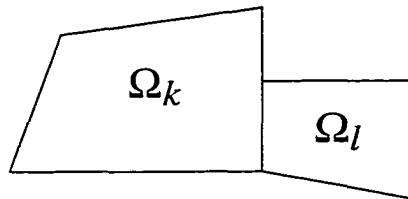


Figure 2.3: An example of an inappropriate FEM subdomain configuration

These general rules regarding element geometry do not impose restrictions on element shape or connectivity, e.g. the number of element sides is not limited. However, the practical difficulty of assigning element shape function typically restricts element geometry to n -triangular or n -rectangular shapes, and requires element edges be completely contained in either one or two elements. The latter requirement implies that configurations such as that shown in Figure 2.3 are not generally allowed.

Node specification Next, node locations are identified within the domain. Because elements are typically polygons (or mapped polygons), nodes are placed at element vertices. Additional nodes are often used to improve the approximation, but for this discussion we consider only nodes at the vertices. The nodal coordinates are written as $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$.

Finite element basis functions Having created the finite element mesh, we now form basis functions. The FEM mesh provides a convenient means for generation of global basis functions with compact support: global basis functions are defined as the union of polynomial functions that are locally defined for each element. These element-based functions are called shape functions and are generally constructed with the following attributes:

1. Shape functions form a complete basis over a single element, e.g. for our VBVP, an arbitrary linear field can be represented (completeness or approximability): $P_1(\Omega_k) \subset \phi_a^k(\mathbf{x})$, where $P_1(\Omega_k)$ is the space of polynomials of degree 1 on (Ω_k)
2. The shape functions are defined to be zero over all other elements, i.e. the support of

the shape functions is compact: $\phi_a^k(\mathbf{x}) \equiv 0$ for $\mathbf{x} \notin \Omega_k$

3. The Kronecker delta property holds with respect to nodes: $\phi_a^k(\mathbf{x}_b) = \delta_{ab}$. Note that in two and three-dimensions, shape functions are non-zero only on edges or faces that contain the node that the shape function is associated with.

Because the FEM basis functions are to be used in the Galerkin methods, the global basis functions ϕ_i are required to be members of the solution space \mathcal{S} . In particular, this demands that the generalized first derivative of the shape function must be square integrable on Ω . For this to be so, the nodal shape functions must satisfy continuity requirements across element boundaries. In the conventional FEM approach, construction of shape functions satisfying these requirements is accomplished by means of a parent element concept.

Parent element The conventional finite element approach is to define shape functions satisfying these requirements for a limited set of specific element geometries and nodal arrangements (e.g. bilinear quadrilateral). While many different elements are currently available, the element geometry is almost exclusively n -simplicial or n -rectangular. Flexibility with respect to the element shape is introduced by means of a coordinate transformation, as shown in Figure 2.4. By using this transformation, the same set of shape functions may be used for each element in the domain, and satisfaction of the required properties is guaranteed.

The finite-element-based approximation of displacement \mathbf{u}_h is then represented locally as a linear combination of shape functions

$$\mathbf{u}_h(\mathbf{x}) = \sum_{a \in C^k} \mathbf{u}_a \phi_a^k, \quad \mathbf{x} \in \Omega_k. \quad (2.34)$$

In equation (2.34), C^k is a list of nodes in element Ω_k , \mathbf{u}_a is a nodal parameter, and ϕ_a^k is the scalar valued shape function that interpolates the nodal data \mathbf{u}_a within the element Ω_k .

Discussion This construction provides the finite element method with many powerful features. Among these we note: The shape functions form a complete local basis, and

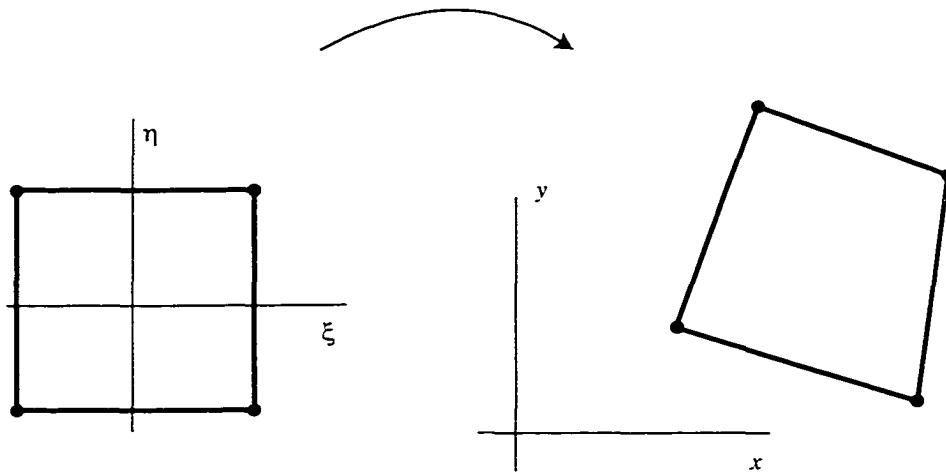


Figure 2.4: A four node quadrilateral element with the parent shown on the left. Shape functions are defined for the parent element and related to the physical element through a coordinate mapping

therefore, provide a convergent sequence as the element size is decreased. The shape functions are polynomial on element interiors and are continuous across shared element faces. This, importantly, means that basis functions formed by a collection of shape functions meet the necessary continuity requirements for admissibility in the set \mathcal{S} , and therefore convergence is guaranteed (provided sufficiently accurate evaluation of integrals). The compact-support property leads to a sparse and, with appropriate nodal numbering, banded systems of equations. The Kronecker delta property gives physical meaning to the discrete solution because the nodal parameters u_a correspond to nodal values of the solution, i.e. $u_h(x_a) = u_a$. Importantly, this property combined with precise functional variation on element edges makes accurate modeling of essential boundary conditions trivial because they may be imposed by setting the nodal parameter to the known value of the displacement: $u_a = \bar{u}(x_a)$ for $x_a \in \partial\Omega_u$. In addition, because the shape functions are well defined on the element interiors and along element edges, and element contributions to the stiffness matrix and force vector (equations (2.26)) are easily computed and because the functions are piecewise polynomials, they are easily integrated using numerical quadrature.

At the same time, there are at least two drawbacks to this approach. First, the element shape must not become too distorted. In general, interior element angles should be as equally sized as possible. A badly distorted element can degrade the accuracy of the FEM solution due to round-off errors involved in coordinate transformations, and severe distortions may potentially cause matrix singularities. Second, the use of the parent element approach requires that the domain partition consist of subdomains that conform topologically to their respective parent elements. In most situations, this demands that elements be mapped to n -simplices or n -rectangles, with n -rectangles being preferred in solid mechanics applications. From this perspective, *a priori* specification of shape functions may be viewed as a constraint on the domain partition procedure, and as previously noted, this renders application of the finite element method awkward for several important types of problems. For three-dimensional problems in solid mechanics, basis functions for hexahedral elements are preferred over those of tetrahedral elements. One reason is that hexahedral elements are computationally more efficient than tetrahedral [9]. Quadratic tetrahedral elements are required in order to achieve the same accuracy as linear hexahedral elements. Additionally, for some problems, up to an order of magnitude more linear tetrahedral elements are required to match the accuracy of linear hexahedral elements.

Meshing complex geometric domains with hexahedral elements has proven to be a significant challenge for mesh generation [10, 13]. While fully automatic, robust unstructured tetrahedral mesh generation algorithms exist that are able to create well-shaped tetrahedral elements for complex geometries, there is no fully automatic, robust algorithm capable of generating an all hexahedral mesh of arbitrary three-dimensional domains [58, 78]. The need for creating partitions consisting of hexahedral elements has prompted significant research in the area of unstructured mesh generation. Many of the current approaches are reviewed in [59], and a significant source for information concerning current meshing technologies refer to [2].

The primary difficulty in generating a hexahedral mesh can be attributed to the geo-

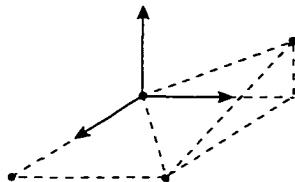


Figure 2.5: Four nodes that form a warped hexahedral face, and two planar tetrahedral faces

metric constraints involved in creating hexahedral subdivisions. In a mesh of well shaped hexahedral elements, each internal node is attached to 8 elements. Deviating from this connectivity quickly degrades the mesh quality by introducing excessive element distortion. On the other hand, the nodal connectivity is much more flexible for tetrahedral meshes. In this case, connecting each node to 20 tetrahedrons tends to optimize element shape, and this connectivity may vary by ± 10 elements and result in a quality mesh. A second reason for the complexity of quality hexahedral meshing is related to spatial placement of element vertices. For tetrahedral meshes, each element face contains three vertices, and because three distinct, but otherwise arbitrary arrangement points define a plane, any spatial arrangement defines an appropriate element face. For hexahedral elements however, each face is defined by four vertices, and an arbitrary arrangement of four points generally does not define a suitable element face, i.e. non-planar placement of vertices results in warping of element faces, see Figure 2.5. This requires significant effort to minimize distortion of hexahedral facets (see Owens [58]).

2.2.3 Meshless methods

Meshless methods avoid the difficulties associated with mesh generation by constructing the approximation subspaces from a distribution of nodes alone. In this subsection, we examine one of these methods, namely the Element-Free Galerkin method (EFG), which has been applied in the analysis of a wide range of problems in solid mechanics, and for which a wealth of information is available.

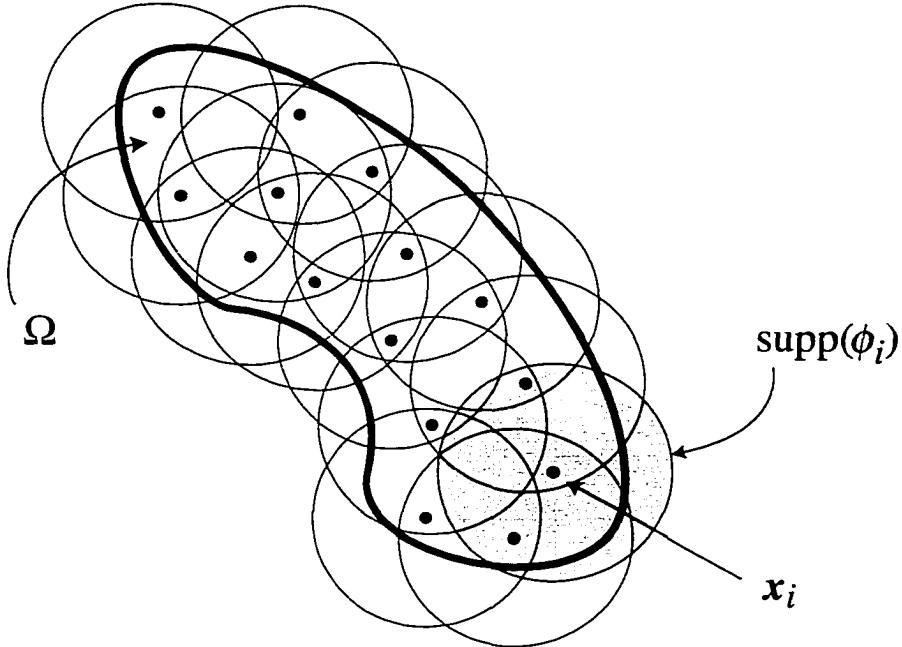


Figure 2.6: An example computational model for meshless methods

Domain discretization The domain discretization in the EFG is much easier than in the FEM. It consists of defining nodes and their associated nodal weight functions. In the EFG, nodes may be arbitrarily placed within the domain Ω at coordinates \boldsymbol{x}_i and along the domain boundary. Once nodes are placed, weight or window functions ω_i with compact support are assigned to the nodes. Figure 2.6 shows an example computational model. The nodes are represented by black dots, and circles represent the supports of the nodal weight functions. In this figure, two or three nodes lie within the weight function support, but in a typical analysis each support would contain five to ten nodes [6].

Basis functions Construction of EFG basis functions is quite different from the FEM procedure. In the FEM, a domain partition was used to construct local polynomial basis functions directly. The EFG, on the other hand, uses the mathematical theory of surface data fitting by a moving least-squares approximation [43] to generate basis functions. The

generation of EFG basis functions begins with an assumed, global representation of the displacement field followed by a weighted least-squares optimization from which nodal basis functions emerge.

Construction of basis functions begins with the following data fitting considerations. Suppose that we wish to construct an approximation u_h of a scalar valued displacement field u , and further assume that u_h is a polynomial with non-constant coefficients. We may write the local approximation of u_h as a linear combination of monomials

$$u_h = \mathbf{p}^T \mathbf{a}(\mathbf{x}) \quad (2.35)$$

where \mathbf{p} is a column vector of monomials forming a complete polynomial basis. For a quadratic basis in two-dimensions, \mathbf{p} is given by

$$\mathbf{p}^T = \begin{pmatrix} 1, & x, & y, & x^2, & xy, & y^2 \end{pmatrix} \quad (2.36)$$

$$\mathbf{a}(\mathbf{x}) = (a_1(\mathbf{x}), \dots, a_l(\mathbf{x})), \quad (2.37)$$

and \mathbf{a} is a coefficient vector of unknown functions. The parameters a_i are chosen such that the approximation is optimal in a least-squares sense. In this specific case, we seek an approximation that minimizes the error between the nodal values of u in the discrete weighted least-squares norm

$$J = \sum_{i=1}^n \omega_i(\mathbf{x} - \mathbf{x}_i) \left(\mathbf{p}(\mathbf{x}_i)^T \mathbf{a} - u_i \right)^2. \quad (2.38)$$

Here, n is the number of nodes whose support are non-zero near \mathbf{x} , u_i are nodal values of u , ω_i are nodal weight functions with compact support that are constructed from positive scalar valued functions. Equation (2.38) may be written in matrix form

$$J = (\mathbf{P}\mathbf{a} - \mathbf{u})^T \mathbf{W} (\mathbf{P}\mathbf{a} - \mathbf{u}) \quad (2.39)$$

where \mathbf{u} is a vector $N \times 1$ of unknown nodal values parameters, \mathbf{P} is the $N \times l$ Vandermonde matrix with rows given by the vector \mathbf{p} evaluated at the nodes \mathbf{x}_i ,

$$\mathbf{P} = [\mathbf{p}(\mathbf{x}_1), \dots, \mathbf{p}(\mathbf{x}_N)],$$

and \mathbf{W} is a diagonal matrix of nodal weight functions,

$$\mathbf{W} = \text{diag} [\omega_1(\mathbf{x} - \mathbf{x}_1), \dots, \omega_N(\mathbf{x} - \mathbf{x}_N)].$$

Minimizing equation (2.39) with respect to $\mathbf{a}(\mathbf{x})$, leads to the following system of linear equations

$$\mathbf{A}(\mathbf{x})\mathbf{a}(\mathbf{x}) = \mathbf{B}(\mathbf{x})\mathbf{u} \quad (2.40)$$

where \mathbf{A} and \mathbf{B} are defined as

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T \mathbf{W} \mathbf{P},$$

$$\mathbf{B}(\mathbf{x}) = \mathbf{P}^T \mathbf{W}.$$

Provided that \mathbf{A} is invertible, we may write the coefficients as

$$\mathbf{a}(\mathbf{x}) = \mathbf{A}^{-1} \mathbf{B}(\mathbf{x}) \mathbf{u}. \quad (2.42)$$

With this expression for \mathbf{a} in terms of the unknown nodal parameters \mathbf{u} the approximation \mathbf{u}_h can now be expressed in the familiar form as

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^N \phi_i \mathbf{u}_i \quad (2.43)$$

where the basis functions are given by

$$\phi_i(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}) \mathbf{A}^{-1}(\mathbf{x}) \mathbf{B}(\mathbf{x}). \quad (2.44)$$

Discussion This procedure generates basis functions having compact support from a distribution of nodes and requires no explicit nodal connectivity. Thus the EFG derives significant flexibility through the freedom from mesh generation. In principal, the method is not subject to the difficulties related to mesh generation, mesh distortion, or accommodation of evolution of the domain boundaries such as crack propagation.

Assuming that essential boundary conditions are satisfied and that the domain is convex, the EFG generates basis functions that are admissible in the weak form. Therefore, the

only question with regard to convergence is whether the basis forms a complete sequence of functions, i.e., does $\mathcal{S}_h \rightarrow \mathcal{S}$ as $h \rightarrow 0$. Completeness may be assessed by examining the construction process which is based on a minimization of the quadratic functional J in equation (2.39). Clearly if u is a polynomial spanned by \mathbf{P} , then J is minimized when $\mathbf{p}^T \mathbf{a} = u$. Therefore, the EFG generates basis functions that can reproduce any polynomial contained in \mathbf{p} and hence leads to a convergent sequence of basis functions [6]. The development of the EFG given here follows that of [7], but interestingly, the EFG basis functions can also be developed by correcting nodal weight functions in the SPH – which are based on a completely different interpolation theory – to satisfy the reproducing conditions, equation (2.30).

For domains that are non-convex, the EFG basis function can become discontinuous, resulting in a nonconforming method. Nonconforming methods are discussed more fully in Chapter 4. Convergence of the methods under these conditions is discussed in [42].

Freedom from the restrictive topology of the FEM also has its drawbacks. One of the primary difficulties with this method is that local basis functions do not interpolate nodal data. That is, the nodal parameters do not correspond to nodal values of the solution. This is a direct result of the fact that the EFG basis functions do not possess the Kronecker delta property

$$\phi_i(\mathbf{x}_j) \neq \delta_{ij}. \quad (2.45)$$

The consequence of this is that application of essential boundary conditions requires special consideration. Several approaches to enforce these conditions have been implemented including Lagrange multipliers, modified variational principals [7], and coupling with finite elements [41].

A second difficulty is integration of the bilinear and linear forms, equations (2.17) and (2.18). Integration of the shape functions is not as easy as in the FEM. The functions are not polynomial and the support is not associated with specific element domains. The most common approach is to create a background mesh of the domain and use Gauss

quadrature in each element. If the order of quadrature is not sufficient, this method can lead to significant errors in the construction of the bilinear form [26].

Meshless approximations often require significantly more computational effort than finite element approximations. Sources of this effort include: the nodal searches required to establish nodal connectivity; a system of equations must be solved at each node to generate approximation basis functions; stiffness matrix bandwidths are typically much larger than in the FEM; and high-order quadrature is required to integrate the weak form because the meshless basis is typically composed of rational functions rather than simple polynomials.

2.2.4 The VETFEM

The Variable-Element-Topology Finite Element Method represents an alternative to the conventional FEM and meshless methods. It seeks to retain all the powerful features of the FEM, while offering a level of simplicity and flexibility in the domain discretization similar to that of meshless methods. The VETFEM basis functions are piecewise polynomial shape functions defined on individual elements. However, no parent element or coordinate mapping is involved in their definition. Instead, polynomial approximations to the shape functions are defined on the element interiors through a constrained minimization process that reproduces the FEM shape function properties presented in Section 2.2.1. In this framework, element topologies are completely arbitrary, and may possess any number of nodes. Because no isoparametric transformation from a parent element is involved, there exists no convexity or other geometric regularity requirement on the elements.

The absence of specific connectivity requirements considerably simplifies automatic mesh generation. This is most clearly seen in the three mesh generation algorithms that are currently implemented. The first method is able to automatically partition an arbitrary, multiply-connected domain. The only required input is a definition of the boundary and a target number of elements. From this data a mesh is generated by first distributing a set of points within the domain, then assigning edges to pairs of points, and finally defining

elements from sets of edges. The second method creates a partition by generating Voronoi cells from a distribution of points. This procedure is entirely a geometric construction and does not require heuristics of any kind. While it is fully automatic, the current implementation does not mesh multiply-connected domains. The third method is a simple recursive algorithm by which user defined regions are subdivided. This method requires the user to provide a subdivision of the domain into a set of nearly convex subdomains.

Chapter 3

VETFEM

In this chapter, the Variable-Element-Topology Finite Element Method is presented as a means to generate basis functions for Galerkin discretization of a two-dimensional, total-Lagrangian formulation of solid mechanics. As in the conventional finite element method, the displacement field is unknown, and is to be determined by seeking an approximate solution to the weak form of the equilibrium equations.

As in the conventional FEM, the VETFEM utilizes a domain partition consisting of non-overlapping elements and associated nodes to generate finite-dimensional approximation spaces. One basis function Φ_i for each node is then defined by composing the shape functions ϕ_i^k for the surrounding elements. The basis functions exhibit the Kronecker delta property. The shape functions are nodal interpolants with compact support, in the sense that $\phi_i^k(\mathbf{x}) \equiv 0 \forall \mathbf{x} \notin \Omega_k$. The VETFEM basis functions are constructed as an assembly of shape functions, and hence exhibit the compact support property as well.

3.1 VETFEM shape functions

The VETFEM shape functions differ from conventional FEM shape functions in that they do not derive from an isoparametric mapping of a parent element. Instead, they are constructed in the physical coordinates of the element by a robust and stable constrained

minimization procedure. The advantage of this approach is that it imposes few restrictions on the topology (number and connectivities among element nodes) of individual finite elements, in contrast to the isoparametric procedure of the conventional FEM. This allows for simpler and more efficient mesh generation. Shape functions constructed using the VETFEM may not form basis functions with the requisite C^0 continuity along shared element edges. The VETFEM procedure attempts to provide continuity by generating functions that vary linearly over each element edge, allowing the shape functions to be determined independently for each element. Exact linear variation, however, may not be possible to achieve because the exact form of the polynomial will depend on the number and arrangement of element nodes as well as on the chosen degree of the polynomial. In some cases, therefore, the composite basis functions will have slight discontinuities at element boundaries, making the VETFEM a nonconforming method. This “variational crime” renders the VETFEM approximation spaces inadmissible in the unmodified variational principle because they are no longer square-integrable.

The inadmissibility of nonconforming approximation spaces does not imply that the quality of nonconforming solutions is adversely affected. On the contrary, for many types of problems, nonconforming approximations exhibit superior performance to conforming ones. It does, however, require care be taken to ensure that the overall method is convergent. In the VETFEM, this is addressed in the form of constraints imposed on the shape function generation. We refer to these constraints as “consistency constraints,” which along with additional “node-point constraints” form the first step of a three-step process used to select the polynomial coefficients for each shape function. The “node-point constraints” relate to the requirement that the shape functions should take values one or zero at each node, in order that the Kronecker delta property of FEM basis functions be preserved. This property greatly simplifies the imposition of displacement boundary conditions, and allows for straightforward accommodation of interface conditions.

The procedure consists of sequentially applying constraints and optimizations to gener-

ate the polynomial shape function coefficients. The constraints are imposed on the space of polynomials such that the resulting shape functions will possess the Kronecker delta property, will be linearly complete, and will exhibit a necessary property for convergence that is made precise below. The latter constraint is necessary because of the possible lack of strict C^0 -continuity at inter-element boundaries, and the departure from the standard Galerkin prescription which this implies. After identifying the space of polynomials satisfying these constraints, a series of at most two minimizations are applied to ensure that the shape functions possess optimal linear variation on each element edge, and optimal smoothness on the element interior.

The form and construction of element-based, nodal shape functions are presented in this chapter as follows: the polynomial form of the VETFEM basis functions is described in Section 3.1, and concepts related to convergence are introduced in Section 3.2. Section 3.3 details the shape function construction procedure which includes: formulation and imposition of the node-point and consistency constraints (Sections 3.3.1 – 3.3.3), and boundary (Section 3.3.4) and domain functional minimizations (Section 3.3.5). Equation assembly, and construction of accurate integration rules on the arbitrary-geometry elements, are presented in Subsection 3.4. An outline of the complete procedure is given in Subsection 3.5, and example shape functions are presented in Section 3.6.

3.2 Polynomial shape functions

This section describes the polynomial form and desired properties of the VETFEM shape functions. As a first step, consider the geometric regions for which VETFEM shape functions are defined. Figure 3.1 depicts a general, two-dimensional, finite region, Ω . This region is a polygon with n vertices. Its boundary $\partial\Omega$ is composed of straight line segments, here called faces and labeled $\{\Gamma_1, \dots, \Gamma_n\}$, and distinct nodes (locally numbered $a = 1, \dots, n$) that are located at the polygon vertices. The number of faces and nodes is arbitrary. The faces are non-intersecting line segments, but with an otherwise arbitrary

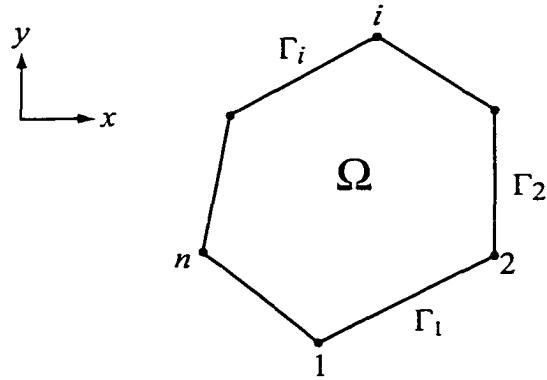


Figure 3.1: A typical VETFEM element in \mathbb{R}^2

orientation. This means, for example, that the angle between two adjacent faces is not restricted, and may include 180° as well as angles approaching 0° and 360° . However, angles at these extremes should be avoided (see Section 3.7.3).

Now consider the form of the shape functions. The present objective is to define node based polynomial shape functions $\phi_a(\mathbf{x})$, ($a = 1, \dots, n$) over Ω that may be written as

$$\phi_a(\mathbf{x}) = \sum_{i=1}^m G_i^a p_i(\mathbf{x}) = \mathbf{G}^a \cdot \mathbf{p}(\mathbf{x}), \quad \text{for } \mathbf{x} \in \partial\Omega, \quad (3.1)$$

In equation (3.1), p_i form a basis for the nodal shape functions and are monomials in \mathbf{x} as appearing in Pascal's triangle (Figure 3.2), and m is the number of terms selected for the polynomial. Normally, m is selected so that the polynomial basis consists of complete rows of Pascal's triangle. For example, a cubic basis in two-dimensions would include the first four rows of Pascal's triangle, and \mathbf{p} would be given by

$$\mathbf{p}(\mathbf{x}) = \left(1, \quad x, \quad y, \quad x^2, \quad xy, \quad y^2, \quad x^3, \quad x^2y, \quad xy^2, \quad y^3 \right). \quad (3.2)$$

In addition, m is chosen based on the geometric complexity of the element Ω , as determined by the number of nodes n in the element (see Subsection 3.8).

Because the functions $\phi_a(\mathbf{x})$ are to be used as shape functions in a Galerkin discretization of the weak form of equilibrium, they should be defined so that:

$$\begin{array}{c} 1 \\ x \quad y \\ x^2 \quad xy \quad y^2 \\ x^3 \quad x^2y \quad xy^2 \quad y^3 \end{array}$$

Figure 3.2: four rows of Pascal's triangle

1. The Kronecker delta property of finite element shape functions is preserved
2. The overall method is consistent, in a sense that is made precise below
3. Each shape function varies linearly over each facet to the extent possible
4. Each shape function is as smooth as possible over the element interior.

Constructing the shape function is accomplished by first choosing the number of monomial terms m and then selecting the coefficients \mathbf{G}_i^a using a three-step procedure, which is the subject of the following subsections. This process requires the solution of small systems of linear equations in which the coefficient matrices are determined by the element geometry, and are fixed for all shape functions for a given element. The right-hand side vectors are, in general, different for each node of the element, so that the coefficients \mathbf{G}_i^a are generated by repeated back-substitution of the factored equations for the element. Before describing these steps in detail, however, the fundamental concepts related to convergence which motivate the particular form of the construction process are discussed.

3.3 Convergence

The VETFEM basis functions may exhibit discontinuities on internal element boundaries, making the VETFEM a nonconforming method. Accordingly, the method constructs an approximate solution from functions that lie outside the function space that contains the solution. Hence, a nonconforming approximation is termed an “external approximation”.

This may be expressed in the notation of Section 2.2, as

$$\mathcal{S}_h \not\subset \mathcal{S}. \quad (3.3)$$

The mathematical effect of the discontinuities is that the basis functions fail to be square-integrable, rendering the approximation functions inadmissible in the variational problem. Thus, all the results related to convergence of the conforming, admissible Galerkin approximations do not apply to the nonconforming approximation. The question of convergence then becomes central.

Nonconforming elements are used extensively in solid mechanics applications to enhance FEM approximations for certain classes of problems. In particular, nonconforming element formulations such as those found in [66, 72, 77] have proven to be effective for problems involving incompressible or near-incompressible materials.

Fundamental to convergence of nonconforming methods are the concepts of completeness and consistency. Completeness refers to the availability of locally exact solutions and was discussed in Section 2.2.1 on page 18. Consistency, on the other hand, refers to the ability of the method to recover the exact solution of the BVP, if it is available in the approximation space. This is analogous to the finite difference notion of consistency, in which consistency is assessed in terms of the polynomial degree to which the Taylor expansion of the exact solution satisfies the discrete equations. In Galerkin-based methods consistency may be assessed, for example, by checking whether an exact linear solution satisfies the discrete equations. In the FEM, provided no variational crimes are committed, the Lax Milgram lemma [21, 52, 57] guarantees that the method is consistent, i.e. if the exact solution is present in the approximation space, it will be selected. In this case, then, convergence is reduced to a question of completeness. However, the Lax Milgram lemma is not applicable to nonconforming methods; therefore, consistency must be assessed in order to determine whether the method is convergent.

3.3.1 Completeness

As in the conventional FEM, the VETFEM function space must be complete to be convergent. Completeness of the approximation space relates to the ability to generate a convergent sequence of spaces as the mesh size parameter $h \rightarrow 0$. In the case of displacement-based solid mechanics, the approximation space is complete if it contains all linear polynomials. As mentioned previously, VETFEM shape functions may not exactly reproduce the desired linear variation on each element face for all element geometries. Therefore, the composite basis functions may exhibit slight discontinuities along inter-element boundaries. This situation, however, does not preclude a linear combination of these discontinuous functions from exactly interpolating an arbitrary linear function over the problem domain.

Completeness of VETFEM shape functions is a result of choosing a polynomial basis that contains all linear terms and optimization procedures that yield a linear solution when nodal values consistent with a linear variation are specified (see Subsections 3.3, 3.4, and 3.9). While completeness of the VETFEM basis functions has been verified numerically, it has not yet been proven analytically.

3.3.2 Consistency

In the case of nonconforming methods, the verification of consistency is essential. The reason consistency comes into question is related to the manner in which the discrete equations are constructed. Specifically, the element global residual is evaluated on an element-by-element basis and simply assembled. For conforming methods, this procedure is equivalent to exact evaluation of the residual on the entire domain. But, when this procedure is applied to nonconforming methods, it fails to account in any manner for the presence of the discontinuities in the global basis functions (see Strang and Fix, 1973, p. 174).

With reference to the notation introduced in Subsection 2.1.2, ignoring the discontinuities in evaluation of the global residual constitutes a shift from the exact bilinear form to

an approximate form. Consider the exact form of the variational problem represented as

$$a(\mathbf{u}_h, \mathbf{v}_h) = f(\mathbf{v}_h) \text{ for all } \mathbf{v}_h \in \mathcal{V}_h. \quad (3.4)$$

Here the bilinear form $a(\cdot, \cdot)$ is given by equation (2.17), and involves integration of the inner product $\sigma(\nabla \mathbf{u}_h) : \nabla \mathbf{v}_h$ over the entire domain. This integration presents a problem for nonconforming methods because \mathbf{u}_h and \mathbf{v}_h are composed of discontinuous basis functions. Thus, computation of the bilinear form involves integrating products of delta functions, which is not a well defined operation. In practice however, the bilinear form is computed element-by-element, and on a single element \mathbf{u}_h and \mathbf{v}_h are composed of continuous shape functions. Therefore, because the discontinuities in the basis are limited to the element boundaries, their affect is not included in the computation. Thus, when the bilinear form is computed in this manner, it is may only be considered an approximation. We represent this approximate bilinear form as

$$a_h(\mathbf{u}_h, \mathbf{v}_h) = f(\mathbf{v}_h) \text{ for all } \mathbf{v}_h \in \mathcal{V}_h. \quad (3.5)$$

Note that the linear form $f(\mathbf{v}_h)$, given by equation (2.18), does not contain derivatives of the shape functions and is therefore evaluated exactly by the element assembly procedure, even for discontinuous basis functions.

From the perspective of shifting to an alternative variational statement, it is clear that there is no reason that solutions to equations (3.4) and (3.5) should converge to the same values, even for the same data $f(\mathbf{v}_h)$. This is why both completeness and consistency are important: for example, although a nonconforming method may be complete, data $f(\mathbf{v}_h)$ corresponding to a linear solution \mathbf{u}_h in equation (3.4) may or may not correspond to the linear solution of the approximate form in equation (3.5).

Consistency requires that both systems do, in fact, converge to the same solution. From a practical standpoint, the “minimum” level of consistency for a second order BVP may be interpreted as a requirement that data $f(\mathbf{v}_h)$, corresponding to an arbitrary linear solution to the desired variational form (3.4), corresponds to exactly the same linear solution in the

approximate form (3.5).

This type of consistency may be achieved in the presence of discontinuous basis functions by imposing some conditions on the formulation of the shape functions. These conditions manifest themselves as constraints on the selection of the polynomial coefficients G_i^a in equation (3.1). The original VETFEM formulation of the consistency constraints, as presented in [64], arose from intuitive arguments related to equilibrium. These “consistency constraints” emerge from consideration of a uniform stress state: the overall method must select the exact, linear displacement field in the case where boundary conditions consistent with the uniform stress field are applied to the body. Specifically, the weak statement of equilibrium

$$\int_{\Omega} \Phi_{a,j} \sigma_{ij} \, dA = \int_{\partial\Omega} \Phi_a t_i \, ds, \quad a = 1, \dots, N, \quad (3.6)$$

must be identically satisfied for any discretization of the body Ω , where t_i are components of traction consistent with the constant, but otherwise arbitrary, stress measure σ_{ij} , and Φ_a is the basis function associated with global node a . (Refer to Table A.2 on page 128 for symbol definitions.) Introducing the relation $t_i = \sigma_{ij} n_j$, where n_j are components of the outward unit normal on $\partial\Omega$, equation (3.6) leads immediately to the requirement

$$\int_{\Omega} \Phi_{a,j} \, dA = \int_{\partial\Omega} \Phi_a n_j \, ds. \quad (3.7)$$

When $\Phi_a \in C^0(\Omega)$, equation (3.7) is simply a statement of the divergence theorem (A.12). However, when the basis functions Φ_a are composed of incompatible shape functions, this equality is not implied in general, and conditions must be imposed such that equation (3.7) continues to hold. Proceeding in this direction, it is first observed that the left-hand expression in equation (3.7) is typically replaced by a sum over element contributions:

$$\sum_{m \in K^a} \int_{\Omega_m} \phi_{a,j}^m \, dA = \int_{\partial\Omega} \Phi_a n_j \, ds. \quad (3.8)$$

In equation (3.8), K^a is a list of elements that contain node a . This substitution follows directly from equation (3.7) in the case of continuous basis functions. If the shape functions ϕ_a^m are incompatible, on the other hand, then equation (3.8) does not strictly follow from

equation (3.7) because the Dirac delta contributions on the element faces are ignored by introduction of the sum of integrals over the elements. The present objective is to elucidate the conditions under which (3.8) does hold, even for discontinuous basis functions Φ_a .

The element shape functions are smooth polynomials within element domains. Accordingly, applying the divergence theorem to each term in the sum on the left-hand side of (3.8) gives

$$\sum_{m \in K^a} \int_{\partial\Omega_m} \phi_a^m n_j \, ds = \int_{\partial\Omega} \Phi_a n_j \, ds, \quad (3.9)$$

in which reference is again made to the notation and data sets defined in Table A.2. From (3.9), a sufficient condition for the weak form of equilibrium, equation (3.9), to hold is

$$\int_{\Gamma_l} \phi_a^m \, ds = \int_{\Gamma_l} \psi_a \, ds \quad \forall l = 1, \dots, M \text{ and } a = 1, \dots, N \quad (3.10)$$

where $a = C_a^m$. In equation (3.10), the functions ψ_a (one corresponding to each node in the mesh) are defined on element faces only, and need not be defined elsewhere. Equation (3.10) constitutes a weakened continuity for element basis functions requiring that only the zero-order moment of the shape functions be continuous across element edges.

The sufficiency of the equality in equation (3.10) to guarantee satisfaction of weak equilibrium is established by observing that both sides of equation (3.9) may be replaced by an appropriate assembly of contributions of the type given by equation (3.10). With reference to the left-hand side of equation (3.9), “appropriate assembly” means a summation over all faces Γ_l that make up the element boundary $\partial\Omega_m$, taking due account of the proper sign for each term in the sum. Specifically, the unit normal vector \mathbf{n}^l to Γ_l may be either an outward or an inward unit normal to element Ω_m ; in the latter case, the corresponding integral in equation (3.10) is subtracted from the summation and not added. The right-hand side of equation (3.9) may similarly be evaluated as the sum of integrals over the appropriate faces, again taking care to assign the proper sign to each term. Carrying out these substitutions in equation (3.9), it is readily seen that, on the left-hand side, the two contributions corresponding to each interior face will cancel pair-wise, leaving only the single contributions from the faces that lie on $\partial\Omega$. These, in turn, will balance the assembly on

the right-hand side. From this argument, it is seen that the key requirement engendered by equation (3.10) relates to the fact that the functions ψ_a are defined for each node, without reference to elements.

In the case of compatible shape functions, equation (3.10) holds automatically, with ψ_a taking the common values of the shape functions from the two elements that share the face (for internal faces). For incompatible shape functions, (3.10) must be imposed. This equality forms the basis for the “consistency constraints” discussed in Subsection 3.4.2. In the current two-dimensional implementation of the VETFEM, ψ_a are chosen with linear variation from one at node a , to zero at all other nodes that are connected by element faces to node a .

In the first implementation of VETFEM, which is presented in [64], the consistency constraints were formulated in terms of a weaker condition than that of equation (3.10). In that approach, rather than enforcing zero-order moment continuity across each element edge independently, the equality of (3.10) was required with respect to the entire element boundary, i.e. the face integrals were changed to boundary integrals

$$\int_{\partial\Omega} \phi_a^m n_j \, ds = \int_{\partial\Omega} \psi_a n_j \, ds, \quad a = 1, \dots, n_m. \quad (3.11)$$

This subtle distinction leads to consistency constraints in the form

$$\int_{\Omega_m} \phi_{a,j}^m \, dA = \int_{\partial\Omega_m} \psi_a n_j \, ds, \quad a = 1, \dots, n_m. \quad (3.12)$$

This form of constraint has the advantage that the domain integration of the left-hand side of (3.12) is well suited to evaluation by general numerical integration routines such as that discussed in Section 3.5. Also, the number of constraints for each element is limited to the spatial dimensionality of the BVP. In contrast, constraints of the form (3.10) that are applied on each edge require numerical integration of the shape functions on each element edge, and the number of constraints imposed on each shape function is equal to the number of element edges.

In this form, while contributions to equation (3.9) associated with internal nodes and

faces will cancel cumulatively, it is not clear whether contributions from external nodes evaluated along external faces will maintain strict satisfaction with the left-hand side of equation (3.9). In particular, consider application of a constant stress along a single element edge. With constraints formulated according to (3.10), the integral of a shape function times the constant stress would be equal to the integral of the reference function times the same constant stress, and the nodal forces would be computed exactly. On the other hand, with constraints formulated according to (3.12), nodal forces along a single edge may not be computed exactly, introducing a slight perturbation into the linear form. Nevertheless, the VETFEM implemented with this form of constraints passes the patch test [64]; however, more investigation is required to validate this form of the consistency constraints.

Subsequent to initial derivation and implementation of the VETFEM, a slightly generalized form of equation (3.10) was found to be a classic requirement for nonconforming methods known as the Céa test [75, 12]. This equality, which is discussed more in the following section, requires numerical integrations to be performed on each element face. The resulting number of constraints on shape functions is therefore equal to the number of faces contained by that element.

Constraints in the form (3.10), ensure that a linear solution to the exact variational BVP (3.4) corresponds to the same linear solution in the approximate variational BVP (3.5). On the other hand, constraints in the form (3.12) may perturb the linear form for certain types of boundary data it may not be possible to establish a similar equivalence. In the sections that follow, application of both types of constraints is explained. This presentation describes the basis from which the constraints utilized in the VETFEM were formulated. A more thorough examination of convergence is included in Chapter 4.

3.4 Shape function construction

This section describes the construction process by which a unique set of shape functions possessing the properties listed in Section 3.2 are generated for a specific element. The

first two subsections describe imposition of the constraints related to the Kronecker delta and consistency requirement of equation (3.10). The next two subsections describe the minimizations which serve two important purposes: first, in combination with the Kronecker delta constraints, they ensure that the method is complete; and second, they ensure that the procedure will identify a unique set of shape functions for any element geometry satisfying the requirements listed in Section 3.2.

3.4.1 Kronecker delta constraints

The first step in determining the values of the polynomial shape function coefficients \mathbf{G}_i^a is identification of the space of polynomials exhibiting the required Kronecker delta property. This property is important because it makes application of displacement boundary conditions straightforward.

Imposition of the Kronecker delta property is straightforward, because functions possessing this property will satisfy the equations:

$$\phi_a(\mathbf{x}_b) = \delta_{ab}, \quad (3.13)$$

In equation (3.13), $\phi_a(\mathbf{x}_b)$ is the shape function associated with node a , evaluated at node b . δ_{ab} is the Kronecker delta, which takes the value of one when $a = b$, and zero otherwise. Substituting the expression for element shape functions, equation (3.1), into equation (3.13) transforms the Kronecker delta condition into a multivariate polynomial interpolation, which can be expressed as an algebraic system of linear equations

$$\mathbf{P}\mathbf{G} = \mathbf{I}. \quad (3.14)$$

In equation (3.14), \mathbf{P} is an $(n \times m)$ Vandermonde matrix with monomial entries p_i evaluated

at the n distinct element nodes $\mathbf{x}_1, \dots, \mathbf{x}_n$,

$$\mathbf{P} = \begin{bmatrix} p_1(\mathbf{x}_1) & p_2(\mathbf{x}_1) & \cdots & p_m(\mathbf{x}_1) \\ p_1(\mathbf{x}_2) & p_2(\mathbf{x}_2) & \cdots & p_m(\mathbf{x}_2) \\ \vdots & \vdots & & \vdots \\ p_1(\mathbf{x}_n) & p_2(\mathbf{x}_n) & \cdots & p_m(\mathbf{x}_n) \end{bmatrix}, \quad (3.15)$$

and \mathbf{G} is the $(m \times n)$ matrix whose columns are vectors of shape function coefficients. Each column of \mathbf{G} , denoted \mathbf{G}^a , represents the shape function associated with node a :

$$\mathbf{G} = \begin{bmatrix} G_1^1 & G_1^2 & \cdots & G_1^n \\ G_2^1 & G_2^2 & \cdots & G_2^n \\ \vdots & \vdots & & \vdots \\ G_m^1 & G_m^2 & \cdots & G_m^n \end{bmatrix}. \quad (3.16)$$

Finally, \mathbf{I} is the $(n \times n)$ identity matrix. Note that the node point constraint equation (3.14) only need be factorized once per element for determination of all the element shape functions. Equation (3.14) has at least one solution provided that \mathbf{I} is in the range of \mathbf{P} . It is clear that \mathbf{I} is contained in \mathbf{P} , provided that the rows of \mathbf{P} form complete polynomials of order $k \geq n-1$ and \mathbf{P} is of full rank (i.e. the points are distinct). Thus, the polynomial order of the approximation, k , is always chosen to exceed the order required to interpolate the nodal data. Because \mathbf{P} is a Vandermonde matrix, it is guaranteed to be full rank, provided that the node points, $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, are distinct [24], which is true by supposition.

Because equations (3.14) must be satisfied by the shape function coefficients, they restrict the space of polynomials used as shape functions, and reduce the dimensionality of the polynomial function space from m to $m - n$. The procedure use to identify this space is detailed in Section 3.4.3, after formulating the consistency constraints.

3.4.2 Consistency constraints

The second set of constraints are the consistency constraints, and in the development of the VETFEM method these have been implemented in two forms. In the initial implementation

of the consistency constraints, the shape functions were selected such that equations (3.12) were satisfied. It is this form that was presented in [64]. The current formulation of the VETFEM, requires that equation (3.10) be satisfied, and this version is presented in this dissertation.

As discussed previously, consistency of the nonconforming approximation with the exact variational boundary value problem (VBVP) can be related to satisfaction of the weak form of equilibrium (3.9). From equation (3.10), it is clear that equation (3.9) will be satisfied, provided that the first moment of the shape functions are continuous across any element interface:

$$\int_{\Gamma_j} \phi_a \, ds = \int_{\Gamma_j} \psi_a \, ds. \quad (3.17)$$

In equation (3.17), the ψ_a is a reference function defined to vary linearly on Γ_j . Substituting (3.1) for ϕ_a yields,

$$\int_{\Gamma_j} p_i G_i^a \, ds = \int_{\Gamma_j} \psi_a \, ds, \quad (3.18)$$

where the repeated subscript implies summation. Because the coefficients G_i^a are constant, equation (3.18) may be rewritten as

$$G_i^a \int_{\Gamma_j} p_i \, ds = \int_{\Gamma_j} \psi_a \, ds. \quad (3.19)$$

The left-hand side integrals depend only on the element shape and are independent of the particular shape function; therefore, the equations for all shape functions of the element may be grouped into the single system of linear equations

$$\mathbf{D}\mathbf{G} = \mathbf{B}. \quad (3.20)$$

The matrix \mathbf{D} has dimension $(n \times m)$, and components that are given by

$$D_{ij} = \int_{\Gamma_i} p_j \, ds, \quad (3.21)$$

where p_j are monomials from Pascal's triangle and Γ_i is edge i of the element. These entries are readily evaluated using numerical quadrature. The components of \mathbf{B} are integrals of

the reference functions ψ_a , which, in the current VETFEM, are chosen to vary linearly on element edges, and are trivially integrated. These conditions represent a further reduction in the dimensionality of the shape function equation. Provided that \mathbf{D} is of full rank, the final dimension of the polynomial space is $(m - 2n)$.

3.4.3 Constraint application

Both of the foregoing sets of constraints must be satisfied by the element shape functions, and therefore the coefficients \mathbf{G} must be chosen so that satisfaction of both (3.14) and (3.20) is achieved. Although various approaches may be taken to identify the space of polynomials satisfying these equations (e.g. Gram-Schmidt algorithm), Gaussian elimination provides a simple and effective means by which the constraints may be imposed simultaneously. First, both sets of constraints are combined into a single system of equations

$$\mathbf{A}\mathbf{G} = \mathbf{X}, \quad (3.22)$$

where,

$$\mathbf{A} = \begin{bmatrix} \mathbf{P} \\ \mathbf{D} \end{bmatrix}, \text{ and } \mathbf{X} = \begin{bmatrix} \mathbf{I} \\ \mathbf{B} \end{bmatrix}.$$

The $2n$ linear constraints represented by (3.22) may not be linearly independent. Gaussian elimination by row reduction is used to classify the coefficients into two groups: bound and free. Bound coefficients are eliminated in the course of the reduction, and are therefore computable once all free unknowns have been assigned values. Coefficients not eliminated are termed free and may be assigned an arbitrary real value without affecting the satisfaction of the constraint equation (3.22). The relationship between the bound and free coefficients is generated by the elimination algorithm which is described in detail in Appendix B.1. The algorithm returns a system of equations that leads to the following relationship

$$\mathbf{G} = \mathbf{S}\bar{\mathbf{G}} + \mathbf{C}. \quad (3.23)$$

In equation (3.23), $\bar{\mathbf{G}}$ is an $(j \times n)$ matrix of free coefficients, where j is the number of free coefficients. Matrices \mathbf{S} ($m \times j$) and \mathbf{C} ($m \times n$) are returned by the reduction algorithm in

B.1. The shape function coefficient vectors appearing as columns of \mathbf{G} yield shape functions that satisfy both the node-point and the consistency constraints for arbitrary coefficient matrices $\bar{\mathbf{G}}$. In the case that \mathbf{A} is $(n \times n)$ and of full rank, \mathbf{S} will be dimension zero, and the coefficients uniquely determined by,

$$\mathbf{G} = \mathbf{C} = \mathbf{A}^{-1} \mathbf{X}. \quad (3.24)$$

If \mathbf{A} is rectangular and/or rank deficient, then the shape function coefficients have not been completely specified by the constraint equations, and the free coefficients, represented by columns of matrix $\bar{\mathbf{G}}$, are chosen in steps two and three of the shape function construction procedure which are described in Sections 3.4.4 and 3.4.5.

3.4.4 Compatibility optimization

The objective of step two in the shape function construction process is to further restrict the coefficient vectors \mathbf{G}^a , so that the shape functions exhibit a variation on each face which is as nearly linear as possible. Initially, this concept was the basis for the entire VETFEM development. As understanding with regard to convergence increased, constraints and other minimizations were added to bring the method to its current form.

The degree to which linear variation on faces can be achieved by the polynomial shape functions depends both on the geometry of the element, as well as on the order of the polynomial basis. Because linear variation of the shape functions is related to continuity of the global basis functions, step two represents “compatibility optimization” of the shape functions.

The polynomial shape functions may be compelled toward piecewise-linear variation on the element boundary (i.e. linear on each face) by minimizing the square-magnitude of their derivatives in the direction of the tangent to the boundary. Accordingly, vectors \mathbf{G}^a are sought such that

$$F_a = \int_{\partial\Omega} \left[\nabla \phi_a \cdot \nabla \phi_a - (\nabla \phi_a \cdot \mathbf{n})^2 \right] ds \quad (3.25)$$

is minimized for each $a = 1, \dots, n$. By substituting (3.1) for ϕ_a , equation (3.25) may be written as

$$F_a = \sum_{i=1}^m \sum_{j=1}^m \int_{\partial\Omega} [\nabla(p_i G_i^a) \cdot \nabla(p_j G_j^a) - (\nabla(p_i G_i^a) \cdot \mathbf{n}) (\nabla(p_j G_j^a) \cdot \mathbf{n})] \, ds \quad (3.26)$$

$$= \int_{\partial\Omega} [(\nabla p_i \cdot \nabla p_j) - (\nabla p_i \cdot \mathbf{n}) (\nabla p_j \cdot \mathbf{n})] G_i^a G_j^a \, ds. \quad (3.27)$$

Because the coefficients are constant, (3.27) may be expressed in matrix form as

$$\mathbf{G}^{aT} \bar{\mathbf{A}} \mathbf{G}^a, \quad (3.28)$$

where

$$\bar{A}_{ij} = \int_{\partial\Omega} (p_{i,k} p_{j,k} - p_{i,k} p_{j,l} n_k n_l) \, ds \quad (i, j = 1, \dots, n). \quad (3.29)$$

$\bar{\mathbf{A}}$ is clearly symmetric, and its entries are easily evaluated integrals of polynomials over the element boundary. Because the components of $\bar{\mathbf{A}}$ are independent of the shape function ϕ_a , equation (3.28) may be generalized to include all element shape functions:

$$\mathbf{F} = \mathbf{G}^T \bar{\mathbf{A}} \mathbf{G}, \quad (3.30)$$

where, \mathbf{F} is an $n \times l$ column vector with entries F_a . Now, considering the constraints imposed on the coefficients of \mathbf{G} , (3.30) may be rewritten using equation (3.23) yielding a modified system

$$\mathbf{F} = (\mathbf{S} \bar{\mathbf{G}} + \mathbf{C})^T \bar{\mathbf{A}} (\mathbf{S} \bar{\mathbf{G}} + \mathbf{C}) \quad (3.31)$$

$$= (\bar{\mathbf{G}}^T \mathbf{S}^T + \mathbf{C}^T) \bar{\mathbf{A}} (\mathbf{S} \bar{\mathbf{G}} + \mathbf{C}) \quad (3.32)$$

$$= \bar{\mathbf{G}}^T \mathbf{S}^T \bar{\mathbf{A}} \mathbf{S} \bar{\mathbf{G}} + 2 \bar{\mathbf{G}}^T \mathbf{S}^T \bar{\mathbf{A}} \mathbf{C} + \mathbf{C}^T \bar{\mathbf{A}} \mathbf{C} \quad (3.33)$$

Minimization of each \mathbf{F} with respect to the free parameters $\bar{\mathbf{G}}$ (see (3.23)) yields

$$\frac{\partial \mathbf{F}}{\partial \bar{\mathbf{G}}^T} = 2 \mathbf{S}^T \bar{\mathbf{A}} \mathbf{S} \bar{\mathbf{G}} + 2 \mathbf{S}^T \bar{\mathbf{A}} \mathbf{C} = \mathbf{0}, \quad (3.34)$$

and rearranging (3.34) leads to the system of equations

$$\mathbf{S}^T \bar{\mathbf{A}} \mathbf{S} \bar{\mathbf{G}} = -\mathbf{S}^T \bar{\mathbf{A}} \mathbf{C}. \quad (3.35)$$

The elimination algorithm (Appendix B.1) is invoked to treat (3.35), from which emerges a representation of the free coefficients $\hat{\mathbf{G}}$ in the form $\hat{\mathbf{G}} = \bar{\mathbf{S}}\hat{\mathbf{G}} + \bar{\mathbf{C}}$. This representation is combined with (3.23) to yield

$$\mathbf{G} = \mathbf{S}\hat{\mathbf{G}} + \mathbf{C} \quad (3.36)$$

$$= \mathbf{S}(\bar{\mathbf{S}}\hat{\mathbf{G}} + \bar{\mathbf{C}}) + \mathbf{C} \quad (3.37)$$

$$= \mathbf{S}\bar{\mathbf{S}}\hat{\mathbf{G}} + \mathbf{S}\bar{\mathbf{C}} + \mathbf{C}. \quad (3.38)$$

Letting $\hat{\mathbf{S}} = \mathbf{S}\bar{\mathbf{S}}$ and $\hat{\mathbf{C}} = \mathbf{S}\bar{\mathbf{C}} + \mathbf{C}$, equation (3.36) may be written as,

$$\mathbf{G} = \hat{\mathbf{S}}\hat{\mathbf{G}} + \hat{\mathbf{C}}. \quad (3.39)$$

In the event that the matrix of free coefficients $\hat{\mathbf{G}}$ is of dimension zero, as is often the case, the shape function coefficients are completely determined by the constraints and the compatibility optimization. However, if the dimension is greater than zero, the coefficient vectors \mathbf{G}^a are not completely determined after the compatibility optimization, which, after all, only references the behavior of the shape functions on the element boundary. Two common situations in which free coefficients are present after the compatibility optimization are when exact linear variation is achieved along element edges, and when the polynomial basis allows for the formation of “bubble functions”, (i.e. nonzero functions that are zero on the element boundary) over the element domain¹. When free coefficients remain, a third step is invoked, in which these coefficients are set by optimizing the smoothness of the shape functions over the element interior.

3.4.5 Smoothness optimization

The third and final step of the shape function construction process is invoked to set any remaining free polynomial coefficients following the imposition of the constraints and the compatibility optimization steps. In this step, the smoothest available polynomial is selected

¹Both these situations are encouraged by a high-order polynomial basis, and by a geometrically simple element.

by minimizing the square-magnitude of the shape function gradients, averaged over the interior of the element. Specifically, the quadratic forms

$$\hat{F}_a = \int_{\Omega} \nabla \phi_a \cdot \nabla \phi_a \, dA \quad (3.40)$$

are minimized with respect to the remaining free coefficients $\hat{\mathbf{G}}^a$ for each $a = 1, \dots, n$.

Substituting (3.1) for ϕ_a , equation (3.40) leads to the quadratic functional

$$\hat{\mathbf{F}} = \mathbf{G}^T \hat{\mathbf{A}} \mathbf{G}, \quad (3.41)$$

where

$$\hat{A}_{ij} = \int_{\Omega} p_{i,k} p_{j,k} \, dA \quad (i, j = 1, \dots, n). \quad (3.42)$$

The components of $\hat{\mathbf{A}}$ are integrals of polynomials over the element interior, which may easily be evaluated by converting the domain integrals to boundary integrals with the divergence theorem. And, looking ahead to Subsection (3.5), the matrix $\hat{\mathbf{A}}$ is needed for the construction of the numerical quadrature rule on the element as well. Substituting equation (3.39) for \mathbf{G} in equation (3.41) and minimizing with respect to the free coefficients $\hat{\mathbf{G}}$ leads to

$$\hat{\mathbf{S}}^T \hat{\mathbf{A}} \hat{\mathbf{S}} \hat{\mathbf{G}} = -\hat{\mathbf{S}}^T \hat{\mathbf{A}} \hat{\mathbf{C}}. \quad (3.43)$$

This system of equations is guaranteed to be of full rank, therefore its solution completes the determination of the polynomial shape function coefficients:

$$\hat{\mathbf{G}} = -\left(\hat{\mathbf{S}}^T \hat{\mathbf{A}} \hat{\mathbf{S}}\right)^{-1} \hat{\mathbf{S}}^T \hat{\mathbf{A}} \hat{\mathbf{C}}. \quad (3.44)$$

After solving for the free coefficients $\hat{\mathbf{G}}$, all the polynomial coefficients are recovered using equation (3.36), so that

$$\mathbf{G} = \hat{\mathbf{S}} \hat{\mathbf{G}} + \hat{\mathbf{C}} \quad (3.45)$$

$$= \hat{\mathbf{S}} \left(\left(\hat{\mathbf{S}}^T \hat{\mathbf{A}} \hat{\mathbf{S}} \right)^{-1} \hat{\mathbf{S}}^T \hat{\mathbf{A}} \hat{\mathbf{C}} \right) + \hat{\mathbf{C}}. \quad (3.46)$$

Note that the form of the relationship between the shape function coefficients and the free coefficients given by equation (3.23) allows for a straightforward presentation of the imposition of constraints and minimizations. There are, however, computational inefficiencies

inherent in this representation. A more numerically efficient algorithm has been implemented in the VETFEM code, and is included in Appendix B.3.

3.5 Numerical integration rule

This section discusses numerical integration for the VETFEM method, and follows [64]. The procedure provides a means to compute integration weights and allows for efficient numerical evaluation of integrals over arbitrary polygon regions.

The element contributions to the global stiffness matrix, equation (2.26), must be evaluated by carrying out the indicated integrals over the element, using a numerical integration rule. In the present context of variable element topology, construction of such an integration rule presents a special problem that does not arise in conventional finite element methodology. The simple geometry of conventional finite elements in the parent element domain implicate locations for the integration points, but in the VETFEM no transformation to a simple, regular parent domain is available. Accordingly, the integration point locations must be chosen in relation to each individual element, and the associated weights then assigned in such a manner to render the overall integration rule “as accurate as possible.”

As discussed in [64], the integration point locations, and the assignment of weights, are treated separately in the VETFEM. Because there is no clear method to specify optimal integration point locations for arbitrary polygonal regions [64], an accuracy-based set of guidelines for the placement of the integration points is difficult to formulate. Therefore, the locations of the integration points are simply chosen to cover the element reasonably well. In particular, integration points are located at the element centroid, and at 70% of the distance from the centroid to each node. Having specified the integration point locations, weights must now be assigned.

Integration weights are sought such that an exact value of the integral is computed, when the integrand is equal to a certain reference function that is defined by the integration point values. In the case of a one-dimensional integral, this reference function may be taken to be

the unique polynomial of order $n-1$ that fits n integration point values. However, depending on the geometric arrangement of the integration points, no such unique polynomial fit may exist in higher dimensions. Accordingly, a somewhat more involved definition of the reference function must be made in relation to the given point distribution.

Consistent with the foregoing remarks, a polynomial function that is as smooth as possible, in a certain sense, is fit to the known values of the integrand at the integration points. This smooth function then serves as the reference function, and is integrated exactly on the element domain. The result is a linear combination of the integration point values, the coefficients of which are the weights. The smooth polynomial interpolant is obtained by choosing a larger number of monomials from Pascal's triangle than there are integration points, and then setting the coefficients of these monomials so that the function passes through the integrand data at the integration points, and the square-magnitude of the gradient is minimized on the element in an average sense. This second condition corresponds to the "as smooth as possible" stipulation.

Proceeding in this direction, an approximate value of the integral

$$J = \int_{\Omega} f(\mathbf{x}) \, dA \quad (3.47)$$

is sought, in which the integrand $f(\mathbf{x})$ is to be approximated by an interpolant $\bar{f}(\mathbf{x})$, and then exactly integrated. This interpolant must satisfy the two conditions

$$\bar{f}(\mathbf{x}_i) = f(\mathbf{x}_i), \quad i = 1, \dots, k \quad (3.48)$$

$$H = \int_{\Omega} \nabla \bar{f} \cdot \nabla \bar{f} \, dA \text{ is a minimum.} \quad (3.49)$$

Here, \mathbf{x}_i is the location of integration point i and k is the number of integration points.

The interpolant \bar{f} is taken to have the form of a polynomial

$$\bar{f} = \sum_{j=1}^l h_j p_j(\mathbf{x}), \quad (3.50)$$

where p_j is the j^{th} monomial appearing in Pascal's triangle, and the h_j are to be set so that conditions (3.48) and (3.49) are satisfied. The number of terms l in the expansion may be

taken as the total number of terms in Pascal's triangle, up through the row following the row in which term number k appears. (Recall that l should be greater than k , so that the reference function \bar{f} retains sufficient flexibility after satisfying condition (3.48) to allow for the minimization given by (3.49).)

The determination of the coefficients h_j can be viewed as a constrained minimization problem, and in fact is very similar to that of finding the polynomial shape function coefficients. Specifically, the constraints (3.48), together with (3.50), may be represented by

$$\mathbf{P}\mathbf{h} = \mathbf{f}, \quad P_{ij} \equiv p_j(\mathbf{x}_i), \quad (i = 1, \dots, k), \quad (j = 1, \dots, l) \quad (3.51)$$

where \mathbf{h} is the column vector of coefficients h_j , and \mathbf{f} is the column vector with components $f_i = f(\mathbf{x}_i)$. Factorizing equation (3.51) with the generalized reduction algorithm in Appendix B.1, leads to the following representation for the unknown vector \mathbf{h} ,

$$\mathbf{h} = \mathbf{B}\bar{\mathbf{h}} + \mathbf{b}. \quad (3.52)$$

where \mathbf{B} , \mathbf{b} are determined by the elimination algorithm and $\bar{\mathbf{h}}$ are free coefficients. To establish this rule independently from a specific set of nodal values \mathbf{f} , we replace \mathbf{B} and \mathbf{b} by their representations given in equations (B.6) and (B.12) of Appendix B.1, yielding the following partitioned matrix system:

$$\mathbf{C}^T \mathbf{h} = \begin{bmatrix} \mathbf{L} \mathbf{R} \mathbf{P} \mathbf{C} \\ \mathbf{I} \end{bmatrix} \bar{\mathbf{h}} + \begin{bmatrix} \mathbf{L} \mathbf{R} \mathbf{f} \\ \mathbf{0} \end{bmatrix}. \quad (3.53)$$

Where \mathbf{L} , \mathbf{R} , \mathbf{C} are elimination, row pivot, and column pivot matrices respectively, and $\mathbf{0}$ is a matrix whose entries are zeros. Note that with appropriate resizing of $\mathbf{0}$, equation (3.53) can be written as

$$\mathbf{h} = \begin{bmatrix} \mathbf{L} \mathbf{R} \mathbf{P} \mathbf{C} \\ \mathbf{I} \end{bmatrix} \bar{\mathbf{h}} + \begin{bmatrix} \mathbf{L} \mathbf{Q} \\ \mathbf{0} \end{bmatrix} \mathbf{f}. \quad (3.54)$$

Equation (3.54) demonstrates that matrix \mathbf{B} is a factorization of \mathbf{P} that is completely independent of the right-hand side vector \mathbf{f} , and that \mathbf{f} can be extracted from \mathbf{b} . Thus,

equation (3.52) can be written as

$$\mathbf{h} = \mathbf{B}\bar{\mathbf{h}} + \mathbf{M}\mathbf{f}, \quad (3.55)$$

for arbitrary data \mathbf{f} . In order to compute \mathbf{M} , we factorize the following system of equations which uses the identity matrix in place of \mathbf{f} :

$$\mathbf{P}\mathbf{X} = \mathbf{I}. \quad (3.56)$$

Factorization of equation (3.56) returns matrices \mathbf{B} and \mathbf{M} needed for construction of equation (3.55). Having thus determined the relationship between the bound and free coefficients necessary for satisfaction of (3.48), we set the free coefficients such that the reference function will be as smooth as possible, through satisfaction of (3.49).

The quadratic form in (3.49) may be written

$$H = \mathbf{h}^T \bar{\mathbf{A}} \mathbf{h}, \quad (3.57)$$

where the $(l \times l)$ matrix $\bar{\mathbf{A}}$ is as defined in (3.42), although possibly with a different dimension. Substituting for \mathbf{h} with equation (3.52), and minimizing H with respect to vector $\bar{\mathbf{h}}$ leads to

$$\bar{\mathbf{h}} = -(\mathbf{B}^T \bar{\mathbf{A}} \mathbf{B})^{-1} \mathbf{B}^T \bar{\mathbf{A}} \mathbf{M} \mathbf{f}. \quad (3.58)$$

Defining a column vector \mathbf{Q} of length l and with components

$$\mathbf{Q} \equiv \int_{\Omega} p_i \, dA, \quad (3.59)$$

equations (3.58) and (3.52) are combined with (3.47) to yield

$$\mathbf{J} \approx -\mathbf{w}^T \mathbf{f}, \quad \mathbf{w}^T \equiv \mathbf{Q}^T \left[\mathbf{1} - \mathbf{B} (\mathbf{B}^T \bar{\mathbf{A}} \mathbf{B})^{-1} \mathbf{B}^T \bar{\mathbf{A}} \right] \mathbf{M}. \quad (3.60)$$

In equation (3.60), \mathbf{w} is the $(k \times 1)$ column vector whose entries are the weights associated with the integration points. With reference to (3.59), it is remarked that these integrals of monomial terms are needed also to form the matrix $\bar{\mathbf{A}}$, and so do not represent a separate calculation.

3.5.1 Consistency constraints and numerical integration

An important relationship exists between the consistency constraints and numerical integration. The consistency constraints provide an equivalence between boundary and domain integrals of equation (3.8). However, implicit in this equivalence is the assumption that the integration is exact, and it can break down when integrations are performed in an inconsistent manner. For example, if consistency constraints are imposed using exact integration, they will not generally satisfy consistency when approximate integration is used to compute the boundary and domain residuals. Numerical tests have shown that shape functions constructed in the manner just described will not pass the patch test. Therefore, if a quadrature rule such as the one suggested in this section is to be used in computation of the global residual, then it should also be used to construct the consistency constraints.

3.6 Outline of the algorithm

In this section, a step-by-step procedure is given for the formation of both the shape function and the integration weights for each element. Formation of the force residual, and assembly of the global stiffness matrix, are not described in detail, as the VETFEM is identical to the conventional FEM in these respects.

For linear problems, the calculations outlined below are, of course, performed only once for each element. In the case of finite deformations, the shape function coefficients and integration weights must be recomputed on each load step only if an updated Lagrangian formulation is being used. In a total Lagrangian formulation, on the other hand, the reference configuration of the body is discretized, and therefore the geometry of the elements does not change from step to step. The shape functions and integration weights are therefore fixed for the entire analysis, and the computational effort associated with their determination becomes a truly negligible fraction of the overall expense of the analysis. Because the updated Lagrangian approach does not engender any clear advantage over the total Lagrangian formulation, the latter is perhaps to be preferred in the context of the

VETFEM.

After reading geometry and boundary-condition input, for each element:

1. Compute integrals of monomials over the boundary of the element. These integrals are needed to form matrices \mathbf{A} in (3.28), $\hat{\mathbf{A}}$ in (3.41) and $\bar{\mathbf{A}}$ in (3.57), as well as vector \mathbf{Q} in (3.59). $\bar{\mathbf{A}}$ and \mathbf{Q} involve domain integrals; however, these may be converted to boundary integrals using the divergence theorem. The integrals of all monomials through the required order may be efficiently evaluated by employing a recursive approach
2. Set and store the locations of the integration points, as described at the end of the previous subsection. This and all calculations should be performed using the element centroid as a local coordinate origin for the current element
3. Compute the weights for the integration points by performing steps (3.a) through (3.e) below. The number L of terms in the interpolant (3.50) is taken to be the total number of terms in Pascal's triangle, up through the row following the row in which term number K appears (recall that K is the number of integration points for the element)
 - (a) Fill the constraint matrix \mathbf{C} (see equation (3.51))
 - (b) Obtain the matrices \mathbf{B} and \mathbf{M} by invoking the reduction algorithm using matrix \mathbf{C} . The columns of \mathbf{M} are obtained by performing the back-substitution phase of the reduction procedure K times, taking the right-hand side column vector to be δ_{ij} , ($j = 1, \dots, K$) on back-substitution cycle number i
 - (c) Fill the \mathbf{Q} -vector (equation (3.59)) and the $\bar{\mathbf{A}}$ -matrix using the integrals of monomials computed in step 1
 - (d) Form $\mathbf{B}^T \bar{\mathbf{A}}$, then form $(\mathbf{B}^T \bar{\mathbf{A}} \mathbf{B})^{-1} \mathbf{B}^T \bar{\mathbf{A}}$
 - (e) Finally, form the column vector \mathbf{w} of weights as given in (3.60).

4. Set the number of terms m in the expansion for the polynomial shape functions (3.1). Experience has shown that entirely satisfactory shape functions are obtained when m is set so that the shape functions are complete polynomials through order $n - 1$, where n is the number of nodes for the element
5. Construct the constraint matrices \mathbf{A} and \mathbf{X} according to equation (3.22), and then invoke the reduction algorithm described in Appendix B.1 with \mathbf{A} and \mathbf{X} as input
6. Construct matrix $\tilde{\mathbf{A}}$ (see equations (3.28), (3.29)), and then form the linear system of equations (3.35). Perform linear reduction on equation (3.35) to obtain the representation of equation (3.39) for the polynomial coefficients
7. If free coefficients remain, then construct matrix $\hat{\mathbf{A}}$ (see equations (3.41), (3.42)), form the linear equations given by equation (3.43), and solve for the free coefficients $\hat{\mathbf{G}}$. Then recover the shape function polynomial coefficients using (3.45).

This procedure involves a number of instances of partial factorization of small systems of linear equations. Typically, on the order of ten linear equations (or less) are factored at a time. Experience has indicated that the computational effort associated with these calculations is only a small fraction of the total effort required, with the exception of small elastostatic analyses. In the case of a total Lagrangian formulation for a large-deformation problem, calculation of the shape function coefficients must be performed once for each element, instead of on every load step; the computational effort associated with constructing the shape functions is truly negligible fraction of the total solution effort in this case.

3.7 Example shape functions

This section presents example shape functions generated by the VETFEM for three specific elements. These examples provide some indication of the flexibility and robustness of the method. In addition, the examples demonstrate that the VETFEM shape functions satisfy the node point and consistency constraints, and form a partition of unity.

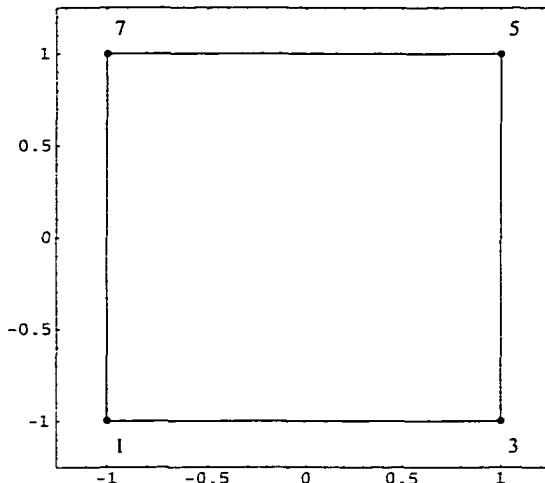


Figure 3.3: Square element with four nodes

3.7.1 Example 1: square element with 4 nodes

The first example involves the square element shown in Figure 3.3. This element has nodes at the geometric corners, and corresponds to the parent element of a standard bilinear quadrilateral. This example demonstrates that the VETFEM shape function generation procedure chooses functions with exact linear variation, when they are available in the approximating polynomial. In this particular case, the VETFEM recovers the standard bilinear quadrilateral shape functions.

The element nodal coordinates are given in Table 3.1. The shape function coefficients generated by the VETFEM procedure are listed in Table 3.2 and the shape functions are plotted in Figures 3.4 and 3.5.

Note that the coefficients are consistent with the finite element bilinear shape functions to machine precision. In this case, exact linear variation was achieved and all coefficients were determined during the compatibility optimization. Tests using higher-order approximations were also performed for this element configuration. In those cases, because of the presence of bubble functions, the smoothness minimization was invoked and the same

shape functions were recovered. Because these functions are conforming, an approximation utilizing them is consistent.

Node	<i>x</i>	<i>y</i>
1	-1.0	-1.0
2	1.0	-1.0
3	1.0	1.0
4	-1.0	1.0

Table 3.1: Nodal coordinates for square element with four nodes

	SF1	SF2	SF3	SF4
1	0.25000E+00	0.25000E+00	0.2500000E+00	0.2500000E+00
<i>x</i>	-0.25000E+00	0.25000E+00	0.2500000E+00	-0.2500000E+00
<i>y</i>	-0.25000E+00	-0.25000E+00	0.2500000E+00	0.2500000E+00
<i>x</i> ²	-0.55511E-16	0.27755E-16	0.8326673E-16	-0.5551115E-16
<i>xy</i>	0.25000E+00	-0.25000E+00	0.2500000E+00	-0.2500000E+00
<i>y</i> ²	-0.41633E-16	-0.41633E-16	0.4163336E-16	0.4163336E-16
<i>x</i> ³	0.14745E-15	-0.14745E-15	-0.1301043E-15	0.1301043E-15
<i>x</i> ² <i>y</i>	-0.62450E-16	0.62450E-16	-0.6245005E-16	0.6245005E-16
<i>xy</i> ²	0.20816E-16	-0.20816E-16	0.2081668E-16	-0.2081668E-16
<i>y</i> ³	0.14745E-15	0.13010E-15	-0.1301043E-15	-0.1474515E-15

Table 3.2: Shape function coefficient values for a square element with four nodes

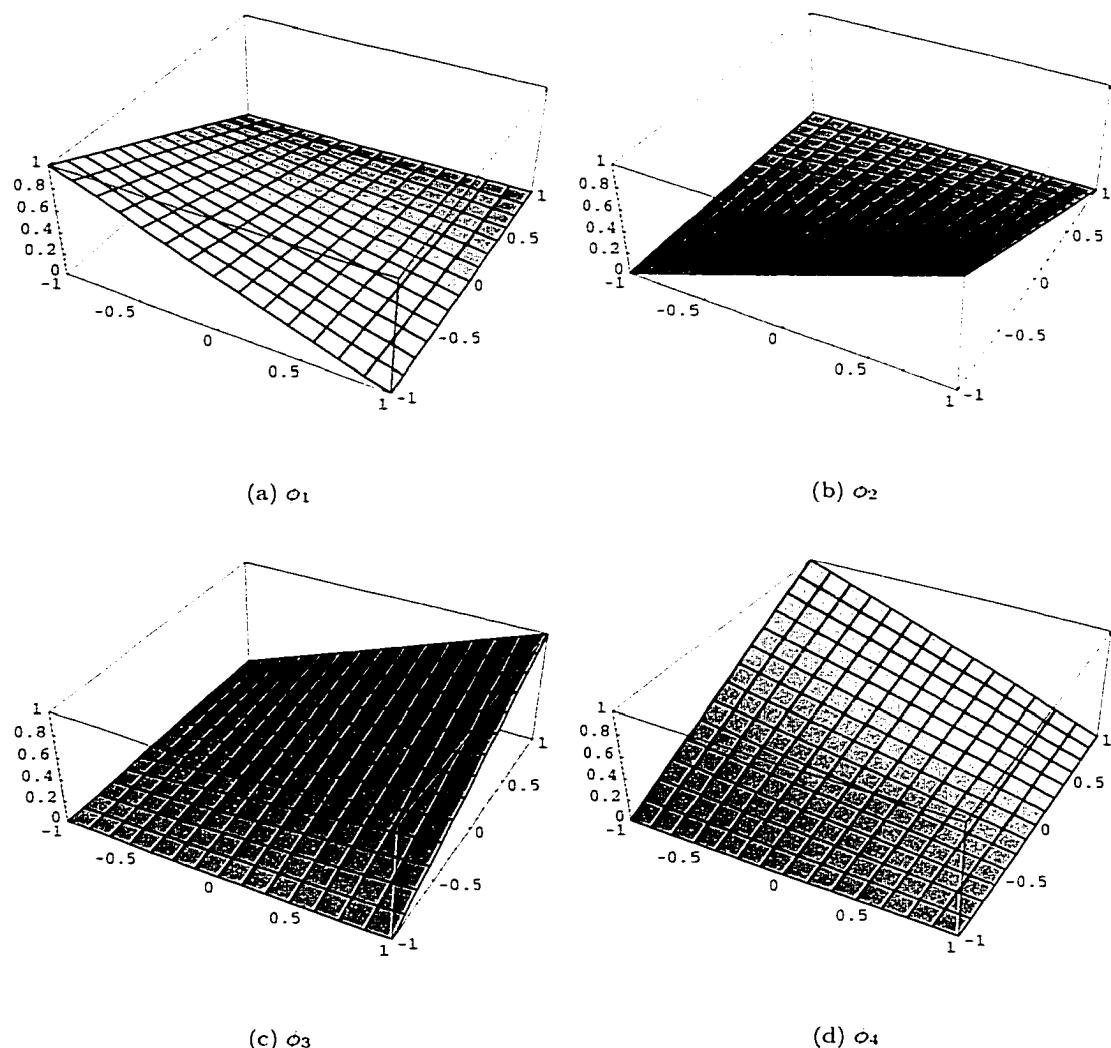


Figure 3.4: Shape functions for a rectangular element with four nodes

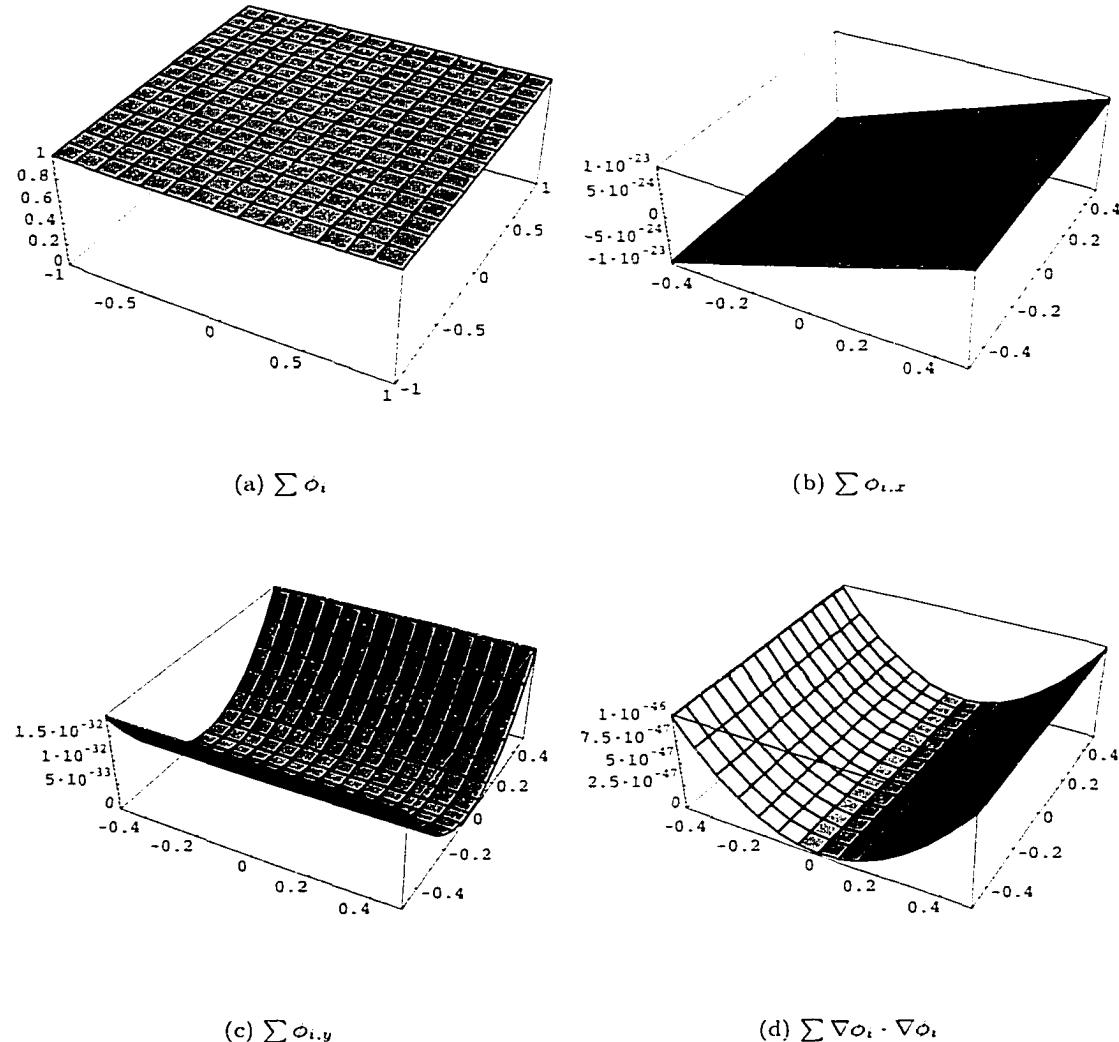


Figure 3.5: The sum of the nodal shape functions is shown in (a). This clearly demonstrates that the shape functions form a partition of unity. Subfigures (b) and (c) plot components of the gradient of the shape function sum. (d) is a plot of the smoothness functional of the shape function sum. Ideally these quantities are zero, and these results show that numerically determined shape functions provide numerically exact solutions

3.7.2 Example 2: square element with 7 nodes

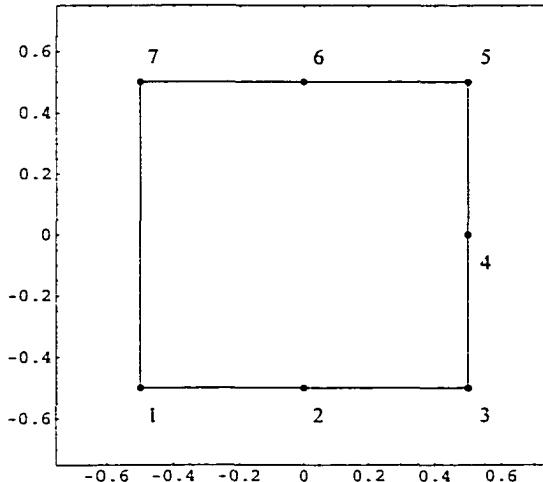


Figure 3.6: Square element with seven nodes

For this example, the element is geometrically square, but contains 7 nodes. The element is centered at (0,0) and contains nodes at the geometric corners as well as at the mid-sides on three edges. The element nodal coordinates are given in Table 3.3. This example illustrates the robustness of the procedure.

Note that exact, linear variation cannot be achieved by finite-order polynomial shape functions along edges in which the shape function is non-zero and the edge contains a mid-side node. This makes the presence of discontinuities likely because, for example, a square element with exact linear variation may adjoin the element in question similar to that depicted in Figure 2.3 on page 21.

The particular arrangement shown in Figure 3.6 proved troublesome in early implementations of the VETFEM, because the space of polynomials of order > 3 contains infinitely many solutions satisfying the continuity optimization. Specifically, note that such polynomials contain bubble functions of the form $c_1(x - 0.5)(x + 0.5) \times c_2(y - 0.5)(y + 0.5)$, that do not contribute to the compatibility functional. The uniqueness of the solution therefore

Node	<i>x</i>	<i>y</i>
1	-0.50	-0.50
2	0.00	-0.50
3	0.50	-0.50
4	0.50	0.00
5	0.50	0.50
6	0.00	0.50
7	-0.50	0.50

Table 3.3: Nodal coordinates for square element with seven nodes

depends on the smoothness optimization which was, in fact, added to the method after observing poor behavior in problems of this kind.

For this problem, all three stages were necessary for determination of the shape function coefficients. The polynomial order of the shape function was chosen to be 6. The total number of coefficients to be determined was 28, and Table 3.4 lists each stage and the number of coefficients eliminated.

stage	bound	free
1	14	14
2	23	5
3	28	0

Table 3.4: VETFEM stages and the number of bound and free coefficients

For this element, the shape function coefficients are listed in Table 3.5. Note that the last column sums the shape function coefficients, and that this sum equals one. This is equivalent to setting the nodal coefficients equal to 1 and expressing the function interpolated over the element. The result is that the constant function is recovered exactly. This fact demonstrates that the method yields functions that form a partition of unity. Additionally, the given functions satisfy the reproducing conditions to numerical precision as

well: that is, if the nodal values are set such that $\mathbf{u}_i = \mathbf{x}_i$, the displacement field recovered is $\mathbf{u}_h = \mathbf{x}$ exactly. Together, these findings demonstrate that the VETFEM shape functions are complete for this (and, in fact, for all) element geometry.

monomial	SF1	SF2	SF3	SF4	SF5	SF6	SF7	\sum
1	0.142854	0.142856	0.142858	0.142860	0.142860	0.142858	0.142856	1.0000E+00
x	0.000000	-0.223380	-0.278551	-0.123967	0.123967	0.278551	0.223380	-3.8550E-13
y	0.285714	0.178140	-0.063577	-0.257420	-0.257420	-0.063577	0.178140	-1.2000E-07
x^2	-0.331566	0.073788	0.298736	-0.206741	-0.206741	0.298736	0.073788	2.0000E-08
xy	0.000000	-0.646522	0.287730	0.518471	-0.518471	-0.287730	0.646522	6.7023E-12
y^2	0.331583	-0.073777	-0.298740	0.206725	0.206725	-0.298740	-0.073777	-2.0000E-08
x^3	0.000000	0.024798	-0.044685	0.055721	-0.055721	0.044685	-0.024798	7.9041E-13
x^2y	-0.171462	0.154482	-0.106905	0.038154	0.038154	-0.106905	0.154482	-6.0000E-08
xy^2	0.000000	-0.074395	0.134054	-0.167163	0.167163	-0.134054	0.074395	5.7211E-13
y^3	0.057154	-0.051494	0.035635	-0.012718	-0.012718	0.035635	-0.051494	0.0000E+00
x^4	0.297284	-0.071880	-0.254998	0.178236	0.178236	-0.254998	-0.071880	2.0000E-08
x^3y	0.000000	0.548610	-0.224300	-0.484563	0.484563	0.224300	-0.548610	-1.4083E-11
x^2y^2	-0.050605	0.045564	-0.031535	0.011274	0.011274	-0.031535	0.045564	0.0000E+00
xy^3	0.000000	0.577874	-0.277032	-0.418807	0.418807	0.277032	-0.577874	-1.4671E-11
y^4	-0.280442	0.056676	0.265515	-0.181970	-0.181970	0.265515	0.056676	-6.0000E-08
x^5	0.000000	0.082548	-0.112398	0.103810	-0.103810	0.112398	-0.082548	-4.2476E-13
x^4y	-0.248158	0.316791	-0.364160	0.171449	0.171449	-0.364160	0.316791	0.0000E+00
x^3y^2	0.000000	-0.379399	0.320170	-0.035761	0.035761	-0.320170	0.379399	-4.3654E-13
x^2y^3	-0.531796	0.292716	0.087303	-0.114121	-0.114121	0.087303	0.292716	-4.0000E-08
xy^4	0.000000	-0.033341	0.241820	-0.483287	0.483287	-0.241820	0.033341	-2.2114E-13
y^5	0.155991	-0.121901	0.055371	-0.011466	-0.011466	0.055371	-0.121901	-4.0000E-08
x^6	-0.123907	-0.034581	0.251303	-0.154768	-0.154768	0.251303	-0.034581	-8.0000E-08
x^5y	0.000000	-0.579247	0.473538	-0.020263	0.020263	-0.473538	0.579247	7.3336E-12
x^4y^2	-0.673593	0.460679	-0.091426	-0.032456	-0.032456	-0.091426	0.460679	1.2000E-07
x^3y^3	0.000000	-0.840491	0.374054	0.674021	-0.674021	-0.374054	0.840491	1.5241E-11
x^2y^4	-0.242540	0.364761	-0.479791	0.236301	0.236301	-0.479791	0.364761	0.0000E+00
xy^5	0.000000	-0.261244	-0.099485	0.694284	-0.694284	0.099485	0.261244	7.9972E-12
y^6	0.307145	-0.130500	-0.137062	0.113989	0.113989	-0.137062	-0.130500	-1.0000E-07

Table 3.5: Shape function coefficient values for a square element with seven nodes

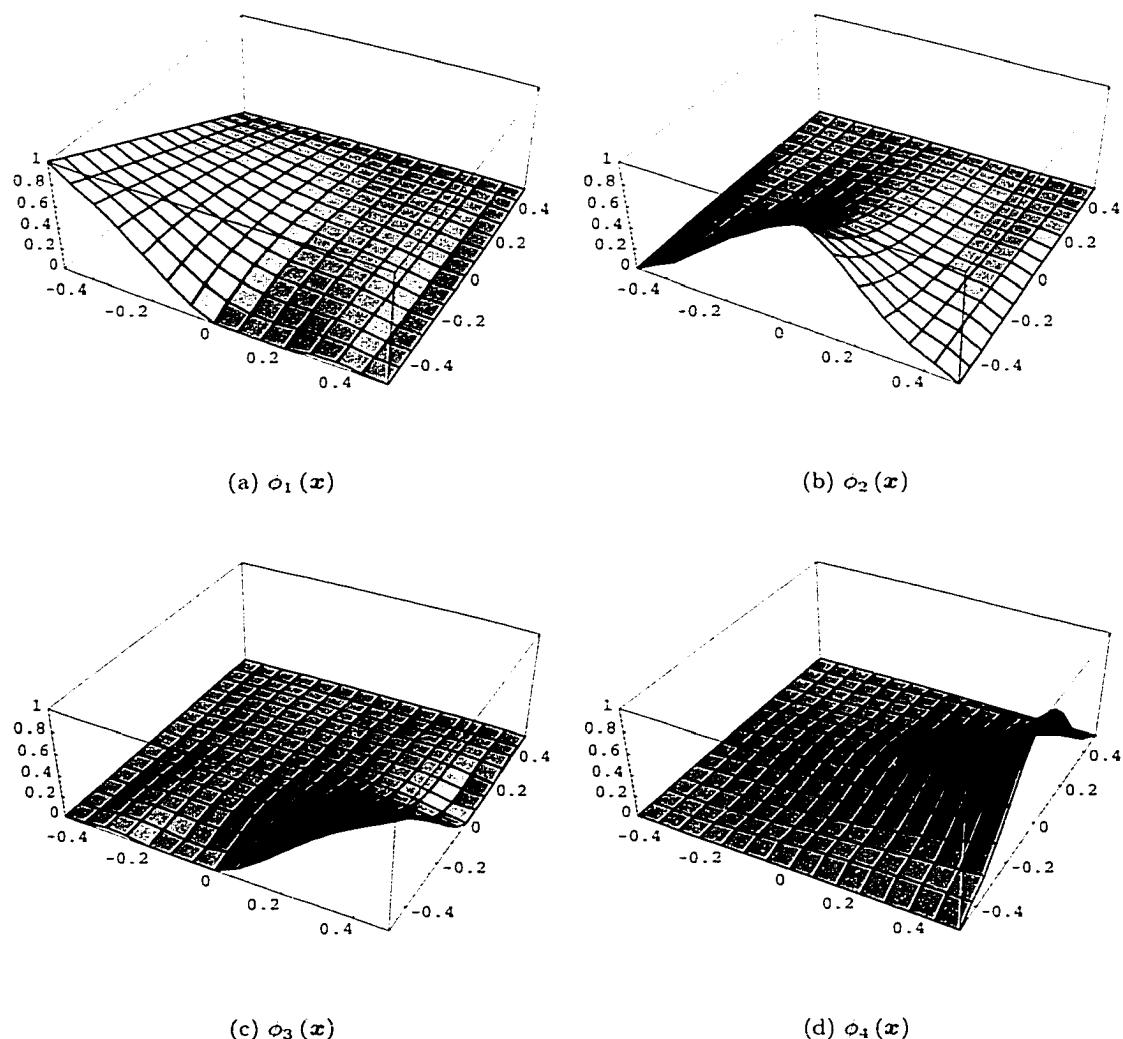


Figure 3.7: Shape functions 1 – 4 for a rectangular element with seven nodes

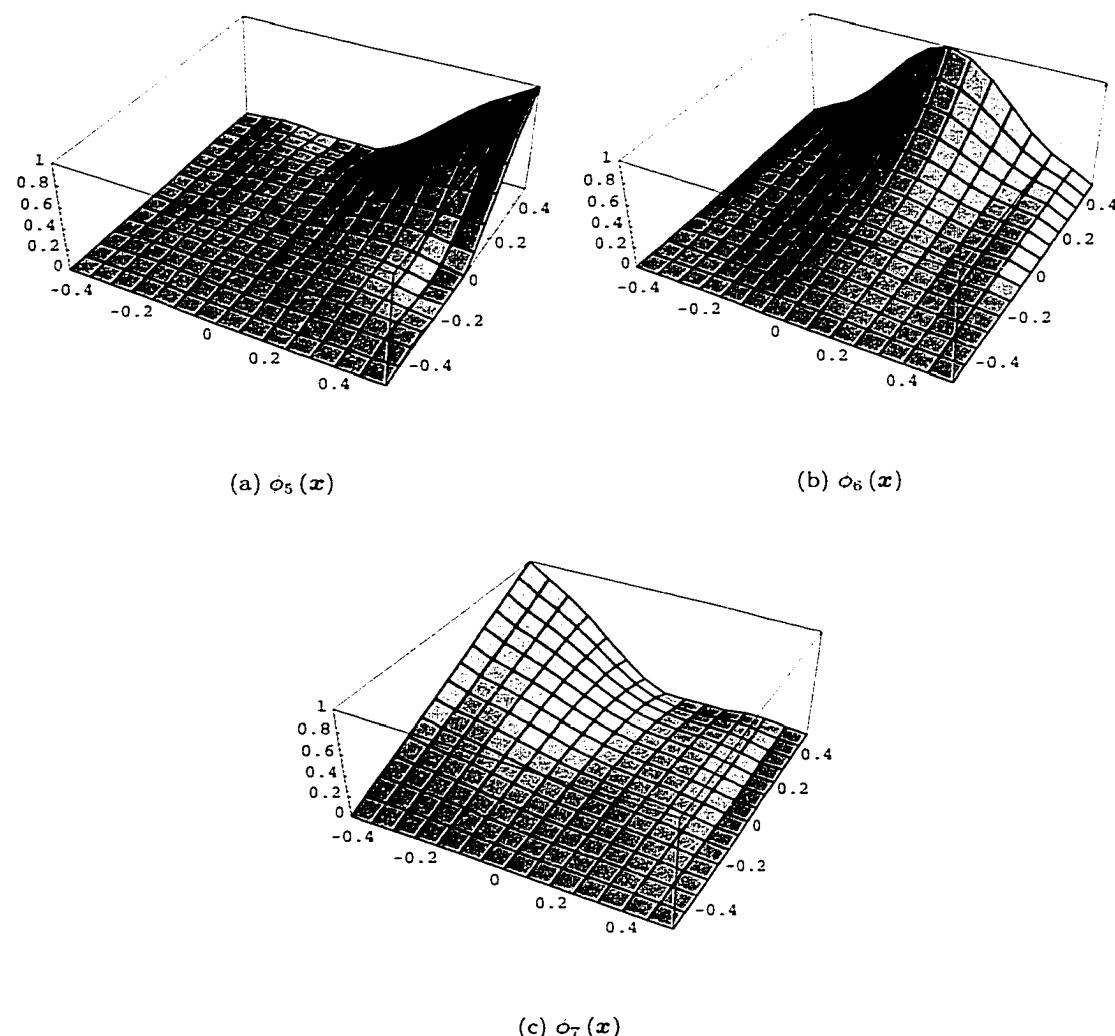


Figure 3.8: Shape functions 5 – 7 for a rectangular element with with seven nodes

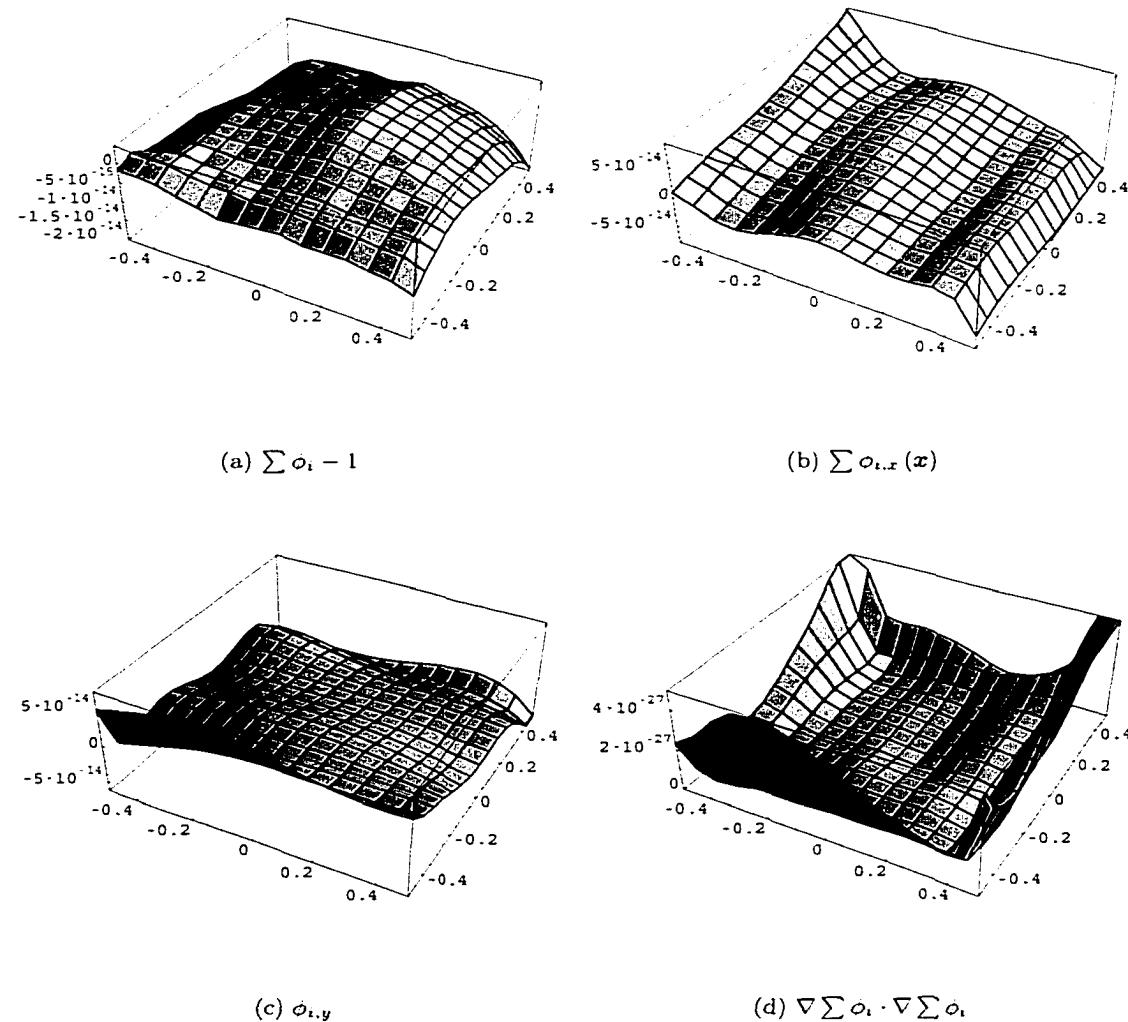


Figure 3.9: The sum of the nodal shape functions minus one is shown in (a). The deviation from unity is on the order of double precision resolution, demonstrating that the shape functions form a partition of unity. Subfigures (b) and (c) plot components of the gradient of the shape function sum. (d) is a plot of the smoothness functional of the shape function sum. The ideal values of these quantities is zero, and these figures show that numerically determined shape functions provide numerically exact solutions

3.7.3 Example 3: concave element

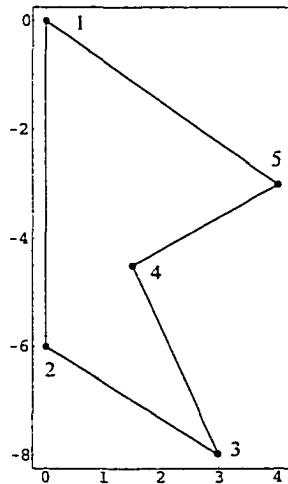


Figure 3.10: Concave element with five nodes

In this example, an element having a significant concavity is considered. The geometry of the element is shown in Figure 3.10. The element has five nodes located at its vertices. The concavity is generated by the location of node 4. The node positions are given in Table 3.6.

Shape function coefficients generated by the VETFEM are listed in Table 3.7, and the resulting functions are plotted in Figures 3.12, 3.14, 3.13 and 3.15. As a visual aid, the shape functions are plotted along with an element box which is composed of the element edges which shown at heights of zero and one. In addition, the far sides of this element box are shown as solid walls, making it possible to see the shape function variation along those edges. The shape functions may be seen to form a partition of unity from Figure 3.15(a) as well as from Table 3.7. Additionally, the functions numerically satisfy the reproducing conditions.

Note that although the construction procedure produces shape functions for concave elements, concavities tends to diminish the influence of distant nodes on a given shape

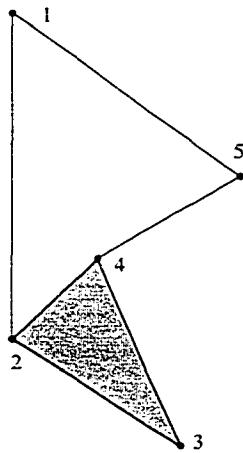


Figure 3.11: Shape function ϕ_3 is nearly zero outside the shaded region

function. For example, shape function ϕ_3 , shown in Figure 3.13, takes the value of one at node 3, and varies approximately linearly to zero at nodes 2 and 4. On the remaining edges, the function approximates the zero function. In addition, due to the concavity at node 4, the function is nearly zero everywhere within the element that lies outside the triangle formed by nodes 2 - 3 - 4, shown as the shaded region in Figure 3.11. Therefore, though the method is able to form functions even for elements with significant concavities, it is perhaps best to use elements that are nearly convex, in order to maximize nodal shape function influence throughout the domain.

As an example of the polynomial form of shape functions, ϕ_1 is given in equation (3.61) below:

$$\begin{aligned}
 \phi_1 = & 0.0647068 - 0.12837x + 0.0722819x^2 - 0.00761868x^3 - \\
 & 0.00203462x^4 + 0.0878375y - 0.0231073xy + 0.0178319x^2y \\
 & - 0.010687x^3y + 0.0325194y^2 + 0.0219927xy^2 - 0.00275322x^2y^2 \\
 & + 0.00502437y^3 + 0.00516659xy^3 + 0.000459756y^4 \quad (3.61)
 \end{aligned}$$

Node	<i>x</i>	<i>y</i>
1	0.0	0.0
2	0.0	-6.0
3	3.0	-8.0
4	1.5	-4.5
5	4.0	-3.0

Table 3.6: Nodal coordinates locations for concave element

monomial	SF1	SF2	SF3	SF4	SF5	\sum
1	6.47E-02	-1.02E-02	-1.18E-01	1.02E+00	4.33E-02	1.00E+00
<i>x</i>	-1.28E-01	-1.81E-01	3.50E-03	9.36E-02	2.12E-01	-1.03E-15
<i>y</i>	8.78E-02	-4.00E-02	-1.14E-01	-3.09E-02	9.71E-02	-1.80E-15
<i>x</i> ²	7.23E-02	1.71E-01	6.54E-02	-4.16E-01	1.07E-01	7.77E-16
<i>xy</i>	-2.31E-02	8.89E-02	-4.05E-02	-8.91E-02	6.38E-02	-4.30E-15
<i>y</i> ²	3.25E-02	1.77E-02	3.75E-02	-9.54E-02	7.66E-03	-2.01E-15
<i>x</i> ³	-7.62E-03	-2.83E-02	-5.49E-03	5.97E-02	-1.83E-02	4.65E-16
<i>x</i> ² <i>y</i>	1.78E-02	-6.02E-03	2.27E-02	-2.80E-02	-6.54E-03	6.73E-16
<i>xy</i> ²	2.20E-02	-2.52E-03	2.73E-02	-4.21E-02	-4.72E-03	7.84E-16
<i>y</i> ³	5.02E-03	-3.69E-03	6.67E-03	-4.81E-03	-3.20E-03	3.80E-16
<i>x</i> ⁴	-2.03E-03	-2.33E-04	-2.47E-03	4.61E-03	1.18E-04	-2.11E-16
<i>x</i> ³ <i>y</i>	-1.07E-02	-1.81E-02	-1.06E-02	5.04E-02	-1.09E-02	0.00E+00
<i>x</i> ² <i>y</i> ²	-2.75E-03	-1.35E-02	-1.54E-03	2.66E-02	-8.81E-03	8.90E-16
<i>xy</i> ³	5.17E-03	2.20E-03	6.04E-03	-1.42E-02	7.95E-04	9.30E-16
<i>y</i> ⁴	4.60E-04	1.28E-03	3.90E-04	-2.93E-03	8.08E-04	2.69E-16

Table 3.7: Shape function coefficient values for a concave element

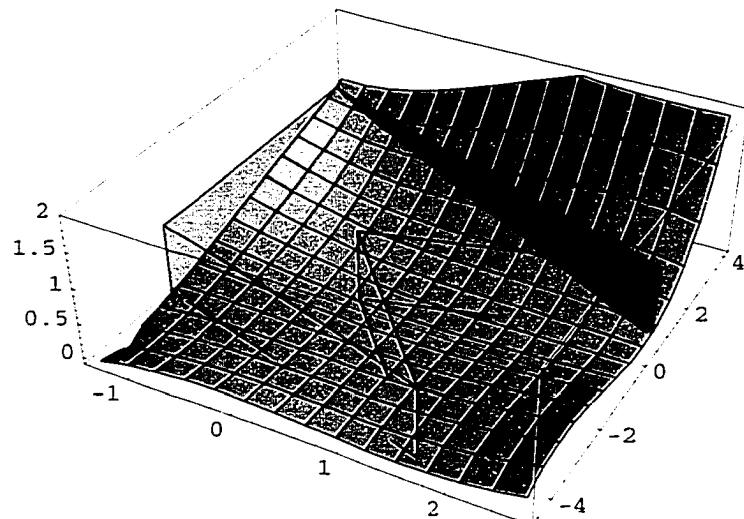
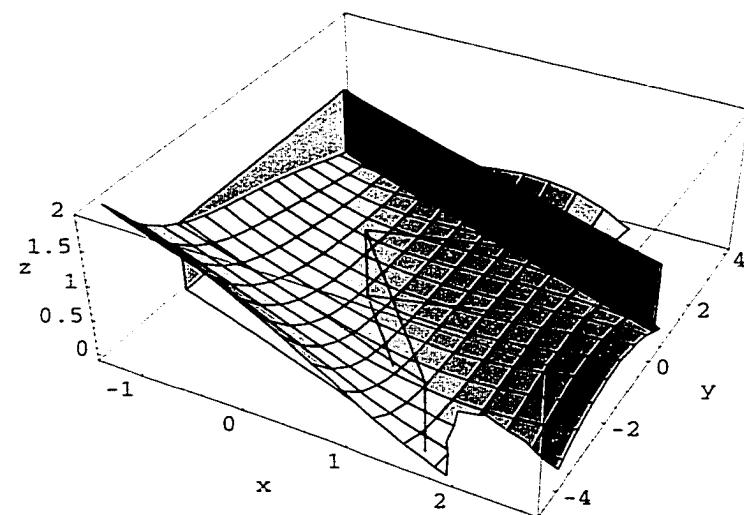
(a) $\phi_1(\mathbf{x})$ (b) $\phi_2(\mathbf{x})$

Figure 3.12: Shape functions 1 – 2 for a concave element with five nodes

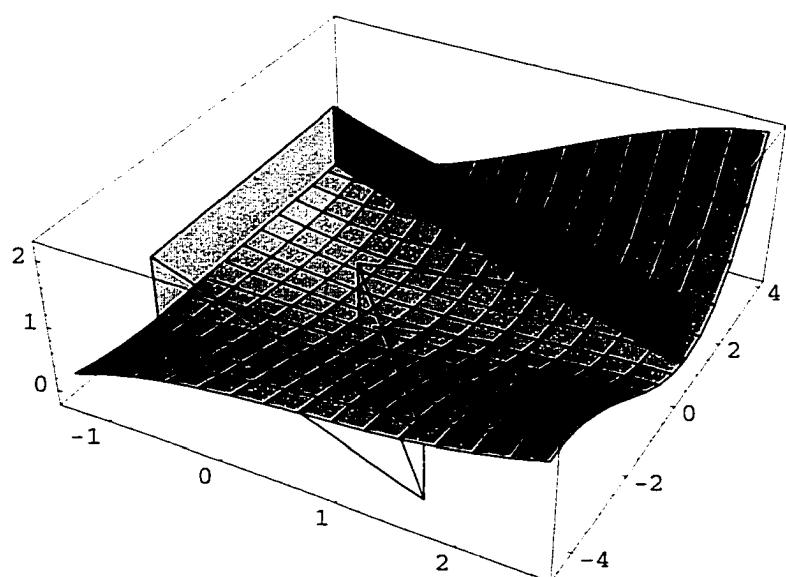


Figure 3.13: Shape function 3 for a concave element with five nodes

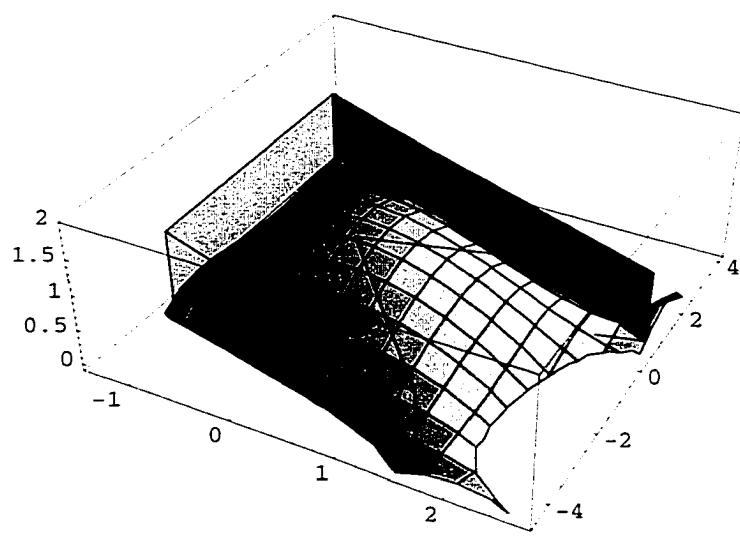
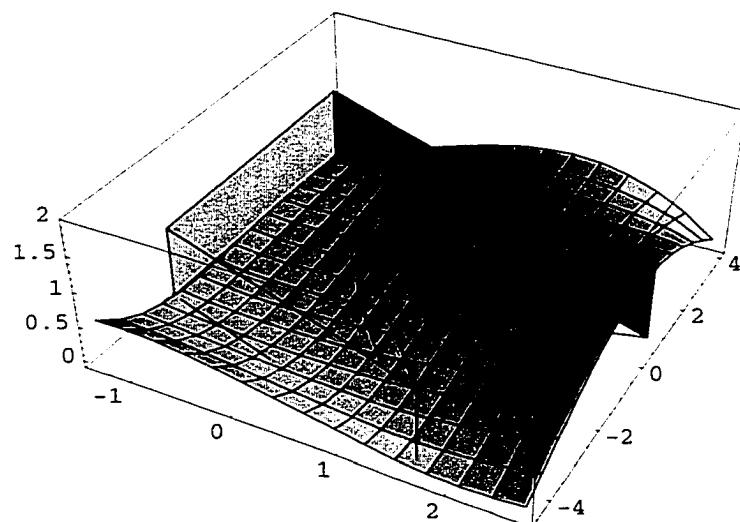
(a) $\phi_4(\mathbf{x})$ (b) $\phi_5(\mathbf{x})$

Figure 3.14: Shape functions 4 – 5 for a concave element with five nodes

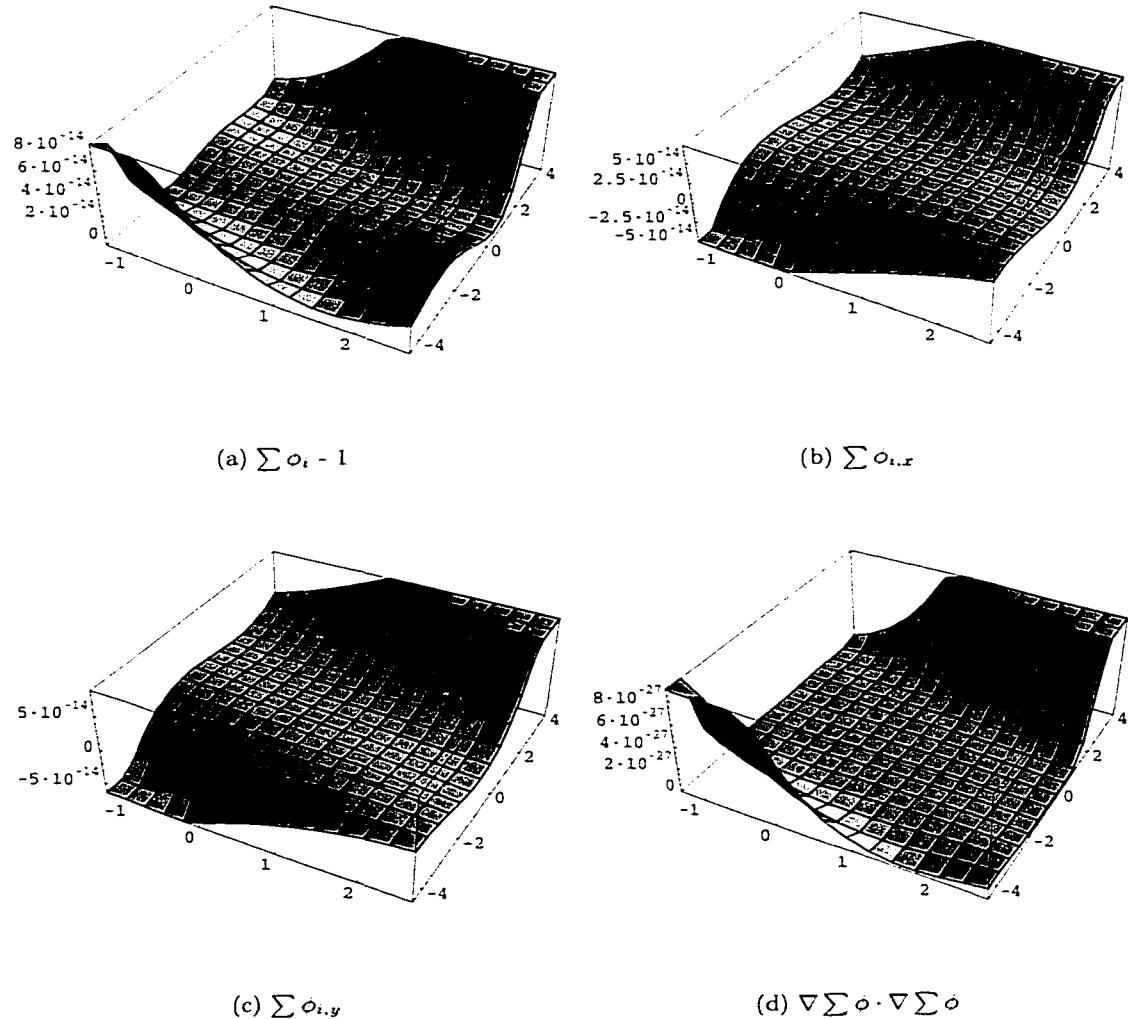


Figure 3.15: The sum of the nodal shape functions minus one is shown in (a). As in the previous example, the deviation from unity is on the order of numerical (double precision) resolution, demonstrating that the shape functions form a partition of unity. Subfigures (b) and (c) plot components of the gradient of the shape function sum. (d) is a plot of the smoothness functional of the shape function sum. The ideal values of these quantities is zero, and these figures show that numerically determined shape functions provide numerically exact solutions

3.8 Closing remarks

The VETFEM procedure provides an effective means of generating polynomial basis functions for geometric regions of very general shape. These functions satisfy the Kronecker delta property, have compact support, and are definable for regions that include geometric concavities. Shape-functions generated by this method may be used in the same manner as are those associated with conventional finite elements. In fact, a mesh need not consist entirely of elements of the type described here; VETFEM and conventional isoparametric elements may occur in any convenient pattern to form a finite element model.

Arguments for conditions related to convergence were presented; however, the convergence properties of the method must be carefully established, and this is the subject of Chapter 4.

Chapter 4

VETFEM convergence

4.1 Nonconforming methods

The VETFEM basis functions are composed of element-based, nodal shape functions that are constructed entirely in terms of the element geometry. The construction procedure used to generate the shape functions does not assure that they can be combined to form piecewise continuous basis functions. In fact, basis functions comprised of VETFEM shape functions are likely to exhibit small jump discontinuities along element boundaries. These discontinuities generate delta functions in basis function derivatives. Thus, the bilinear form which contains square integrals of basis-function derivatives is not well defined. Stated in functional analysis terms, this means that the basis functions do not belong to the space that contains the exact solution. While this fact does not mean that the basis cannot yield a satisfactory approximation, it has a profound effect on questions of convergence. When used as a basis for a finite element approximation, such functions are termed “nonconforming”, and the approximation is called “external”. The discussion in this chapter is limited to elliptic boundary value problems.

Nonconforming functions are frequently implemented in the finite element method for cases in which they provide a better approximation than a conforming basis of similar polynomial order. Or, the nonconforming basis may be significantly easier to generate.

Fully-integrated, low-order conforming elements are well known to provide poor results for problems such as beam bending, and for problems involving incompressible materials such as Stokes problem for an incompressible viscous fluid or standard plasticity models. In the case of pure bending of a 2-D beam, the linearity of the basis functions causes the deformation to be accompanied by shear deformation. This generates fictitious or parasitic shear strains and results in very poor approximations. While this difficulty may be resolved by reduced integration, or by increasing the polynomial order of the shape-function interpolants, it can also be addressed with nonconforming elements. The Wilson element, for example, introduces extra internal degrees of freedom through quadratic functions [38]. These additional shape functions enable the element to “bend,” eliminating the fictitious shear and greatly improving the approximation accuracy with minimal change in computational effort.

For problems involving nearly incompressible material behavior, the divergence of the displacement field is zero (within the context of the small deformation theory). This constraint is problematic for low-order elements because the only functions in the global approximation space with a zero divergence correspond to zero strain. Here too, the problem may be addressed by higher-order functions or selective reduced integration [37, 49]. However, the convergence tends to be sub-optimal. On the other hand, nonconforming elements can be used to enhance approximations. For the case of fully incompressible materials, hybrid elements are generally used, however, some of these approaches have been shown to be equivalent to nonconforming elements [11, 23, 30].

For modeling fourth-order differential equations, nonconforming elements simplify the construction of element shape functions, and often provide better results than does a continuous basis. The difficulty in specifying basis functions for these problems is related to the requirement that they possess piecewise-continuous first derivatives. On the other hand, it is relatively easy to define shape functions that provide C^0 continuity along element edges[80]. The nonconforming functions tend to be significantly more flexible than

the conforming ones and provide more accurate answers [44, 80].

The VETFEM uses nonconforming shape functions for the simple reason that, in the case of arbitrary polygonal elements, nonconforming functions are much easier to specify than conforming ones. The VETFEM shape functions are defined in a straightforward and efficient manner for each polygonal element in turn, without reference to the behavior of any other shape functions. This procedure, however, assures strict continuity of the nodal basis functions only at the nodes and not along element facets. On the other hand, a procedure that provides pointwise compatibility of shape functions on the boundaries of arbitrary polygonal regions would be much more difficult to conceive. It would require coupling the generation of shape functions, and would likely impose restrictions on the element topologies.

Perhaps the primary consequence of using nonconforming functions in a finite element approximation is that the general results related to convergence of Galerkin approximations do not apply, and therefore convergence evaluations must be linked to specific boundary value problems. A convergence evaluation of a finite element approximation to an elliptic BVP is often broken into three parts: the elliptic stability condition, consistency, and completeness. For conforming methods, elliptic stability is inherited from the variational boundary value problem (VBVP) – if the VBVP is elliptic on the entire solution space, then it will be elliptic on the subspace spanned by the approximation basis [57, 68]. Consistency in the context of the FEM does not appear to be precisely defined; however, a general analogy may be drawn to consistency in the finite difference setting. In finite difference methods, consistency may be considered broadly as a measure of the size of the approximation error. In that sense, the error in a Galerkin approximation is measured in the energy norm and Céa's lemma provides a general error bound. Céa's lemma reveals that the standard Galerkin approximation is optimal on the given approximation space; i.e. it minimizes the energy norm. Completeness in both the conforming and nonconforming methods is related to the highest-order complete polynomial contained in the approxima-

tion space. Thus, for a conforming method, because the approximation is guaranteed to be optimal, convergence for a stable BVP depends entirely on completeness of the approximation space. This is summarized in the so called approximability condition: a Galerkin approximation to a BVP of order $2m$ will converge to the exact solution, provided that the finite element approximation space contains all piecewise polynomials of order m . In the case of nonconforming elements, completeness by itself is does not provide a sufficient condition for convergence.

Convergence evaluations for nonconforming approximations typically yield very few general results, and the elliptic stability and consistency conditions must be specifically addressed. Ellipticity is no longer inherited from the boundary valued problem because the nonconforming approximation space is not contained in the solution space. Therefore, the method must be shown to satisfy the so-called stability condition for each boundary value problem for which the nonconforming functions are used. Consistency in the sense previously alluded to takes a central role in a convergence evaluation, and many tests have been devised to establish consistency. Completeness of a nonconforming method is precisely the same as for the conforming variety, and is a general property of the basis functions.

Because the VETFEM is a nonconforming method, the first consideration with regard to its convergence is ellipticity of the VBVP on the VETFEM approximation space. This evaluation is, of course, related to the underlying BVP, and here we are primarily concerned with the equations of elasticity. The analysis of this boundary value problem itself is nontrivial, and is established through Korn's inequalities. To establish the stability of the VBVP on the nonconforming function space requires that a discrete version of the second Korn's inequality be shown to hold for the nonconforming method [11, 30]. This proof is not included in this dissertation. Instead, convergence of the VETFEM is analyzed in parallel by two means. First, it is shown analytically to be stable for the simpler, but related Poisson equation. At the same time, the numerical implementation of the VETFEM for the linear elastic VBVP is tested empirically for satisfaction of stability (and consistency).

Convergence assessments are typically performed as numerical tests and less commonly as analytic evaluations. Numerical tests, such as the patch test, are designed to establish completeness and convergence of nonconforming methods in a simple straightforward fashion, and provide results that are entirely satisfactory for specific nonconforming applications. Alternatively, a more general analytic evaluation may be performed that yields more definite and general conclusions regarding a specific shape function formulation. These tests, however, involve significant background in functional analysis and are often hard to perform. In this dissertation, both approaches are adopted in evaluation of the VETFEM. An analytic evaluation is performed for the VETFEM-based approximation to Poisson's equation, with Dirichlet boundary conditions. This problem is chosen because it can be considered as a special case of the more general elasticity equations, and in this simplified context, the elliptic stability may be readily demonstrated. Secondly, the VETFEM is numerically evaluated using the classical patch test in the context of two-dimensional linear elasticity.

The analytic evaluations provided in this dissertation are intended to be correct and meaningful convergence evaluations, but they are not intended to constitute rigorous mathematical proofs. Such evaluations are left in the hands of more competent mathematicians.

4.2 Convergence analysis of the VETFEM

As mentioned previously, convergence evaluations include elliptic stability, consistency and completeness. As previously indicated, to arrive at analytic results that shed light on the fundamental aspects of convergence we first evaluate the VETFEM using the Poisson equation as a model boundary value problem.

Model problem The Poisson equation is a simple second-order boundary value problem. The strong form of the BVP equation is given by:

$$-\Delta u = f \text{ in } \Omega \in \mathbb{R}^2 \quad (4.1a)$$

$$u = \bar{u} \text{ on } \partial\Omega. \quad (4.1b)$$

In equation (4.1), u is a scalar-valued function of x , Δ is the Laplacian, and the domain, Ω , is assumed to be a convex polygonal domain. To formulate the weak or variational form equivalent to equation (4.1), we first multiply (4.1a) by a test function v which vanishes on the domain boundary $\partial\Omega$, integrate the residual over the problem domain, and apply the divergence theorem. This process yields

$$\int_{\Omega} \nabla u \cdot \nabla v \, da = \int_{\Omega} f v \, da. \quad (4.2)$$

In considering approximate solutions to equation (4.2), we note the following: the bilinear form on the left-hand side involves integrals of gradients, and therefore the trial solutions u and test functions v must have integrable first derivatives. Further, for the case that $u = v$ as in the Ritz method, the functions must have square-integrable first derivatives. These properties, along with the implied behavior of the functions u and v on the boundary, are summarized in the following function spaces. First, we define the space of functions that are square-integrable on the domain, denoted $L^2(\Omega)$ and given by

$$L^2(\Omega) = \left\{ u : \int_{\Omega} |u|^2 \, da < \infty \right\}, \quad (4.3)$$

to which we add the Sobolev spaces, which possess (generalized) derivatives that are members of the $L^2(\Omega)$ function space:

$$H^1(\Omega) = \{ \partial^\alpha u \in L^2(\Omega), |\alpha| \leq 1 \} \quad (4.4)$$

$$H_0^1(\Omega) = \{ u \in H^1(\Omega), u|_{\partial\Omega} = 0 \} \quad (4.5)$$

$$H_e^1(\Omega) = \{ u \in H^1(\Omega), u|_{\partial\Omega} = \bar{u} \}. \quad (4.6)$$

The superscript indicates the number of derivatives that are in $L^2(\Omega)$, and the subscript e indicates satisfaction of the essential boundary conditions. These spaces define the collection

of functions from which candidate solutions to the VBVP are to be selected. With these definitions in hand, the weak form of equation (4.1) is expressed as: find $u \in H_e^1(\Omega) \equiv V$, such that

$$a(u, v) = l(v) \quad \text{for all } v \in H_0^1 \equiv V_0, \quad (4.7)$$

where the bilinear form $a(u, v)$ is given by the left-hand side of equation (4.2):

$$a(u, v) = \int_{\Omega} \nabla u \cdot \nabla v \, da, \quad (4.8)$$

and the linear form is given by the right-hand side of equation (4.2):

$$l(v) = \int_{\Omega} f v \, da. \quad (4.9)$$

Galerkin approximation. The Ritz-Galerkin approximation of equation (4.7) is constructed by specifying a set of functions ϕ_i that span a subspace, V^h , of H_e^1 , and posing the weak form on that space: Find $u_h \in V^h$ such that

$$a(u_h, v_h) = l(v_h) \quad \text{for all } v^h \in V_0^h. \quad (4.10)$$

Because V^h is spanned by the functions ϕ_i , we express u_h and v_h as linear combinations of the basis functions,

$$u_h = \sum_{i=1}^{\bar{N}} u_i \phi_i + \sum_{i=\bar{N}+1}^N u_i \phi_i, \quad (4.11)$$

$$v_h = \sum_{j=1}^{\bar{N}} v_j \phi_j. \quad (4.12)$$

In (4.11) and (4.12), let the interior nodes be the first \bar{N} nodes. Substituting these expressions for u_h and v_h into equation (4.10), and requiring the result to hold for any choice of $\{v_1, \dots, v_n\}$ leads to a discrete form of the VBVP which may be expressed as a system of linear equations,

$$\mathbf{K} \mathbf{u}_h = \mathbf{F}, \quad (4.13)$$

in which \mathbf{K} will be of dimension $\bar{N} \times \bar{N}$ with components given by

$$K_{ij} = a(\phi_i, \phi_j) \quad \text{and} \quad F_j = f(\phi_j). \quad (4.14)$$

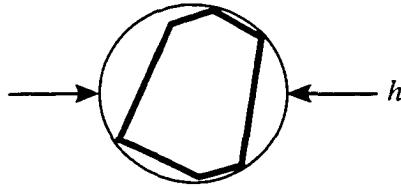


Figure 4.1: The mesh size parameter h is typically taken as the diameter of the smallest circle enclosing the largest element of the mesh

VETFEM basis. Constructing the basis of the approximation space involves creating a subdivision of the problem domain, and then using the VETFEM procedure to generate the element shape functions. To this end, suppose that the problem domain Ω is partitioned into a set of polygonal subdomains $\{\Omega_1, \Omega_2, \dots, \Omega_{LM}\}$, where the union of these subdomains is equivalent to the problem domain: $\Omega = \bigcup_{k=1}^{LM} \Omega_k$. Because the domain is polygonal by supposition, it can be covered exactly by a set of VETFEM-admissible elements. Further, consider a sequence of domain partitions, in which each partition, denoted T_i , is characterized by a mesh size parameter h_i . Here, h_i is a scalar taken to be the diameter of the smallest circle enclosing the largest element of the mesh: $h_i = \max_{1 \leq k \leq LM} \text{diam}(\Omega_k)$ (see Figure 4.1). Assume further that as i increases, h_i approaches zero.

For each mesh T_i , the VETFEM procedure generates a set of shape functions that are used to construct a basis for the approximation space V_h . The emphasis here is that by using the VETFEM, V_h may not be H^1 , and therefore $V_h \not\subset V$. In particular, this means that the integrals $\nabla u \cdot \nabla v$ in the bilinear form (equation 4.10) are not well defined. The linear form in (4.13) may be computed precisely, because the shape functions are H^1 on Ω . Thus, the element-based construction procedure effectively ignores the discontinuities in the basis functions in relation to the bilinear form. To account for this discrepancy, we note that the computed bilinear form is an approximation of the true bilinear form (4.8) and write it as

$$a_h(u, v) = \sum_k \int_{\Omega_k} \nabla u \cdot \nabla v \, da. \quad (4.15)$$

This may be considered shifting from the exact bilinear form to an approximate bilinear form, which is indicated in equation (4.15) by the subscript h . Posed on the nonconforming space, the Poisson problem is then to find $u_h \in V^h$, such that

$$a_h(u_h, v) = l(v) \quad \text{for all } v \in V_0^h. \quad (4.16)$$

The question at hand is whether solutions to equation (4.16) will converge in some sense to the true solution under mesh refinement ($h \rightarrow 0$). Specifically, the discrete solution u_h converges to the solution u of the true BVP, if

$$\lim_{h \rightarrow 0} \|u - u_h\|_h = 0, \quad (4.17)$$

where $\|\cdot\|_h$ denotes a suitably defined norm. To define the natural norm for this function space, consider the inner product induced semi-norm, which is given by

$$|u|_h = \sqrt{a_h(u, u)}. \quad (4.18)$$

We note that the semi-norm is a norm on V^h because the shape functions are continuous at the nodes. Therefore, by requiring that $u \in V^h$ we define the energy norm,

$$\|u\|_h = \sqrt{a_h(u, u)}, \quad u \in V^h. \quad (4.19)$$

Practically speaking, we are concerned with whether, as we move to successively finer meshes T_i , the approximation space will contain the true solution (completeness); and if it does, whether the true solution satisfies the discrete equations related to equation (4.16) (consistency). However, first we must determine, for the model BVP, whether a bounded solution exists.

Stability condition Stability of the approximate system means that a bounded solution exists. Stability is not generally addressed in the FEM because the true solution provides a strain energy bound on the approximate solution as demonstrated by Céa's lemma [68]. However, because a nonconforming basis shifts the problem to a space $V^h \not\subset V$, the fact

that the BVP is elliptic on V (and thus a bounded solution exists), is no indication as to whether a bounded solution exists on V^h .

A general bound for nonconforming methods was developed by Strang, and is referred to as Strang's lemma. Application of this bound hinges on the so-called “stability condition”, that requires the approximate bilinear form be V^h -elliptic (coercive, or positive definite [52]) in the sense that there exists a constant c independent of the mesh size h such that

$$a_h(u, u) \geq c\|u\|_h^2 \text{ for all } u \in V^h. \quad (4.20)$$

The uniform ellipticity of the Poisson problem on the VETFEM function space is relatively straightforward to show, using the argument presented in [11]. First, if the VETFEM functions are continuous, the bilinear form $a_h(\cdot, \cdot) = a(\cdot, \cdot)$ (this is referred to as a canonical extension of V^h to V), thus ellipticity on V^h is inherited from ellipticity on V , because $V^h \subset V$. If instead, V^h is nonconforming, the bilinear form is positive-definite in Ω because $a_h(v, v) = 0$ implies v is piecewise constant. Further, given the continuity at the nodes, a zero boundary condition implies $v \equiv 0$. Therefore, the Poisson equation is uniformly elliptic on the nonconforming space.

Completeness Completeness is related to the order of the PDE: an approximation space for a PDE of order $2m$ is complete if it contains all polynomials of order m . Both the Poisson equation and elasticity equations are second order, thus completeness simply means that the approximation space must contain all linear solution fields. Thus as the $h \rightarrow 0$ and the local behavior of the solution over any element becomes predominately linear, that solution can be represented precisely. The completeness of VETFEM approximation spaces is discussed in Chapter 3.

Consistency At this point, we know that a unique solution to the approximate problem exists; and as the mesh is refined, the expanding approximation space will come continuously closer to including the true solution of the Poisson equation. Now the question is

whether the solution to the discrete equations approaches the true solution. Or, alternatively, how well does the true solution satisfy the discrete equations? This issue was first made mathematically concrete and termed “consistency” by Strang, in what has become known as “Strang’s lemma”. In this lemma, Strang demonstrated that the general error estimate applicable to nonconforming methods with elliptic bilinear forms [21, 68] is:

$$\|u - u_h\|_h \leq C \left(\inf_{v_h \in V_h} \|u - v_h\|_h + \sup_{v_h \in V_h} \frac{|a_h(u, v_h) - l(v_h)|}{\|v_h\|_h} \right). \quad (4.21)$$

The first term on the right-hand side is identical to that of equation (2.28), and therefore equation (4.21) is a generalization of Céa’s lemma. This term is a measure of the “distance” between the approximation and the true solution, and shows that the distance is optimal in the norm based on the approximate norm. But how close is the approximate norm to the true norm? This is quantified by the second term, called the consistency term. The consistency term is induced by shifting to the approximate bilinear form, and is caused by the nonconformity of the element-based shape functions. This is clearly seen by noting that, because there are no delta functions in the linear form $l(v_h)$, it is exactly evaluated by the element-by-element evaluation. Therefore, from equation (4.8) the linear form may be replaced by the exact bilinear form, and the last term of equation (4.15) may then be rewritten as

$$\frac{|a_h(u, v_h) - l(v_h)|}{\|v_h\|_h} = \frac{|a_h(u, v_h) - a(u, v_h)|}{\|v_h\|_h}. \quad (4.22)$$

This form shows that the second term in equation (4.10) is a measure of the error introduced by shifting from the true bilinear form to the approximate one. Note that, if the VETFEM functions are conforming, e.g. if four node rectangular elements are used, then $V^h \subset V$ and $a_h(u, v_h) = a(u, v_h)$, thus the consistency term vanishes and Strang’s lemma becomes equivalent to Céa’s lemma.

Strang’s lemma is applicable only to problems in which the approximate bilinear form a_h of equation (4.15) is uniformly elliptic on the approximation space, and therefore application of this lemma must be accompanied by a demonstration that the ellipticity of the bilinear form.

The Strang lemma shows that convergence of nonconforming methods is dependent on the vanishing of the consistency term. Here, the desirable property is for the consistency term to vanish as the mesh-size parameter decreases. This is often termed the “consistency condition,” and is expressed as:

$$\lim_{h \rightarrow 0} \sup_{v_h \in V_h} \frac{|a_h(u, v_h) - l(v_h)|}{\|v_h\|_h} = 0. \quad (4.23)$$

There are various numerical and analytic means to evaluate the behavior of the consistency term. These range from simple numerical tests to more difficult analytic evaluations. The earliest, and perhaps the most common approach is the engineering patch test [81, 79]. This test provides a straightforward numerical means to demonstrate both that the method is complete and consistent. This approach is limited, however, by the fact that it establishes convergence of a limited set of boundary data [4], thus it is a necessary condition but its sufficiency is disputed. On the other hand, this test has been regularly used since the 1960’s and has proven to be a reliable indicator of convergence for practical problems of solid mechanics. Strang demonstrated that the patch test may also be applied in a more general variational setting. A more general analytic test was proposed by Stummel, which he called the “generalized patch test” [69]. In contrast to the conventional patch test, this test does not depend on the particular boundary value problem under consideration, and its results are applicable to all problems for which the nonconforming method is stable. This test is more difficult to apply than the patch test, but provides a necessary and sufficient condition for convergence for a sequence of nonconforming spaces.

4.2.1 The patch test

The patch test is the simplest and most common means of gaining insight into the convergence properties of nonconforming finite elements. The test typically consists of subjecting a regular, arbitrarily partitioned domain to boundary conditions implicating a linear analytic solution. The test is passed if the approximate method reproduces the linear solution exactly. That this suggests the method is convergent may then be intuitively deduced by

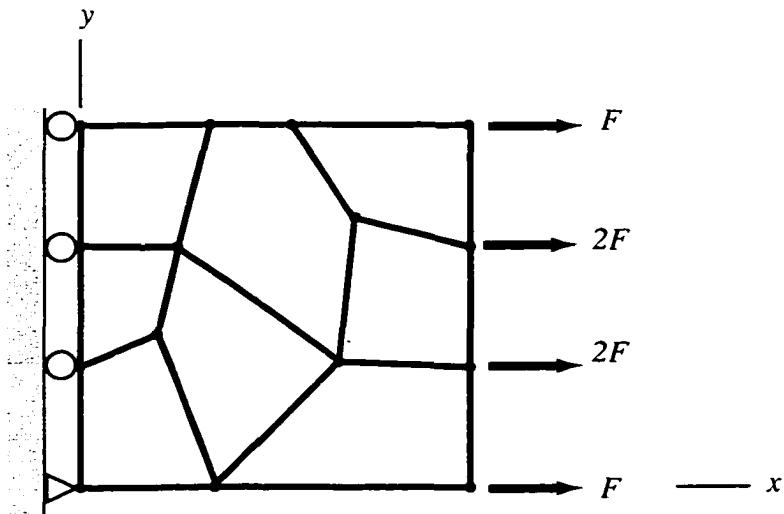


Figure 4.2: A patch test including essential and natural boundary conditions

reasoning that, as the element size diminishes ($h \rightarrow 0$) and the local solution becomes predominately linear, because the method reproduces any locally linear variation exactly, the approximate solution will converge to the true solution. The assumption implicit in this reasoning is that correspondence between $a(\cdot, \cdot)$ and $a_h(\cdot, \cdot)$ for a globally linear solution implies correspondence for a locally linear solution. This conclusion, however reasonable, has apparently never been proven. This section contains the results from numerical and analytic forms of the patch test and demonstrates that the VETFEM elements pass both forms of the test.

Numerical patch test The patch test is used in the context of a VETFEM code that generates numerical approximations to the linear elastic BVP described in Chapter 2.

The numerical patch test is most thoroughly described in [79], which presents three versions of the test, denoting them A, B, and C. Test A consists of forming the stiffness matrix and forcing vector and checking whether the exact nodal displacement vector \mathbf{u} satisfies the discrete equations. In test B, the exact boundary displacements $\bar{\mathbf{u}}$ are prescribed at all boundary nodes, and the discrete equations are solved for nodal values of displacement.

The solution is then checked against the exact nodal solution. These tests are meant to demonstrate that for linear solutions $a_h(u, v_h) = a(u, v_h)$, and also to constitute a necessary condition for convergence. Test C involves applying traction boundary conditions along with minimal displacement restraints to prevent rigid body motion. This test is meant to demonstrate sufficient conditions for convergence, although it appears that this test is not universally accepted as conclusive [8, 4].

Several numerical patch tests (e.g. Figure 4.2) were performed on the VETFEM. The tests included a wide variety of element shapes, some of which included concave elements. The tests included some of type B, but were generally hybrids of the B and C forms consisting of both displacement and traction boundary conditions [64]. In these tests, a region such as that shown in Fig. 4.2 was partitioned into a set of plain-strain elements. A specific example is included to demonstrate the results. In this example, the material response to strain was linear and isotropic, with elastic constants $E = 1E7$ and $\nu = 0.3$, and the traction applied to the patch is $T_x = 2E5$. The exact solution is

$$u_x = 0.01820x$$

$$u_y = 0.00546y$$

The patch test was conducted using both formulations of the consistency constraints discussed in Chapter 3. The displacement results are included in Table 4.1. The numerical solution matched the exact solution to machine precision. In addition, the computed strain and stress fields were constant and the stresses matched the constant traction to machine precision. The results of all patch tests conducted were consistent with this example. These tests demonstrate that the shape-function formulation incorporated properties which would enable these functions to pass the patch test in the context of two-dimensional linear elasticity, for either type of VETFEM consistency constraints.

The extent to which the numerical form or “engineering” form [38] of the patch test provides proof of consistency in the VETFEM is more limited than for most nonconforming elements. In typical applications, the test is applied to elements that have a fixed, but

Node		coordinate		displacement	
Number		x	y	u_x	u_y
1		0.0000000E+00	0.0000000E+00	0.0000000E+00	0.0000000E+00
2		0.3500000E+01	0.0000000E+00	0.6370000E-01	0.7538975E-14
3		0.6000000E+01	0.0000000E+00	0.1092000E+00	-0.9133255E-14
4		0.0000000E+00	0.2000000E+01	0.0000000E+00	-0.1560000E-01
5		0.1500000E+01	0.2000000E+01	0.2730000E-01	-0.1560000E-01
6		0.3000000E+01	0.2000000E+01	0.5460000E-01	-0.1560000E-01
7		0.6000000E+01	0.2000000E+01	0.1092000E+00	-0.1560000E-01
8		0.0000000E+00	0.4000000E+01	0.0000000E+00	-0.3120000E-01
9		0.1500000E+01	0.4000000E+01	0.2730000E-01	-0.3120000E-01
10		0.3000000E+01	0.4000000E+01	0.5460000E-01	-0.3120000E-01
11		0.6000000E+01	0.4000000E+01	0.1092000E+00	-0.3120000E-01
12		0.0000000E+00	0.6000000E+01	0.0000000E+00	-0.4680000E-01
13		0.3500000E+01	0.6000000E+01	0.6370000E-01	-0.4680000E-01
14		0.6000000E+01	0.6000000E+01	0.1092000E+00	-0.4680000E-01

Table 4.1: Example nodal values of displacement from the patch test are numerically equivalent to the exact analytic solution

mappable shape-function formulation. That is, each element in the mesh utilizes the same known shape functions that have been prescribed on a suitably defined parent element. The VETFEM, however, does not incorporate prescribed shape functions. Instead it only prescribes properties of the shape functions. The specific polynomial functions depend entirely on the geometry of the elements used in the particular patch. This being the case, subjecting a patch of elements to the numerical patch test essentially validates the specific basis functions defined for the specific elements used in the test, and if enough successful tests are performed, strongly suggests that the construction procedure is valid. Therefore, we turn to the variational form of the patch test, which allows for evaluation of the method without relying on specific mesh configurations.

Analytic patch test In order to demonstrate that the shape-function construction process utilized by the VETFEM generates functions with properties that guarantee passage of the test, we use the more general variational form of the patch test. In this application we switch to the Poisson equation. Strang [68, 67] demonstrated that the patch test is equivalent to testing whether for an arbitrary polynomial P_m of degree m , and for each basis function $\phi_j \in V_h$, there holds

$$a_h(P_m, \phi_j) = a(P_m, \phi_j). \quad (4.24)$$

In equation (4.24), $a(P_m, \phi_j)$ is the exact bilinear form and $a_h(P_m, \phi_j)$ is the approximate form. Note that $a(\cdot, \cdot)$ contains Dirac delta functions at element boundaries, but is well defined because the gradient of ϕ_j is multiplied by a smooth function in the integrand of $a_h(\cdot, \cdot)$. For the Poisson problem we are concerned that a linear field be represented exactly, in which case the polynomial we consider is linear ($m = 1$) so that $\Delta P = 0$ in Ω . Now, setting $P_m = P_1 = a + bx + cy$, we have the following expression for the left-hand side of equation (4.24):

$$a_h(P_1, \phi_j) = \sum_{k \in K^j} \int_{\Omega_k} \nabla P_1 \cdot \nabla \phi_j \, da, \quad (4.25)$$

where K^j is the set of elements that contain node j . Because the gradient of P_1 is constant,

$$a_h(P_1, \phi_j) = \sum_{k \in K^j} \nabla P_1 \cdot \int_{\Omega_k} \nabla \phi_j \, da. \quad (4.26)$$

Further, $\nabla \phi_j$ is smooth in each element, so that the Divergence theorem may be used to rewrite equation (4.26) as

$$a_h(P_1, \phi_j) = \sum_{k \in K^j} \nabla P_1 \cdot \int_{\partial \Omega_k} \phi_j \mathbf{n}_l \, ds. \quad (4.27)$$

The integrals in equation (4.27) are clearly non-zero, therefore in order for equation (4.24) to hold, the sum must include, as Strang put it, “lucky” cancellations. To examine this further, we break the boundary integrals into a sum of line integrals along the element faces,

$$a_h(P_1, \phi_j) = \sum_{k \in K^j} \sum_{l \in F^k} \nabla P_1 \cdot \int_{\Gamma_l} \phi_j \mathbf{n}_l \, ds, \quad (4.28)$$

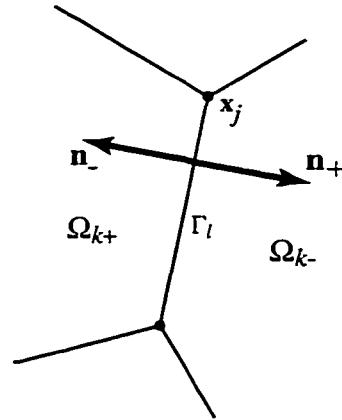


Figure 4.3: Elements sharing face Γ_l . Shape function contributions to the bilinear form cancel pairwise on internal edges provided that the Céa test is satisfied

where F^k is the set of faces Γ_l contained by element Ω_k . Because element face Γ_l is straight, the outward unit normal \mathbf{n}_l is constant and may be taken outside the integral, so that

$$a_h(P_1, \phi_j) = \sum_{k \in K^j} \sum_{l \in F^k} C_{kl} \int_{\Gamma_l} \phi_j \, ds \quad (4.29)$$

where $C_{kl} = \nabla P_1 \cdot \mathbf{n}_l$. Note: By construction, shape function integrals $\int_{\Gamma_l} \phi_j \, ds$ are known constant values, and are equal to linear functions integrated on the element faces. Similarly, the value of the integral for faces that do not contain node j is zero. Therefore, equation (4.29) only involves contributions from integrals on element faces that contain node j . As shown in Figure 4.3, each interior face Γ_l is shared by two elements, Ω_{k+} and Ω_{k-} . Therefore, two integrals are computed for each element face. Further, the values of $\int_{\Gamma_l} \phi_j \, ds = 0$ and ∇P_1 are identical for each integration. Only the outward normals differ, and they differ only in sign: $\mathbf{n}^{+l} = -\mathbf{n}^{-l}$. Therefore, each term in equation (4.29) will either be zero or cancel pairwise over adjacent elements. These facts together constitute the “lucky” cancellation necessary to conclude that

$$a_h(P_1, \phi_j) = 0. \quad (4.30)$$

Turning now to the right-hand side of equation (4.24) it is observed that this term is not

strictly zero as a consequence of equation (4.2) because $\phi_j \notin V$. However, because $u = P_1$ is smooth, Green's first identity yields

$$a(P_1, \phi_j) = \int_{\Omega} \nabla P_1 \cdot \nabla \phi_j \, da \quad (4.31)$$

$$= \int_{\Omega} -\Delta P_1 \phi_j \, da - \int_{\partial\Omega} \phi_j \nabla P_1 \cdot \mathbf{n} \, ds. \quad (4.32)$$

The first term on the right-hand side vanishes because P_1 is linear. Because the domain boundary is a polygon, the boundary integral may be written as the sum of integrals along the polygonal edges, i.e.

$$a(P_1, \phi_j) = \sum_{\Gamma_l \in \partial\Omega} \nabla P_1 \cdot \mathbf{n}_l \int_{\Gamma_l} \phi_j \, ds. \quad (4.33)$$

As mentioned above, the line integrals $\int_{\Gamma_l} \phi_j \, ds$ are constants that all cancel pairwise on element edges. Thus,

$$a_h(P_1, \phi_j) = 0. \quad (4.34)$$

This result demonstrates that VETFEM finite elements pass the patch test for the Poisson problem with Diriclet boundary conditions. This means that, for smooth data (in which the linear form can be integrated exactly), the consistency term in the Strang lemma equation (4.21) vanishes as the mesh size is reduced. The key ingredient for passing the analytic form of the patch test was the continuity of the zero-order moment of the shape functions across element edges, and it is this property that caused the VETFEM elements to pass the numerical patch test for the equations of linear elasto-statics.

Unlike the approximability condition for conforming methods, these evaluations demonstrate that the patch test is passed only for the bilinear forms corresponding to the displacement formulation of the linear elasto-statics problem and for the Poisson equation with homogenous boundary conditions. It is desirable, however, to demonstrate the convergence properties of the VETFEM elements in a manner that is not tied to a specific VBVP.

4.2.2 The generalized patch test

The generalized patch test was introduced by Stummel [69] and establishes necessary and sufficient conditions for the convergence of a nonconforming method. In contrast to the traditional patch test, this method is not related to a specific BVP. The test establishes that a method is convergent provided that, for every bounded sequence of trial-function spaces V_h in which the mesh size parameter $h \rightarrow 0$, the following relation holds:

$$T_i(\psi, u_h) = \sum_{k=1}^M \int_{\partial\Omega_k} \psi u_h n_l ds \rightarrow 0 \quad i = 1, 2 \quad (4.35)$$

for all test functions $\psi \in C_0^\infty(\Omega)$. In equation (4.35), n_l are components of the outward normal vector on the boundary of element Ω_k , and C_0^∞ is the set of infinitely differentiable functions that are zero on $\partial\Omega$.

To evaluate the VETFEM in light of the generalized patch test, suppose that K is a sequence of domain partitions, and that each partition in K satisfies the normal subdivision requirements that: $\Omega = \bigcup_{k=1}^m \Omega_k$ and $\Omega_i \cap \Omega_j = \emptyset$ for $i \neq j$. Further suppose that each partition is characterized by a mesh size parameter h in which $h = \max_{1 \leq k \leq m} \text{dia}(\Omega_k)$, and finally that $\psi \in C^\infty(\Omega)$, $\psi = 0$ on $\partial\Omega$.

Equation (4.35) consists of integrals around boundaries of individual elements, and because element boundaries are composed of piecewise continuous element faces, it can be written as

$$T_i(\psi, u_h) = \sum_{k=1}^M \sum_{l \in F^k} \int_{\Gamma_l} \psi u_h n_l ds \quad i = 1, 2. \quad (4.36)$$

Notice that element faces are either in $\partial\Omega$, or are common sides of two elements. If they are on the domain boundary, the contributions of those integrals to T_i will be zero because ψ is homogenous on the domain boundary. Therefore, consider the set L of interior element faces ($\Gamma_1, \Gamma_2, \dots, \Gamma_L$). Equation (4.36) can then be rewritten as

$$T_i(\psi, u_h) = \sum_{l=1}^L \int_{\Gamma_l} (\psi u_h^+ n_i^+ - \psi u_h^- n_i^-) ds \quad i = 1, 2. \quad (4.37)$$

In equation (4.37), the integrals over element faces are composed of two values, one from the element on the each side of the face (see Figure 4.3). The superscripts +/- represent

quantities defined with respect to elements on either side of the face, so that, e.g. $u_h^+(\mathbf{x})$, $\mathbf{x} \in \Gamma_l$ is derived from the element whose outward unit normal is $\mathbf{n}_l = \mathbf{n}_l^+(\mathbf{x})$. With this definition for the positive sense of \mathbf{n} , it is clear that $\mathbf{n}_l^- = -\mathbf{n}_l^+$, and equation (4.37) may be further simplified to

$$T_i(\psi, u_h) = \sum_{l=1}^L \int_{\Gamma_l} \psi (u_h^+ - u_h^-) n_i \, ds \quad i = 1, 2. \quad (4.38)$$

Denoting the jump in the displacements as $[u]$, and noting that \mathbf{n} is constant for each face, equation (4.38) may be written

$$T(\psi, u_h) < \sum_{l=1}^L \int_{\Gamma_l} \psi [u] \, ds, \quad (4.39)$$

where T is written for the magnitude of the vector T_i . Parameterizing ψ and $[u]$ in terms of the side length measure s , and then expanding ψ in a Taylor series around a point $b \in \Gamma_l$ yields

$$\int_{\Gamma_l} \psi [u] \, ds = \psi(b) \int_{\Gamma_l} [u] \, ds + \psi'(b) \int_{\Gamma_l} (s - b) [u] \, ds + \dots H.O.T. \quad (4.40)$$

Because the element shape functions satisfy the Ceà conditions (i.e. integrals of $[u]$ are continuous over any edge), two components of (4.40) vanish, leaving

$$\int_{\Gamma_l} \psi [u] \, ds = \psi'(b) \int_{\Gamma_l} s [u] \, ds + \dots H.O.T. \quad (4.41)$$

By neglecting higher order terms and applying the Cauchy-Schwarz inequality

$$\left| \int_{\Gamma_l} \psi [u] \, ds \right| \leq |\psi'(b)| \sqrt{\int_{\Gamma_l} s^2 \, ds} \sqrt{\int_{\Gamma_l} [u]^2 \, ds}, \quad (4.42)$$

and because $\psi \in C^\infty$, the first term may be bound by its maximum value C_1 on any face,

$$|\psi'(b)| \leq C_1 = \max_{b \in \Gamma_l} |\psi'(b)| \quad (4.43)$$

The size of the second term on the right-hand side of (4.42) is $O(h^{1.5})$. The third term is the integral of the square of the jump. Because both u_h^+ and u_h^- are polynomial functions of s , the integral in the third term may be written as,

$$\int_{\Gamma_l} [u]^2 \, ds = \int_{\Gamma_l} (a_0 + a_1 s + a_2 s^2 + \dots H.O.T.) \, ds \quad (4.44)$$

which as $s \rightarrow 0$ bounds this third term as order $h^{0.5}$:

$$\sqrt{\int_{\Gamma_l} [u]^2 ds} = O(h). \quad (4.45)$$

Using these bounds for the three terms in equation (4.42), and then substituting equation (4.39) yields

$$T < \sum_{l=1}^L Ch^2. \quad (4.46)$$

As the mesh size decreases, the number of element faces increases as the square of h , giving the final bound for the size of T as

$$T \leq C \quad (4.47)$$

This simple result does not allow us to establish that the method passes the generalized patch test. It does not however prove that the test is not passed. In developing this bound, no use was made of the fact that piecewise linear behavior can be represented exactly which may allow for a tighter bound on the size of $[u]$ of equation (4.44). This result, however, strongly suggests that, in addition to continuity of moments of shape functions, higher order moments may be required to achieve strict convergence of the VETFEM. These issues are the subject of ongoing research.

Chapter 5

Numerical Results

In this chapter, the results of some simple plane-strain problems solved by the VETFEM are presented, demonstrating the capabilities and convergence properties of the method.

5.1 Cantilever beam

Consider a cantilever beam subject to end load, as shown in Figure 5.1. The displacements are prescribed as essential boundary conditions at $x = 0$, whereas tractions are prescribed on the remaining boundary segments. The following parameters were used for the cantilever beam problem: $E = 1000$; $\nu = 0.3$; the depth of the beam is one unit and the length of the beam is ten units. Several meshes were used for this analysis, one of which is shown in Figure 5.2.

This simple example provides a comparison between the behavior of VETFEM elements and conventional finite element formulations (full integration implementation). This beam bending problem requires a relatively fine mesh because fictitious shear strains are introduced by the fully integrated linear element shape functions. The computed tip deflection was within 2.8% between the VETFEM and conventional FEM. The VETFEM solutions were stiffer than the FEM results, and mesh (b) was slightly more flexible than mesh (c). The conventional mesh was also used as input to the VETFEM code; the results for the

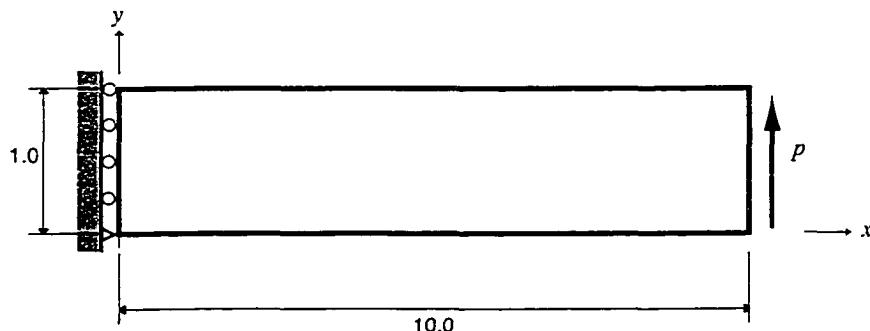


Figure 5.1: Cantilever beam problem

VETFEM and conventional FEM using the same mesh were identical to within 0.2%. Because the elements are all rectangular, the shape functions for the two methods are identical, thus the small difference is due to the differences in the numerical integration rules for the two methods.

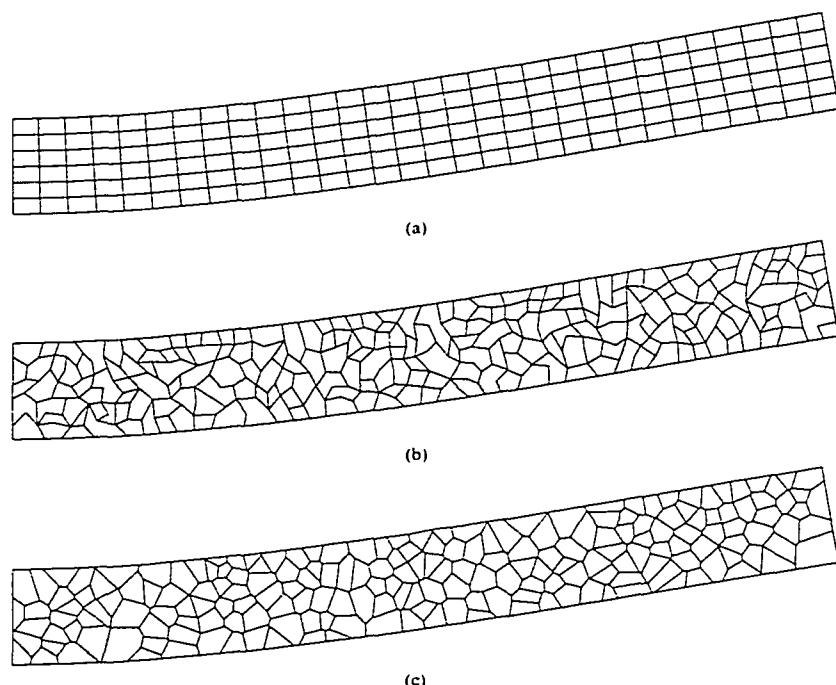


Figure 5.2: Example deformed VETFEM meshes of the cantilever beam.
(a)standard quadrilateral mesh; (b) mesh created by randomly placing nodes, joining nodes by segments, and defining elements; (c)mesh based on Voronoi partition from randomly placed seed points

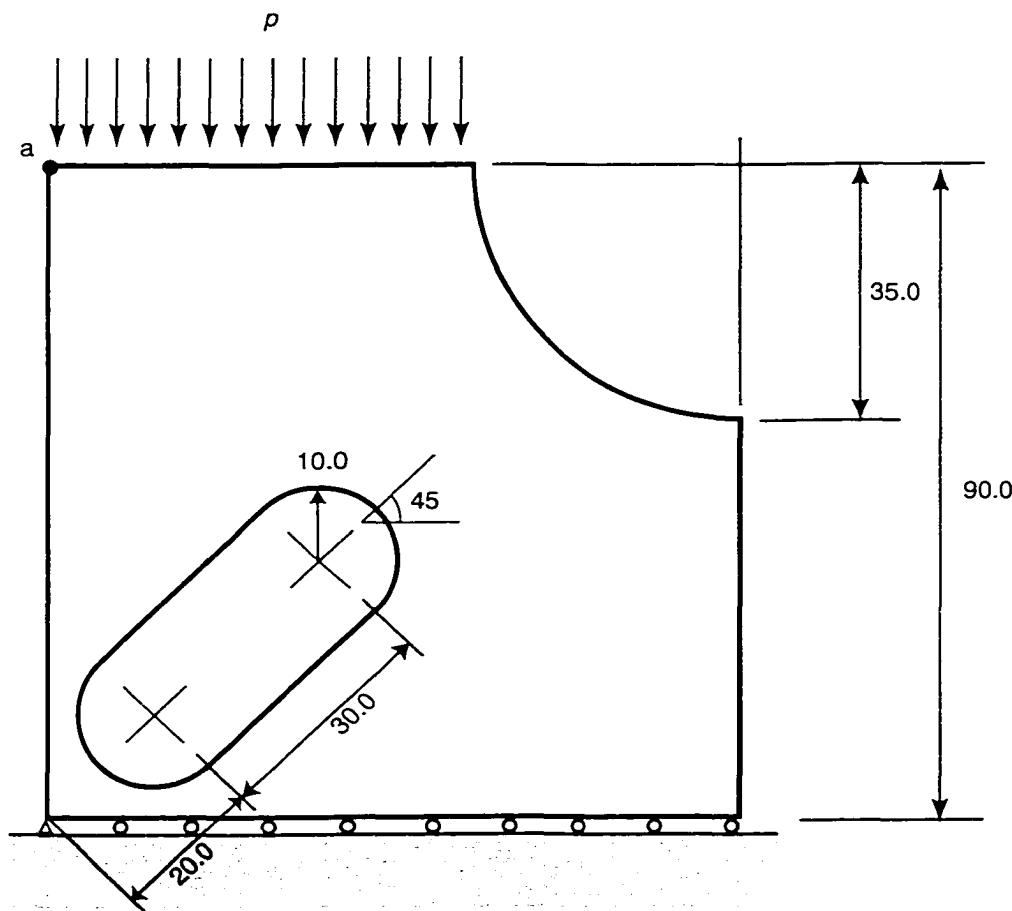


Figure 5.3: Slotted plate subjected to a constant traction p . The displacement sampling point is located at a

5.2 Slotted plate

As a second example, consider the slotted plane-strain panel shown in Figure 5.3. The problem was analyzed using several VETFEM meshes, and a FEM mesh consisting of bilinear quadrilateral elements. Four example meshes are shown in Figure 5.4. The vertical displacements at the point labelled “ a ” in Figure 5.3 are summarized in Table 5.1. As in the case of the plane-strain beam, the quadrilateral mesh was used as input to the VETFEM method as well. The resulting displacements differed negligibly from those computed by

the conventional FEM.

type	name	nodes	elements	displacement
FEM	Mesh a	614	540	1.00000
VETFEM	Mesh a	614	540	1.00053
VETFEM	Mesh b	554	328	0.99008
VETFEM	Mesh c	822	489	1.00517
VETFEM	Mesh d	1072	637	1.00752

Table 5.1: Comparison of results from the slotted plate problem using using FEM and VETFEM

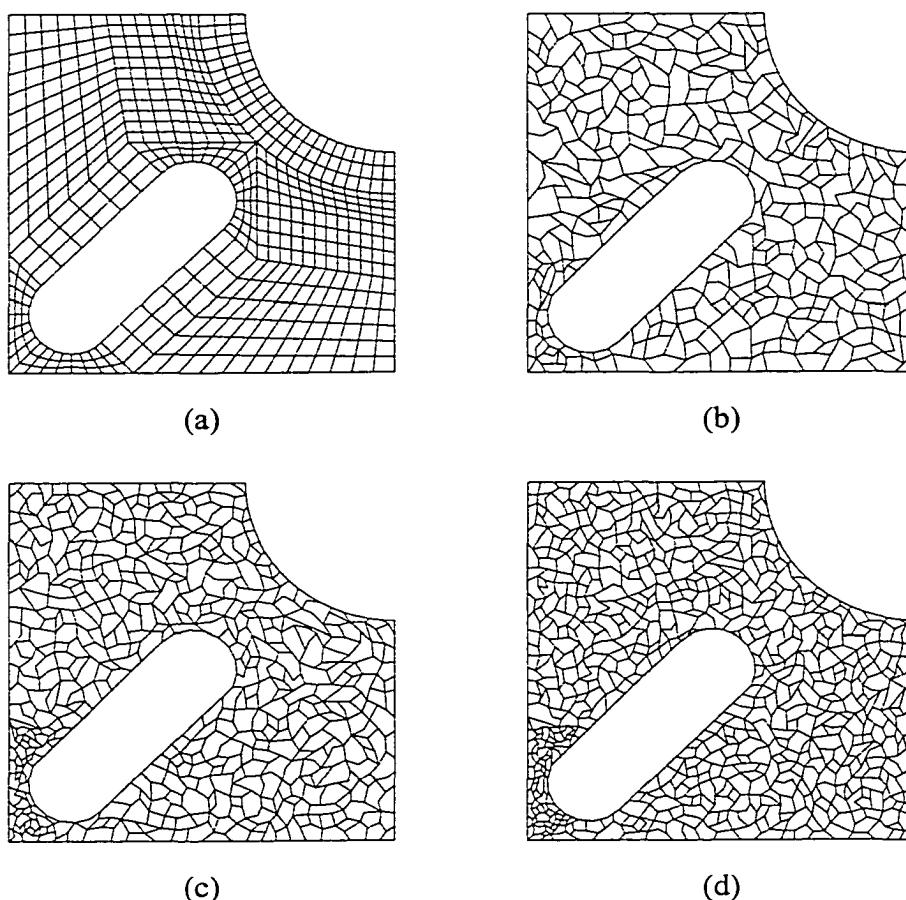


Figure 5.4: Meshes used in slotted plate problem. Quadrilateral mesh: with (a) 540 elements, 614 nodes; VETFEM meshes: (b) 328 elements, 554 nodes (c) 489 elements, 822 nodes (d) 637 elements, 1072 nodes

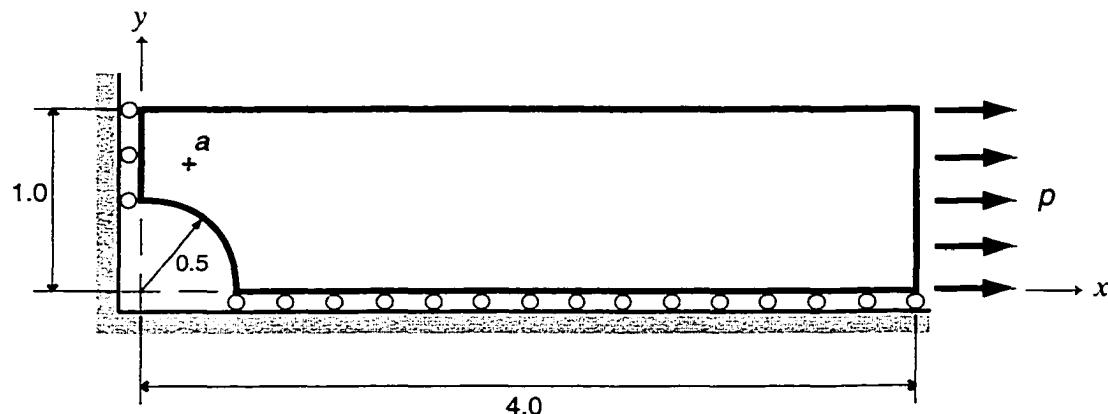


Figure 5.5: One quarter of a strip plate with a circular hole subjected to a constant traction p . The point a denotes the stress and displacement sampling point

5.3 Plate with a circular hole

In this example, we seek to provide insight into the convergence properties of VETFEM displacement and stress solutions. A typical FEM convergence study consists of analyzing a problem using a sequence of progressively finer meshes. Mesh definitions begin with an initial, coarse mesh from which subsequent meshes are successively constructed by self-similar refinement (e.g. quadrilaterals are subdivided into four quadrilaterals) of the preceding mesh. Each mesh has an associated approximation space that is defined by the element shape functions, the “smallest” approximation space being associated with the initial mesh. Because the mesh refinement is self-similar, the approximation space associated with a refined mesh contains all previous approximation spaces as proper subsets. This occurs because the coarse mesh shape functions can be written as linear combinations of fine mesh shape functions. As a result, the approximate solution on a fine mesh must be equal to or better than a coarse mesh because the Galerkin method guarantees that the best approximation solution will be selected. This idea is represented in the left half of Figure 5.6. With the VETFEM, it is not generally possible to construct a sequence of self-similar meshes because element shapes are arbitrary polygons. The lack of self-similar mesh re-

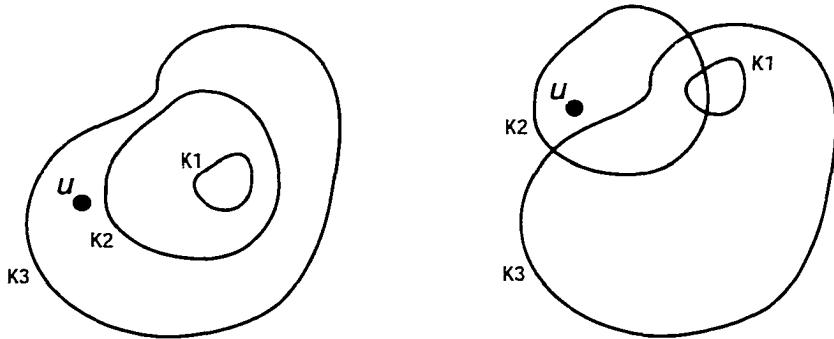


Figure 5.6: This figure illustrates function spaces generated by self-similar and non-self-similar mesh refinement. The function spaces appear as “bodies” labeled K_i , and u represents the exact solution. The spaces on the left correspond to self-similar mesh refinement. Function spaces related to coarse meshes (e.g. K_1) are always subsets of spaces corresponding to more refined mesh spaces, and the approximation improves as the function space grows. The spaces on the right arise from non-self-similar mesh refinement. Here smaller approximation spaces may or may not be subsets of the larger spaces, and the quality of the approximation is affected by both the size and position the function space

finement, means that coarse mesh approximation spaces may not be contained in refined mesh solution spaces (i.e. shape functions of coarse meshes cannot be precisely represented as linear combinations of the refined mesh shape functions). In this case, it is possible for an approximation solution generated by a coarse mesh to be better than that of a more refined mesh. This idea is illustrated in the right half of Figure 5.6. Because of these facts, a VETFEM convergence study must include many evaluations so that the convergence trends may be observed.

Consider a long, flat, linearly elastic sheet with a circular hole at its center, and that is subjected to an applied tensile load p . Boundary conditions include symmetry on the left and bottom edges, and a constant applied traction applied on the right edge. Due to the geometric and boundary-condition symmetries only one quarter of the plate is analyzed, as shown in Figure 5.5. This evaluation examines both the displacement and the first principal stress at point a . Note: an analytic series solution of arbitrary precision is found in cf. [36],

Mesh Sequence	Figure	Page
MS1	5.12	115
MS2	5.13	117
MS3	5.14	119
MS4	5.15	121

Table 5.2: Page references for mesh data

however in this analysis we compare VETFEM solutions with sequences of meshes such that $h \rightarrow 0$.

Approximately 150 analyses were performed with various mesh configurations. There were, loosely speaking, four separate sequences of meshes, denoted: MS1, MS2, MS3, and MS4. The sequences differed in the way mesh refinement was accomplished. In mesh sequence MS1, for example, the initial mesh was refined by strongly concentrating elements in the region of the displacement and stress sampling point a ; whereas in MS4, mesh refinement was uniform throughout the domain. Mesh sequences MS2 and MS3 represent intermediate levels of mesh gradation. Meshes are characterized by the parameter h which is taken as the diameter of the smallest circle enclosing the element that contains the sampling point a . This differs from the typical approach setting h equal to the diameter of the circle enclosing the largest element. The reason the latter was not adopted is it does not reflect local mesh refinement. That is, unless the size of the largest elements are reduced, the mesh size parameter h will remain unchanged. In the case of MS1, this measure of mesh refinement may over-predict the “size” of the approximation space because the mesh near the stress concentration at $(0.5, 0.0)$ was not significantly refined.

In each mesh, the location of the sampling point corresponded to the centroid of a single element. Sample meshes and mesh details for each mesh sequence are provided on the pages shown in the Table 5.2. The vertical displacement of point a for these analyses is shown in Figures 5.7 and 5.8. Figure 5.7 on page 110 plots the vertical displacement of the

sampling point against the element size, h , with displacement data broken down by mesh sequence. This figure shows that the displacement solution clearly converges; however, the results for small values h vary by as much as 3.9%. The data also shows that MS4 tends to provide a better approximation than the other groups for similar values of h . This scatter may be due to inconsistencies introduced by the non-conforming shape functions. Consider, for example, an edge Γ_l on which a homogenous essential boundary condition is specified. The VETFEM shape functions will not in general exactly satisfy this essential boundary condition pointwise along the boundary. Instead, the VETFEM approximation “weakly” satisfies the essential boundary conditions in that $\mathbf{u}^h = \mathbf{0}$ (Kronecker delta property) at the nodes on this edge, and $\int_{\Gamma_l} \mathbf{u}^h \cdot \mathbf{n} \, ds = 0$ by the Céa constraints. Despite the fact that these conditions allow the VETFEM to pass the classical form of the patch test, the results from the generalized patch test strongly suggest that they are insufficient to ensure strict convergence. These numerical results appear to confirm this contention. In order to clearly pass the generalized patch test, the Céa constraints of order one, i.e. $\int_{\Gamma_l} s\phi \, ds$ must be continuous across element boundaries. This form of constraints is the subject of ongoing research.

Figure 5.8, which is a log plot of the same data, shows that the convergence of the displacement is of order less than one. However, the quality of the approximation for many relatively coarse meshes appears to be quite good.

The computed values of the first principal stress at point a are shown in Figures 5.9, 5.10, and 5.11. Figure 5.9 on page 112 shows plots of the first principal stress against the element size h , and is broken down by mesh sequence. In this plot it is apparent that, although the displacement solutions computed by mesh sequence 4 tended to be more accurate than those of the other sequences, the stress solution may not be. The stress solution at small h varied by only 0.08%. Figure 5.10 shows a log plot of the stress solution data. In this plot, there is a clear convergent trend, however, when the y-axis is compressed, the stress convergence appears to be nearly constant as shown in Figure 5.11 . This corresponds to

the behavior predicted by the generalized patch test.

From this evaluation, it appears that convergence properties of the VETFEM may benefit by incorporating higher order Céa consistency requirements. The current implementation provides that the zero-order moment of the shape functions are continuous across element boundaries. This may easily be increased to the first moment. This change would cause the VETFEM to clearly pass the Generalized patch test, and may enhance the method's convergence properties. This modification is being pursued in ongoing research.

Recently, an instability was identified in the implementation of the generalized Gaussian elimination routine that adversely impacted the quality of shape functions for some element topologies. Preliminary numerical results using a modified implementation indicate that significantly improved convergence results may be achieved. This is currently an area of active research.

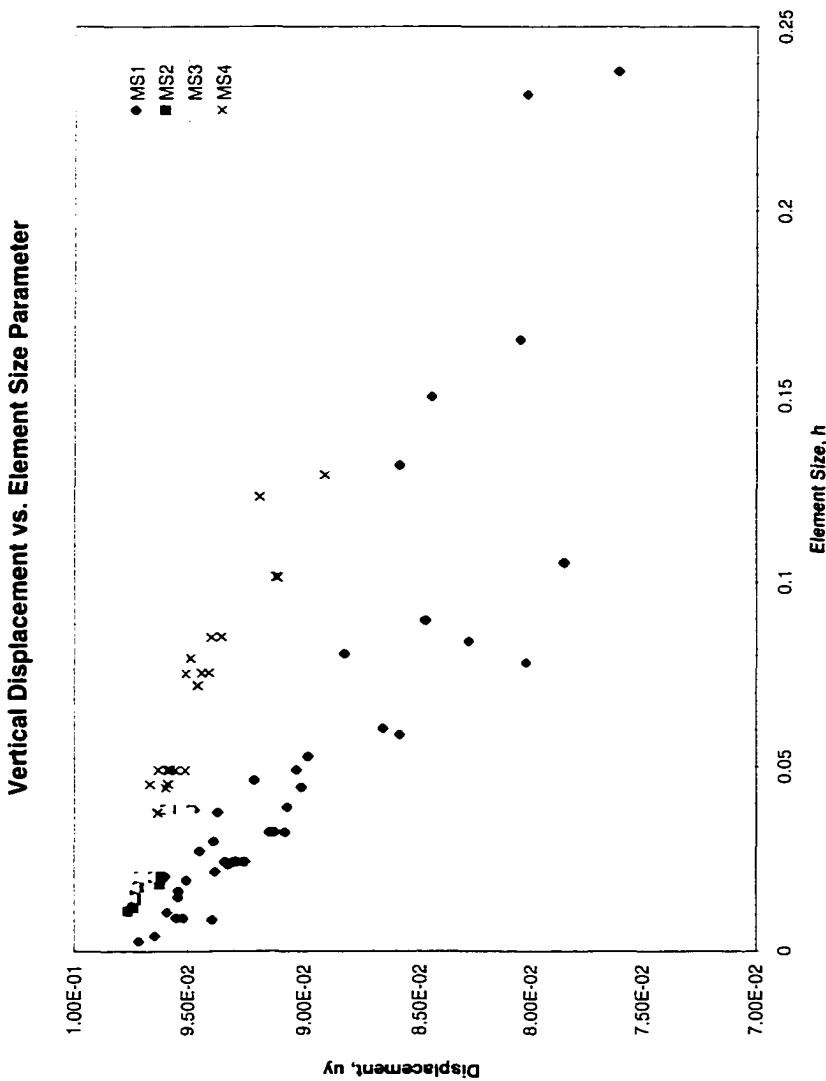


Figure 5.7: In this figure, the vertical displacement of point a is plotted against mesh size parameter h

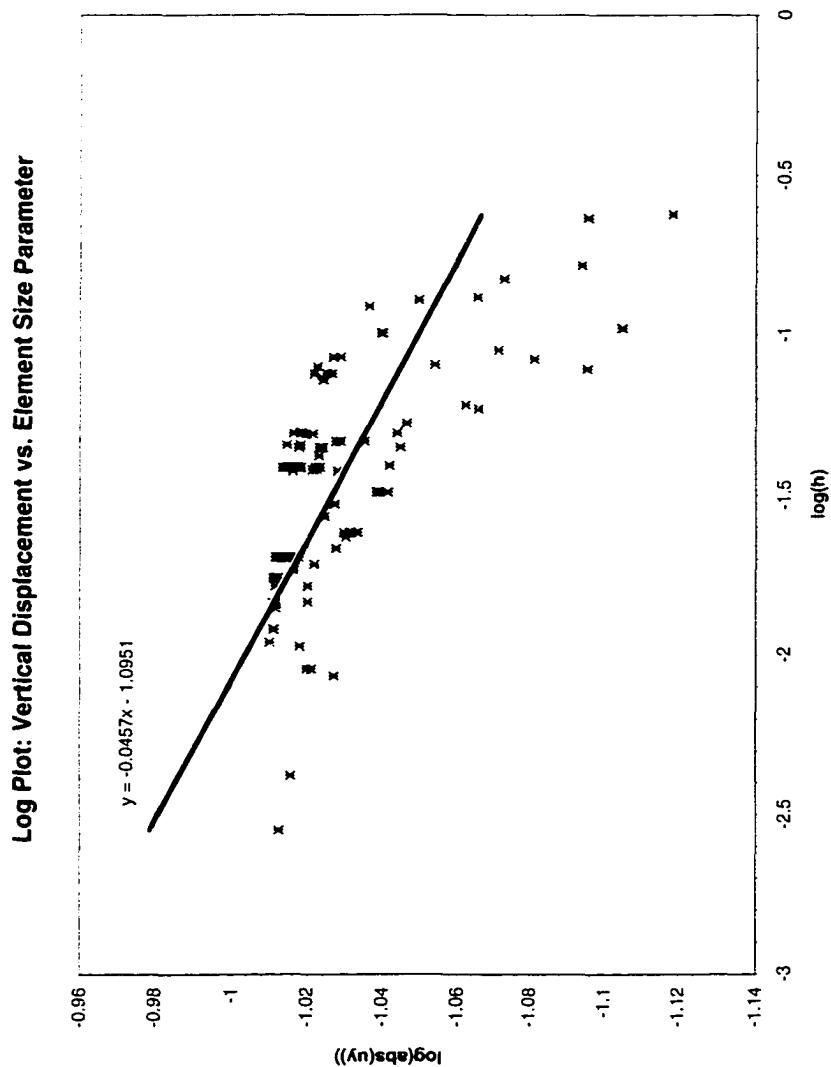


Figure 5.8: In this figure, the log of the vertical displacement of point a is plotted against the mesh size parameter h .

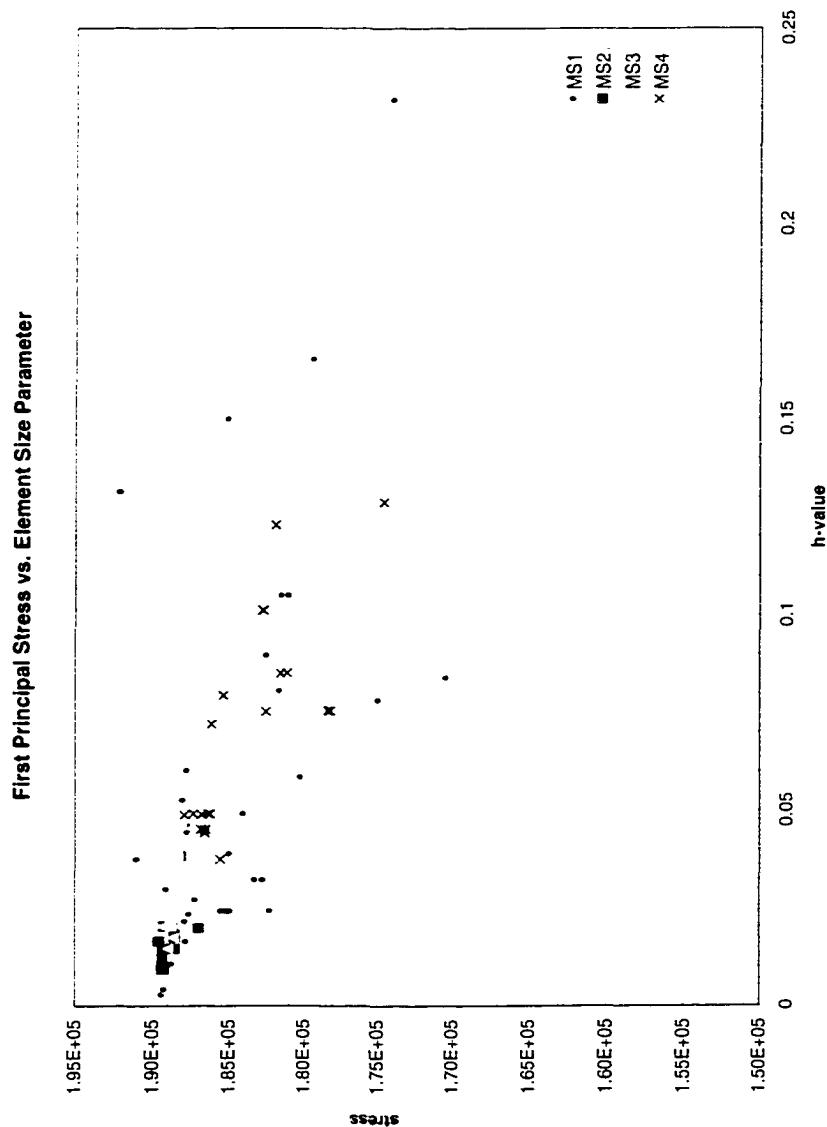


Figure 5.9: In this figure, the first principal stress at point a is plotted against the mesh size parameter h

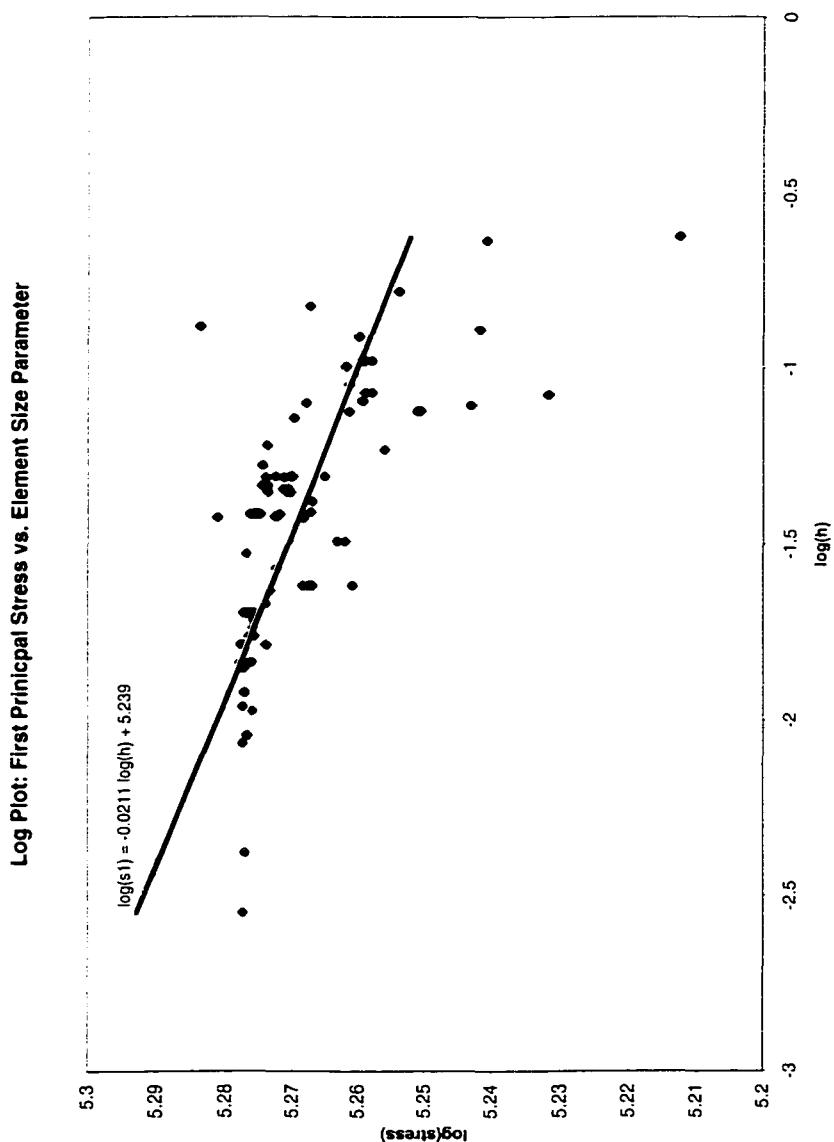


Figure 5.10: In this figure, the log of the first principal stress at point a is plotted against the log of the mesh size parameter h

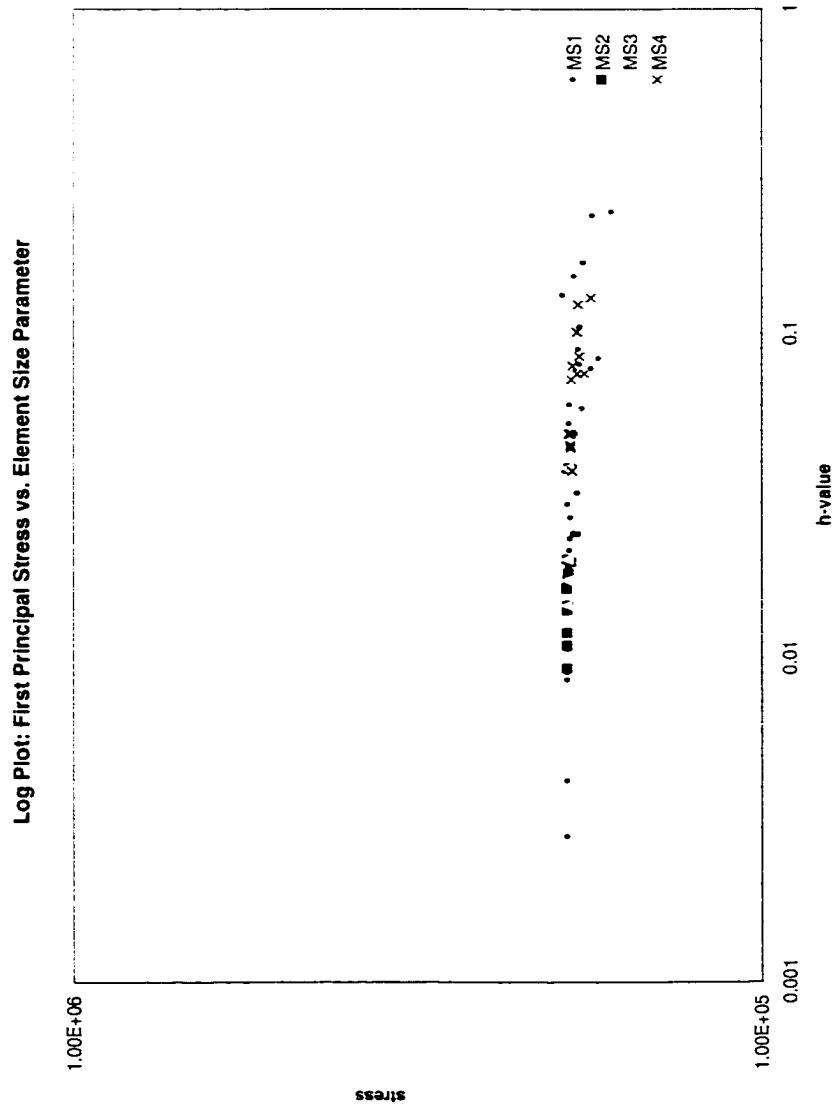


Figure 5.11: In this figure, the log of the first principal stress at point a is plotted against the log of the mesh size parameter h . With the stress scale expanded, the convergence of the method appears as very nearly constant

5.3.1 Mesh data

In this evaluation, VETFEM meshes were broken down into four mesh sequences labeled: MS1, MS2, MS3, and MS4. This section briefly describes each mesh sequence; provides a table listing the range of elements, nodes, and mesh size parameter h for each sequence, and contains example meshes from each sequence.

Mesh Sequence 1 (MS1) This series of meshes concentrates elements in a small region near the stress sampling point at $(0.25, 0.7)$. The size of the element containing the stress sampling point ranges from an h -value of 0.002827 to 0.237936. Table 5.3 lists mesh data from selected meshes, and example meshes are provided in Figure 5.12.

mesh	elements	nodes	element h	mesh h
1	29	53	0.2379	0.4493
2	31	58	0.1654	0.4494
3	33	59	0.1050	0.4489
4	35	62	0.1050	0.4491
5	39	70	0.1500	0.4479
6	41	71	0.0840	0.4493
7	43	77	0.0897	0.4380
8	45	75	0.0783	0.4445
9	59	97	0.0603	0.4201
10	65	104	0.0806	0.4199
11	67	107	0.0587	0.4534
12	79	123	0.0445	0.4458
13	85	132	0.0492	0.4458
14	97	148	0.0390	0.4458
15	103	157	0.0323	0.4458
16	99	150	0.0323	0.4458
17	101	155	0.0323	0.4458
18	121	181	0.0242	0.4458
20	141	207	0.0242	0.4458
25	1001	1345	0.0086	0.5909
30	871	1173	0.0042	0.3924
31	1747	2340	0.0028	0.3383
min	29	53	0.0028	0.3383
max	1747	2340	0.2379	0.5909

Table 5.3: Mesh Sequence 1 selected data. Note that element h refers to the size of the element containing the stress sampling point and that mesh h refers size of the largest element in the mesh

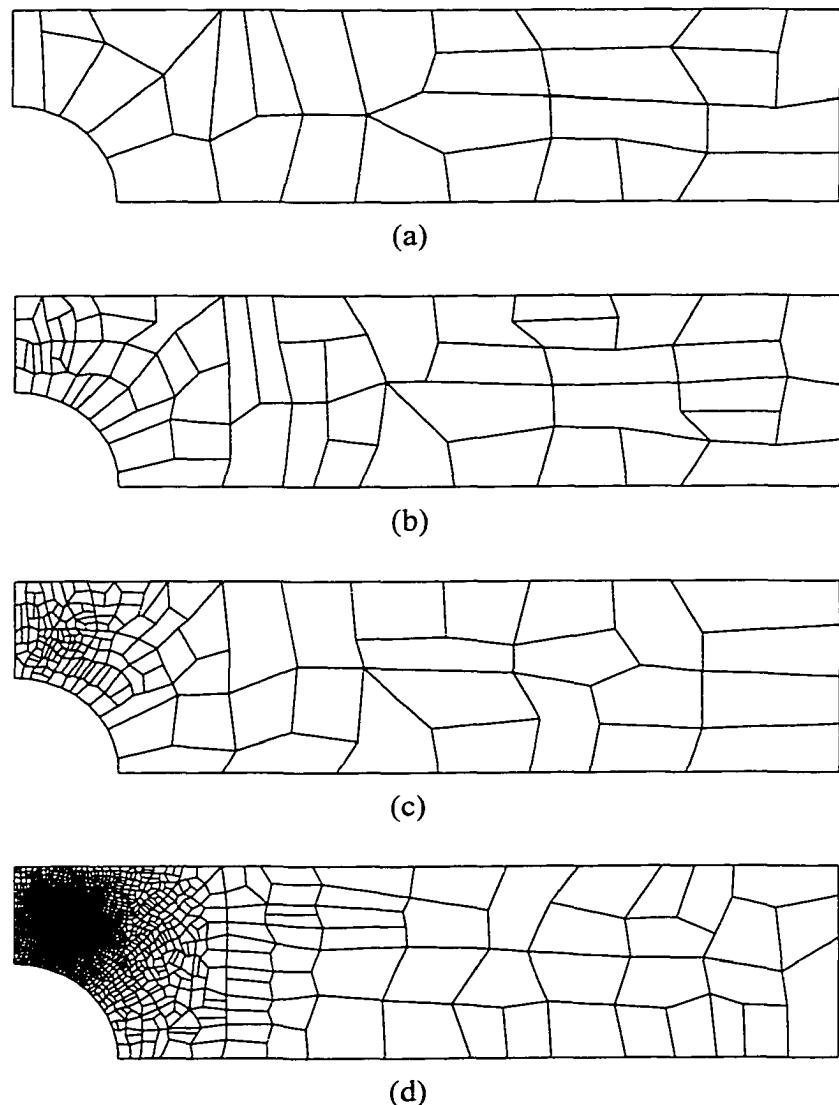


Figure 5.12: Example meshes from sequence 1. (a) Mesh 1, (b) Mesh 10, (c) Mesh 20, (d) Mesh 30

Mesh Sequence 2 (MS2) In this sequence, mesh refinement extends farther from the stress sampling point than in MS1. Table 5.4 lists mesh data from selected meshes, and example meshes are shown in Figure 5.13.

mesh	elements	nodes	element h	mesh h
1	333	478	0.0384	0.1580
2	353	505	0.0384	0.1443
3	381	540	0.0384	0.1443
4	405	571	0.0384	0.1443
5	431	610	0.0384	0.1428
6	455	642	0.0199	0.1428
7	471	660	0.0183	0.1428
8	503	700	0.0202	0.1367
9	535	742	0.0202	0.1414
10	549	762	0.0202	0.1414
11	561	776	0.0202	0.1414
12	573	792	0.0202	0.1414
13	585	805	0.0202	0.1304
14	609	840	0.0202	0.1304
15	659	909	0.0202	0.1245
16	697	962	0.0202	0.1245
17	681	939	0.0202	0.1245
18	635	876	0.0202	0.1245
19	717	989	0.0202	0.1245
20	747	1028	0.0202	0.1245
25	949	1311	0.0202	0.1239
30	1035	1429	0.0202	0.1049
40	2083	2812	0.0110	0.0773
min	333	478	0.0110	0.0773
max	2641	3547	0.0384	0.1580

Table 5.4: Mesh Sequence 2: Data for selected meshes. Note that element h refers to the size of the element containing the stress sampling point and that mesh h refers size of the largest element in the mesh

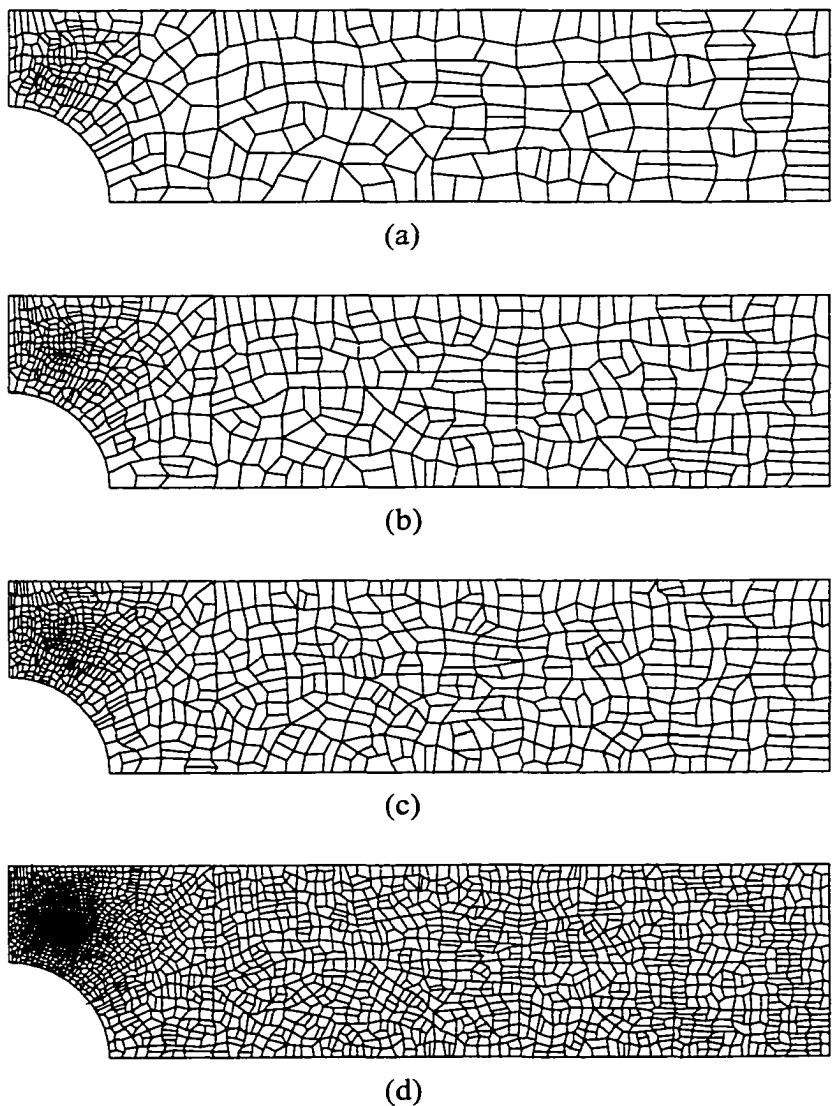


Figure 5.13: Example meshes from sequence 2. (a) Mesh 1, (b) Mesh 10, (c) Mesh 20, (d) Mesh 40

Mesh Sequence 3 (MS3) In this sequence, mesh gradation is modest but present. Table 5.5 lists mesh data from selected meshes, and example meshes are shown in Figure 5.14.

mesh	elements	nodes	element h	mesh h
1	247	360	0.0464	0.1568
2	267	386	0.0464	0.1580
3	285	410	0.0443	0.1568
4	305	439	0.0443	0.1485
5	319	459	0.0418	0.1485
6	319	459	0.0418	0.1485
7	337	485	0.0379	0.1485
8	363	520	0.0379	0.1485
9	415	585	0.0384	0.1465
10	449	629	0.0384	0.1458
11	491	684	0.0384	0.1458
12	533	742	0.0384	0.1304
13	553	769	0.0384	0.1245
14	603	837	0.0384	0.1245
15	657	910	0.0384	0.1245
16	711	982	0.0384	0.1245
17	773	1065	0.0202	0.1239
18	821	1132	0.0202	0.1049
19	865	1192	0.0202	0.1040
20	913	1255	0.0202	0.0955
25	1177	1613	0.0202	0.0831
30	1731	2358	0.0174	0.0773
min	247	360	0.0146	0.0747
max	1919	2607	0.0464	0.1580

Table 5.5: Mesh Sequence 3: Data for selected meshes. Note that element h refers to the size of the element containing the stress sampling point and that mesh h refers size of the largest element in the mesh

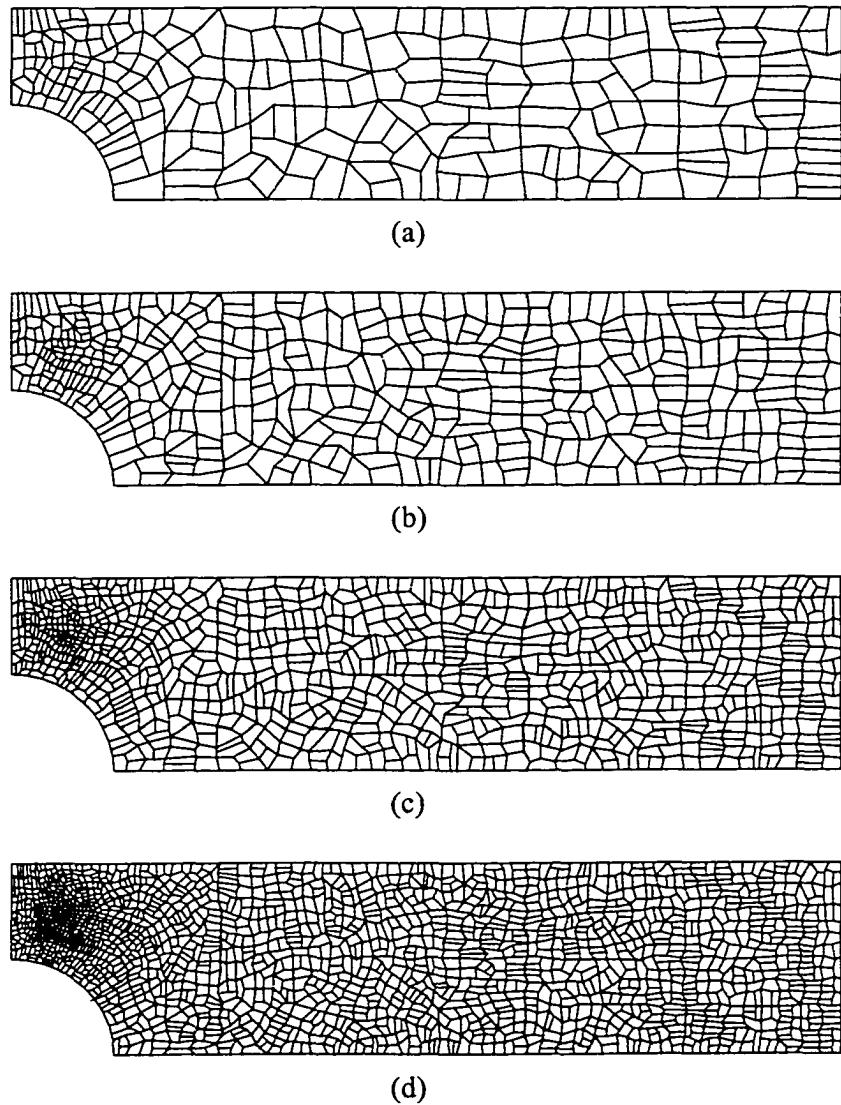


Figure 5.14: Example meshes from sequence 3. (a) Mesh 1, (b) Mesh 10, (c) Mesh 20, (d) Mesh 30

Mesh Sequence 4 (MS4) This sequence includes uniform mesh refinement, and no concentration of elements within the domain. Table 5.6 lists mesh data from selected meshes, and example meshes are shown in Figure 5.15.

mesh	elements	nodes	element h	mesh h
1	195	288	0.1287	0.1725
2	213	309	0.1229	0.1580
3	227	331	0.1013	0.1465
4	243	352	0.1013	0.1465
5	253	367	0.1013	0.1465
6	293	427	0.0852	0.1465
7	341	482	0.0852	0.1454
8	341	482	0.0852	0.1454
9	387	547	0.0755	0.1458
10	413	581	0.0755	0.1340
11	437	615	0.0753	0.1245
12	477	668	0.0795	0.1245
13	567	792	0.0722	0.1245
14	657	912	0.0722	0.1137
15	749	1034	0.0490	0.0972
16	823	1136	0.0490	0.0972
17	931	1279	0.0492	0.0850
18	967	1327	0.0492	0.0850
19	995	1364	0.0492	0.0850
20	1039	1425	0.0492	0.0854
25	1517	2066	0.0376	0.0747
min	195	288	0.0376	0.0747
max	1517	2066	0.1287	0.1725

Table 5.6: Mesh Sequence 4: Data for selected meshes. Note that element h refers to the size of the element containing the stress sampling point and that mesh h refers size of the largest element in the mesh

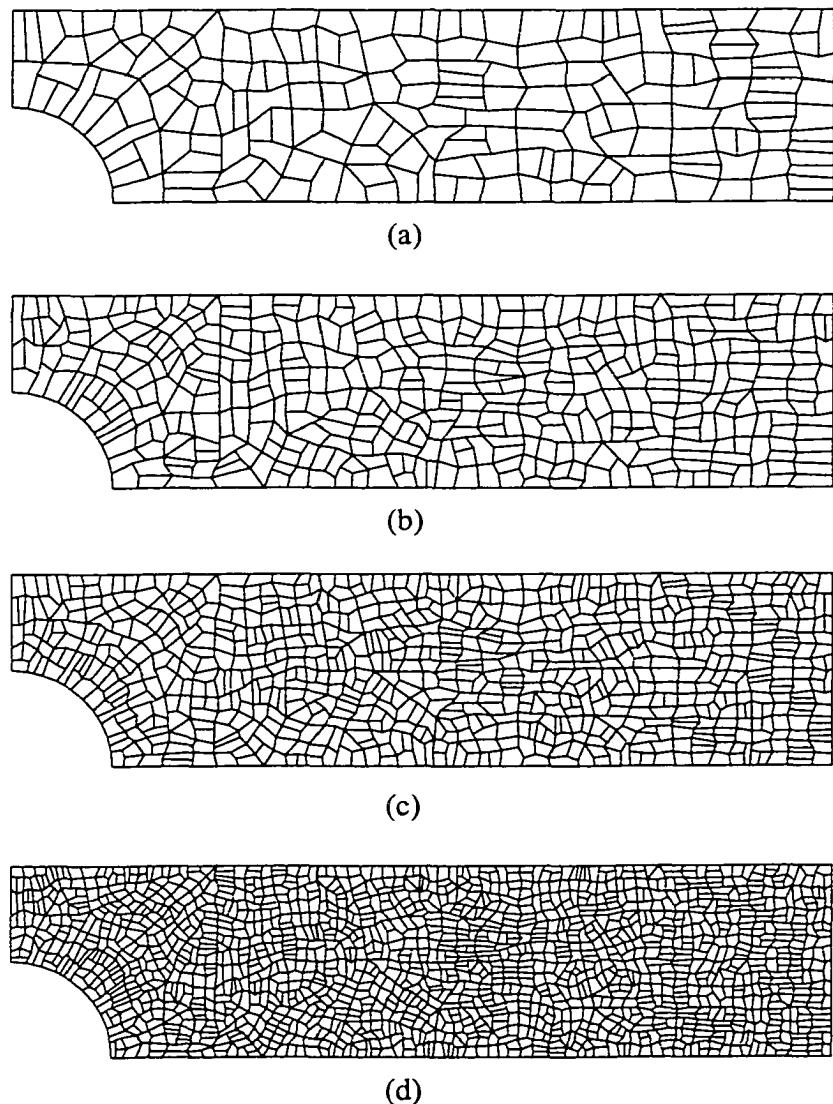


Figure 5.15: Example meshes from sequence 4. (a) Mesh 1, (b) Mesh 10, (c) Mesh 15, (d) Mesh 25

Chapter 6

Conclusion

The Variable-Element-Topology Finite Element Method, presented here in the context of linear elasticity, provides a general means for generating shape functions for finite elements with general polygon shapes. The VETFEM retains the favorable properties of the standard FEM, such as straightforward imposition of boundary conditions, precise representation of domain boundaries and internal interfaces, and a sparse and banded global equation structure. At the same time, because of the significant flexibility with respect to element topology, mesh construction is nearly as simple as meshless discretizations.

The crucial aspect in which the VETFEM differs from the conventional FEM is that no isoparametric transformation to a parent element is used to define the shape functions. Accordingly, the geometry of VETFEM elements is not constrained by the topology of a parent element, or by the potential for singularities in an isoparametric transformation. The VETFEM shape functions are defined as polynomials in the physical coordinates of the problem, and are optimized for inter-element compatibility and interior smoothness. The global basis functions are generally nonconforming; however, convergence of the overall method is assured by the completeness of the approximation space and the satisfaction of the Céa conditions of order zero. It is noted that improved convergence may be obtained by increasing the order of the Céa condition satisfied by the shape functions. Similarly, the

Kronecker delta property of standard FEM basis functions is assured through imposition of an additional constraint.

The VETFEM formulation as presented is readily extensible to problems involving near-incompressibility, large deformations and three-dimensional implementations. An approach for nearly incompressible materials has been devised. The common FEM approaches for such problems involve using either selective reduced integration for standard elements, or using hybrid elements based on mixed formulations. The approach taken for the VETFEM is to maintain the displacement based approach and avoid volumetric strain locking by an appropriate modification to the test function spaces. Specifically, the basis functions are formulated so that each possesses constant divergence on the element, thus avoiding poor element performance as the incompressible limit is approached.

Although the example calculations presented in this dissertation involve small deformations in two-dimensions, there exist no essential barriers to extending the methodology to finite deformations and three-dimensional problems. In a finite-deformation setting, there is some advantage in adopting a total-Lagrangian formulation, in which only a single reference configuration is discretized. The element-level calculations then need be performed only once. The element-level calculations in the VETFEM, namely the calculation of the polynomial shape-function coefficients and the integration weights, are more extensive than those of the conventional FEM for both two- and three-dimensional problems. However, these calculations scale only linearly with the size of the problem. Element-level operations can therefore be expected to account for only a small fraction of the total computer time in large, production-level analyses. Furthermore, because the element-level calculations are independent for each element, they are highly parallelizable.

The VETFEM is completely compatible with the standard FEM, so that the two methods may be applied simultaneously to the same problem, in any spatial arrangement. No special consideration need be given the interface between VETFEM and FEM mesh regions. Indeed, a conventional FEM mesh is entirely suitable for use as a VETFEM mesh.

Accordingly, the prospect of using the VETFEM in transition regions, between regions that are easily meshed in conventional finite elements, is a particularly attractive feature of the proposed method.

Appendix A

Reference Information

A.1 Notation

The $L^2(\Omega)$ space. This is the space of functions with bounded square-integrals

$$L^2(\Omega) = \left\{ u : \int_{\Omega} |u(x)|^2 dx < \infty \right\} \quad (\text{A.1})$$

The standard Sobolev space H^1 .

$$H^1(\Omega) = \{ \partial^\alpha u \in L^2, |\alpha| \leq 1 \} \quad (\text{A.2})$$

The norm in the space H^1 .

$$\|v\|_{m,\Omega} = \left(\sum_{|\alpha| \leq m} \int_{\Omega} |\partial^\alpha v|^2 dx \right)^{1/2} \quad (\text{A.3})$$

The semi-norms.

$$|v|_{m,\Omega} = \left(\sum_{|\alpha|=m} \int_{\Omega} |\partial^\alpha v|^2 dx \right)^{1/2} \quad (\text{A.4})$$

The space C^k . The space of k times differentiable real functions on Ω

$$C^k(\Omega) = \{ \partial^\alpha u \text{ continuous}, |\alpha| \leq k \} \quad (\text{A.5})$$

The space $C^\infty(\Omega)$. C^∞ functions with compact support. The support of a function is simply the closure of the set of all points x such that $u(x) \neq 0$. A function with compact support $K_\phi = \text{supp}(\phi)$ contained in Ω means that K_ϕ is bounded and $K_\phi \subset\subset \Omega$.

$$C_\infty^k(\Omega) = \{\partial^\alpha u \text{ continuous}, |\alpha| \leq k\} \quad (\text{A.6})$$

Gradient operator.

$$(\nabla \mathbf{u})_{ij} = u_{i,j} \quad (\text{A.7})$$

Symmetric gradient operator.

$$\nabla_s \mathbf{u} = \frac{1}{2} (\nabla \mathbf{u} + \nabla^T \mathbf{u}) \quad (\text{A.8})$$

Null Space Let A be a linear operator, the null space of A is the set of all elements u such that

$$N(A) = \{u \in X : Au = 0\} \quad (\text{A.9})$$

A.2 Symbols

Symbol	Definition
N	number of nodes in the mesh
M	number of elements in the mesh
Ω_m	region occupied by element m
n_m	number of nodes contained by element m
$C_a^m (a = 1, \dots, n_m)$	list of nodes contained by element m
K^α	elements that contain node α
ϕ_a^m	shape function a for element m
Φ_j	global basis function number j
$\partial\Omega_l$	element face l
\mathbf{n}^I	unit normal on $\partial\Omega_l$
G^m	set of faces contained by element m
L^α	set of faces that contain node α
A^l	set of nodes contained by $\partial\Omega_l$

Table A.1: Definitions of symbols and data sets

A.3 Definitions

Function support The support of a continuous function defined on a domain in \mathbb{R}^n is the closure of the set $\{x : u(x) \neq 0\}$.

Compact support The support of a continuous function is called compact support with respect to Ω if it is bounded and it is a subset of the interior of a set: $K_\phi \subset\subset \Omega$ defined on a domain in \mathbb{R}^n is the closure of the set $\{x : u(x) \neq 0\}$. Outside the support of a function, it is natural to define it to be zero. When Ω is a bounded set, it is equivalent to say that u vanishes in a neighborhood of $\partial\Omega$.

V-elliptic bilinear forms Given a bilinear form $a : H \times V \rightarrow \mathbb{R}$, where V is an inner product space, we say that a is *V – elliptic* if there exists a constant $\alpha > 0$ such that

$$a(v, v) \geq \alpha \|v\|_V^2 \quad \forall v \in V. \quad (\text{A.10})$$

A.4 Theorems and identities

Theorem A.1 (Divergence Theorem)

$$\int_{\Omega} \nabla \cdot \mathbf{u} \, dA = \int_{\partial\Omega} \mathbf{u} \cdot \mathbf{n} \, ds \quad (\text{A.11})$$

Theorem A.2 (Green's Theorem)

$$\int_{\Omega} u \frac{\partial v_i}{\partial x_j} \, dA = \int_{\partial\Omega} u v n_j \, ds - \int_{\Omega} \frac{\partial u_i}{\partial x_j} v \, dA \quad (\text{A.12})$$

Theorem A.3 (Lax-Milgram lemma) Let V be a Hilbert space, let $a(\cdot, \cdot)$ be a continuous *V-elliptic* bilinear form, and let f : be a continuous linear form. Then the abstract variational problem: Find an element u such that

$$u \in V,$$

$$a(u, v) = f(v) \quad \forall u, v \in V, \quad (\text{A.13})$$

has one and only one solution.

Appendix B

VETFEM Software

B.1 General Gaussian elimination

This section describes a generalized form of Gaussian elimination that may be used to factorize rectangular or rank deficient systems of equations using uses row and column pivoting. Consider the following, general system of equations:

$$\mathbf{A}\mathbf{X} = \mathbf{B} \quad (\text{B.1})$$

Suppose \mathbf{A} is a $(m \times n)$ matrix in which $(m \leq n)$, \mathbf{X} is $(n \times l)$ and \mathbf{B} is $(m \times l)$. We wish to solve (B.1) for \mathbf{X} . Provided that $m < n$ and $\text{rank}(\mathbf{A}) > \text{rank}(\mathbf{B})$, a unique solution does not exist. Therefore, we identify a space of solutions. To accomplish this we use Gaussian elimination with row and column pivoting. This process will classify coefficients \mathbf{X} into bound \mathbf{X}_b and free \mathbf{X}_f coefficients. The precise values of bound coefficients are functions of the free coefficients. Once the free coefficients are specified, the bound coefficients are computed. Free coefficients may take on any real values. The process follows that of a standard Gaussian elimination scheme which is composed of a sequence of elimination cycles in which the coefficient matrix is transformed via row pivots, column interchanges and eliminations (for a detailed explanation see [73]).

In a typical factorization of the coefficient matrix \mathbf{A} in equation (B.1), the k th elimina-

tion cycle is composed of the following:

$$\begin{array}{c}
 \left[\begin{array}{ccccc} 1 & \times & \times & \times & \times \\ & \times & \times & \times & \times \\ & \times & x_{ik} & \times & \times \end{array} \right] \xrightarrow{\boldsymbol{P}_k} \left[\begin{array}{ccccc} 1 & \times & \times & \times & \times \\ & \times & x_{ik} & \times & \times \\ & \times & \times & \times & \times \end{array} \right] \xrightarrow{\boldsymbol{Q}_k} \\
 \text{Pivot selection} \qquad \qquad \qquad \text{Row Interchange}
 \end{array}$$

$$\left[\begin{array}{ccccc} 1 & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & \times & \times & \times & \times \end{array} \right] \xrightarrow{\boldsymbol{L}_k} \left[\begin{array}{ccccc} 1 & \times & \times & \times & \times \\ & x_{ik} & \times & \times & \times \\ & 0 & \times & \times & \times \end{array} \right] \xrightarrow{\text{Elimination}}$$

Column Interchange

(B.2)

Here the bold symbols indicate the matrix entries operated upon in the indicated procedure. The algorithm proceeds until the coefficient matrix is fully factorized, at which time it may be expressed as the partitioned matrix:

$$\left[\begin{array}{c|cc} 1 & \times & \times \\ 1 & \times & \times \\ 1 & \times & \times \end{array} \right] = \left[\begin{array}{c|c} \boldsymbol{I} & \boldsymbol{C} \end{array} \right]$$
(B.3)

Where \boldsymbol{I} is the $(l \times l)$ identity matrix and \boldsymbol{C} is a $(l \times n - l)$ matrix. The complete factorization can be expressed as a matrix product:

$$\boldsymbol{L}_{n-1} \boldsymbol{P}_{n-1} \cdots \boldsymbol{L}_2 \boldsymbol{P}_2 \boldsymbol{L}_1 \boldsymbol{P}_1 \boldsymbol{A} \boldsymbol{Q}_1 \boldsymbol{Q}_2 \cdots \boldsymbol{Q}_{n-1}. \quad (\text{B.4})$$

This may be rewritten in the form

$$(\boldsymbol{L}'_{n-1} \cdots \boldsymbol{L}'_1) (\boldsymbol{P}_{n-1} \cdots \boldsymbol{P}_1) \boldsymbol{A} (\boldsymbol{Q}_1 \cdots \boldsymbol{Q}_{n-1}) = \boldsymbol{L} \boldsymbol{P} \boldsymbol{A} \boldsymbol{Q}, \quad (\text{B.5})$$

where \boldsymbol{L} are eliminations, \boldsymbol{P} are pivots and \boldsymbol{Q} are column interchanges. Thus the factorization of (B.1) becomes

$$(\boldsymbol{L} \boldsymbol{P} \boldsymbol{A} \boldsymbol{Q}) \boldsymbol{Q}^T \boldsymbol{X} = \boldsymbol{L} \boldsymbol{P} \boldsymbol{B} \quad (\text{B.6})$$

$$(\boldsymbol{L} \boldsymbol{P} \boldsymbol{A} \boldsymbol{Q}) \bar{\boldsymbol{X}} = \boldsymbol{L} \boldsymbol{P} \boldsymbol{B} \quad (\text{B.7})$$

where $\bar{\mathbf{X}}$ accounts for column interchanges in the factorization of \mathbf{A} :

$$\mathbf{X} = \mathbf{Q}^T \bar{\mathbf{X}}. \quad (\text{B.8})$$

The matrix product \mathbf{LPAQ} represents the factorized form of \mathbf{A} , which has the partitioned form (B.3). Thus (B.7) can be written as a partitioned system,

$$\left[\begin{array}{c|c} \mathbf{I} & \mathbf{C} \end{array} \right] \left[\begin{array}{c} \mathbf{X}_b \\ \hline \mathbf{X}_f \end{array} \right] = \mathbf{LPB}. \quad (\text{B.9})$$

Here \mathbf{X}_b is the $(l \times l)$ matrix of bound coefficients, and \mathbf{X}_f is the $(n - l \times l)$ of free coefficients. Expanding (B.9) and rearranging terms leads to

$$\mathbf{X}_b = \mathbf{LPB} - \mathbf{CX}_f. \quad (\text{B.10})$$

This form gives an expression for the bound coefficients \mathbf{X}_b as a function of the free coefficients \mathbf{X}_f . With reference to (B.10), it allows us to write $\bar{\mathbf{X}}$ in terms of the free coefficients

$$\bar{\mathbf{X}} = \begin{bmatrix} \mathbf{LPB} - \mathbf{CX}_f \\ \mathbf{X}_f \end{bmatrix} = \begin{bmatrix} \mathbf{LPB} \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} -\mathbf{C} \\ \mathbf{I} \end{bmatrix} \mathbf{X}_f \quad (\text{B.11})$$

where $\mathbf{0}$ is the $(n - l \times n - l)$ matrix of zeros and \mathbf{I} is the $(n - l \times n - l)$ identity matrix. Finally by equation (B.8, we write we return the coefficient order the original arrangement to find

$$\mathbf{X} = \mathbf{Q}^T \begin{bmatrix} \mathbf{LPB} \\ \mathbf{0} \end{bmatrix} + \mathbf{Q}^T \begin{bmatrix} -\mathbf{C} \\ \mathbf{I} \end{bmatrix} \mathbf{X}_f \quad (\text{B.12})$$

$$\mathbf{X} = \mathbf{S} + \mathbf{CX}_f \quad (\text{B.13})$$

Bibliography

- [1] High-resolution trabecular bone reconstruction. [online]. Berkeley Orthopaedic Biomechanics Laboratory. <<http://biomech2.me.berkeley.edu/trab2.html>>, November 2000.
- [2] Meshing research corner. [online]. <<http://www.andrew.cmu.edu/user/sowen/mesh.html>>, November 2000.
- [3] ADAMS, M. F. Parallel multigrid solvers for 3d unstructured finite element problems in large deformation elasticity and plasticity. *International Journal For Numerical Methods in Engineering* 48, 8 (2000), 1241–1262.
- [4] BABUŠKA, I., AND NARASIMHAN, R. The babuška-brezzzi condition and the patch test: an example. *Computer methods in applied mechanics and engineering* 140 (1997), 183–199.
- [5] BELYTSCHKO, T., KRONGAUZ, Y., DOLBOW, J., AND GERLACH, C. On the completeness of meshfree particle methods. *International Journal For Numerical Methods in Engineering* 43, 5 (1998), 785+.
- [6] BELYTSCHKO, T., KRONGAUZ, Y., ORGAN, D., FLEMING, M., AND KRYSL, P. Meshless methods: An overview and recent developments. *Computer Methods in Applied Mechanics and Engineering* 139, 1-4 (1996), 3–47.
- [7] BELYTSCHKO, T., LU, Y. Y., AND GU, L. Element-free galerkin methods. *International Journal For Numerical Methods in Engineering* 37, 2 (1994), 229–256.
- [8] BELYTSCHKO, T., LUI, W., AND MORAN, B. *Nonlinear Finite Elements for Continua and Structures*. Wiley, New York, 2000.
- [9] BENZLEY, S. E., PERRY, E., MERKLEY, K., AND CLARK, B. A comparison of all hexagonal and all tetrahedral finite element meshes for elastic and elastic-plastic analysis. In *Proceedings, 4th International Meshing Roundtable* (1996), pp. 179–191.

- [10] BLACKER, T. Meeting the challenge for automated conformal hexahedral meshing. In *Proceedings, 9th International Meshing Roundtable, Sandia National Laboratories* (2000), pp. 11–19.
- [11] BRENNER, S. C., AND SCOTT, L. R. *The mathematical theory of finite element methods*. Texts in applied mathematics ; 15. Springer-Verlag, New York, 1994.
- [12] BREZZI, F., AND FORTIN, M. *Mixed and hybrid finite elements methods*. Springer series in computational mathematics ; 15. Springer-Verlag, New York, 1991.
- [13] CANANN, S. A., SAIGAL, S., AND OWEN, S. J. Editorial - special issue on 'unstructured mesh generation'. *International Journal For Numerical Methods in Engineering* 49, 1-2 (2000), 1–2.
- [14] CHARRAS, G. T., AND GULDBERG, R. E. Improving the local solution accuracy of large-scale digital image-based finite element analyses. *J Biomech* 33, 2 (2000), 255–259.
- [15] CHEN, J. K., BERAUN, J. E., AND JIH, C. J. An improvement for tensile instability in smoothed particle hydrodynamics. *Computation Mech* 23, 4 (1999), 279–287.
- [16] CHEN, J. S., PAN, C., ROQUE, C., AND WANG, H. P. A lagrangian reproducing kernel particle method for metal forming analysis. *Computational Mechanics* 22, 3 (1998), 289–307.
- [17] CHEN, J. S., PAN, C., AND WU, C. T. Large deformation analysis of rubber based on a reproducing kernel particle method. *Computational Mechanics* 19, 3 (1997), 211–227.
- [18] CHEN, J. S., PAN, C., WU, C. T., AND WING, K. L. Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Computer Methods in Applied Mechanics and Engineering* 139 (1996), 195–227.
- [19] CHEN, J. S., PAN, C. H., AND WU, C. T. Application of reproducing kernel particle method to large deformation contact analysis of elastomers. *Rubber Chemistry and Technology* 71, 2 (1998), 191–213.
- [20] CIARLET, P. G. *The finite element method for elliptic problems*. Studies in mathematics and its applications ; v. 4. North-Holland, Amsterdam ; New York New York, 1978.
- [21] CIARLET, P. G., AND LIONS, J. L. *Handbook of numerical analysis*, vol. II. North-Holland, Amsterdam ; New York New York, 1990.

- [22] CLOUGH, R. The finite element in plane stress analysis. In *Conference papers; American Society of Civil Engineers 2d Conference on Electronic Computation* (1960).
- [23] CROUZEIX, M., AND RAVIART, P. Conforming and nonconforming finite element methods for solving the stationary stokes equations i. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 7, R-3 (1973), 33–76.
- [24] DE BOOR, C., AND RON, A. Computational aspects of polynomial interpolation in several variables. *Mathematics of Computation* 58, 198 (1992), 705–727.
- [25] DEMMEL, J. W., GILBERT, J. R., AND LI, X. Y. S. An asynchronous parallel supernodal algorithm for sparse gaussian elimination. *Siam Journal On Matrix Analysis and Applications* 20, 4 (1999), 915–952.
- [26] DOLBOW, J., AND BELYTSCHKO, T. Numerical integration of the galerkin weak form in meshfree methods. *Computational Mechanics* 23, 3 (1999), 219–230.
- [27] DUARTE, A. A review of some meshless methods to solve partial differential equations. Tech. Rep. TICAM 95-06, University of Texas, 1995.
- [28] DUARTE, C. A., AND ODEN, J. T. An h-p adaptive method using clouds. *Comput Method Appl Mech Eng* 139, 1-4 (1996), 237–262.
- [29] DYKA, C. T., RANDLES, P. W., AND INGEL, R. P. Stress points for tension instability in sph. *Int J Numer Method Eng* 40, 13 (1997), 2325–2341.
- [30] FALK, R. Nonconforming finite element methods for the equations of linear elasticity. *Mathematics of Computation* (1991), 529–550.
- [31] GALLAGHER, R., PADLOG, J., AND BIJLAARD, P. Stress analysis of heated complex shapes. *ARS Journal* (1962), 700–707.
- [32] GINGOLD, R., AND MONAGHAN, J. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the Royal Astronomical Society* 181 (1977), 375–389.
- [33] GOPALAKRISHNAN, J., AND PASCIAK, J. E. Multigrid for the mortar finite element method. *Siam Journal On Numerical Analysis* 37, 3 (2000), 1029–1052.
- [34] HERRMANN, L. A bending analysis for plates. In *Matrix methods in structural mechanics : [proceedings of the conference held at Wright-Patterson Air Force Base, Ohio, 26-28 October 1965* (1965).

- [35] HORGAN, C. Korn's inequalities and their applications in continuum mechanics. *SIAM Review* 37 (1995), 491–511.
- [36] HOWLAND, R. On the stresses in the neighbourhood of a circular hole in a strip under tension. *Philosophical transactions of the royal society of London, series A* 229 (1930), 49–86.
- [37] HUGHES, T. J. R. Generalization of selective integration procedures to anisotropic and nonlinear media. *International Journal For Numerical Methods in Engineering* 15 (1980), 1413–1418.
- [38] HUGHES, T. J. R. *The finite element method : linear static and dynamic finite element analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1987.
- [39] IRONS, B. M. A frontal solution program for finite element analysis. *International Journal For Numerical Methods in Engineering* 2 (1970), 5–32.
- [40] JOHNSON, G. R., AND BEISSEL, S. R. Normalized smoothing functions for sph impact computations. *Int J Numer Method Eng* 39, 16 (1996), 2725–2741.
- [41] KRONGAUZ, Y., AND BELYTSCHKO, T. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Comput Method Appl Mech Eng* 131, 1-2 (1996), 133–145.
- [42] KRYSL, P., AND BELYTSCHKO, T. Element-free galerkin method: Convergence of the continuous and discontinuous shape functions. *Computer Methods in Applied Mechanics and Engineering* 148, 3-4 (1997), 257–277.
- [43] LANCASTER, P., AND SALKAUSKAS, K. Surfaces generated by moving least squares methods. *Mathematics of Computation* 37, 155 (1981), 141–158.
- [44] LASCAUX, P., AND LESAINT, P. Some nonconforming nite elements for the plate bending problem. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique* 9 (1975), 9–53.
- [45] LIBERSKY, L. D., PETSCHEK, A. G., CARNEY, T. C., HIPP, J. R., AND ALLAHDADI, F. A. High strain lagrangian hydrodynamics - a 3-dimensional sph code for dynamic material response. *Journal of Computational Physics* 109, 1 (1993), 67–75.
- [46] LIPITAKIS, E. A., AND GRAVVANIS, G. A. A class of explicit preconditioned conjugate gradient methods for solving large finite element systems. *International Journal of Computer Mathematics* 44, 1-4 (1992), 189–206.

- [47] LISZKA, T., AND ORKISZ, J. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers & Structures* 11 (1980), 83–95.
- [48] LUCY, L. B. A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal* 82 (1977), 1013–1024.
- [49] MALKUS, D., AND HUGHES, T. J. R. Mixed finite element methods - reduced and selective integration techniques: a unification of concepts. *Computer Methods in Applied Mechanics and Engineering* 15 (1978), 63–81.
- [50] MAS-GALIC, S., AND P., R. A particle method for first order symmetric systems. *Numerische Mathematik* 51 (1987), 323–352.
- [51] MELENK, J. M., AND BABUSKA, I. The partition of unity finite element method: Basic theory and applications. *Computer Methods in Applied Mechanics and Engineering* 139, 1-4 (1996), 289–314.
- [52] MITCHELL, A. R., AND WAIT, R. *The finite element method in partial differential equations*. J Wiley, London ; New York, 1977.
- [53] MONAGHAN, J. J. Why particle methods work. *SIAM Journal of Scientific and Statistical Computing* 3, 4 (1982), 422.
- [54] NAYROLES, B., TOUZOT, G., AND VILLON, P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Computational Mechanics* (1992), 307–318.
- [55] NIEBUR, G., FELDSTEIN, M., YUEN, J., AND KEAVENY, T. Nonlinear finite element simulations with tissue strength assymetry accurately predict failure of trabecular bone. *Journal of Biomechanics* 33, 12 (2000), 1575–1583.
- [56] ODEN, J. T., AND CAREY, G. F. *Finite elements, mathematical aspects*. The Texas finite element series ; v. 4. Prentice-Hall, Englewood Cliffs, N.J., 1983.
- [57] ODEN, J. T., AND REDDY, J. N. *An introduction to the mathematical theory of finite elements*. Pure & Applied Mathematics. Wiley-Interscience, N.Y., 1976.
- [58] OWENS, S. *Non-Simplicial Unstructured Mesh Generation*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, 1999.
- [59] OWENS, S. J. A survey of unstructured mesh generation technology. In *Proceedings, 7th International Meshing Roundtable* (Sandia National Lab, 1998), pp. 239–267.

- [60] PESTEL, E. Dynamic stiffness matrix formulation by means of hermitian polynomials. In *Matrix methods in structural mechanics : [proceedings of the conference held at Wright-Patterson Air Force Base, Ohio, 26-28 October 1965]* (1965).
- [61] PIAN, T. Element stiffness matrices for boundary compatibility and for prescribed stresses. In *Matrix methods in structural mechanics : [proceedings of the conference held at Wright-Patterson Air Force Base, Ohio, 26-28 October 1965]* (1965).
- [62] RANDLES, P. W., AND LIBERSKY, L. D. Smoothed particle hydrodynamics: Some recent improvements and applications. *Comput Method Appl Mech Eng* 139, 1-4 (1996), 375-408.
- [63] RANDLES, P. W., AND LIBERSKY, L. D. Normalized sph with stress points. *Int J Numer Method Eng* 48, 10 (2000), 1445-U4,1453-1462.
- [64] RASHID, M., AND GULLETT, P. On a finite element method with variable element topology. *Computer Methods in Applied Mechanics and Engineering* 190 (2000). 1509-1527.
- [65] REDDY, B. D. *Introductory functional analysis : with applications to boundary value problems and finite elements*. Texts in applied mathematics ; 27. Springer, New York, 1998.
- [66] SIMO, J. C., AND ARMERO, F. Geometrically nonlinear enhanced strain mixed methods and the method of incompatible modes. *International Journal For Numerical Methods in Engineering* 33 (1992), 1413-1449.
- [67] STRANG, G. Variational crimes in the finite element method. In *The mathematical foundations of the finite element method with applications to partial differential equations* (New York,, 1972), A. K. Aziz, Ed., Academic Press, pp. xiii, 797.
- [68] STRANG, W. G., AND FIX, G. J. *An analysis of the finite element method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.
- [69] STUMMEL, F. The generalized patch test. *SIAM Journal on Numerical Analysis* 16, 3 (1979), 449-471.
- [70] SWEGLE, J. W., HICKS, D. L., AND ATTAWAY, S. W. Smoothed particle hydrodynamics stability analysis. *J Comput Phys* 116, 1 (1995), 123-134.
- [71] TAN, L. H., AND BATHE, K. J. Studies of finite element procedures - the conjugate gradient and gmres methods in adina and adina-f. *Computers & Structures* 40, 2 (1991), 441-449.

- [72] TAYLOR, R. L., BERESFORD, P. J., AND WILSON, E. L. A nonconforming element for stress analysis. *International Journal For Numerical Methods in Engineering* 10 (1976), 1211–1219.
- [73] TREFETHEN, L. N., AND BAU, D. *Numerical linear algebra*. Society for Industrial and Applied Mathematics, Philadelphia, 1997.
- [74] TURNER, M., CLOUGH, R., MARTIN, H., AND TOPP, L. Stiffness and deflection analysis of complex structures. *Journal of the aerospace sciences* 23 (1956), 805–823.
- [75] VEUBEKE, B. F. D. Variational principles and the patch test. *International Journal for Numerical Methods in Engineering* 8 (1974), 783–801.
- [76] WHEELER, M. F., AND YOTOV, I. Multigrid on the interface for mortar mixed finite element methods for elliptic problems. *Computer Methods in Applied Mechanics and Engineering* 184, 2-4 (2000), 287–302.
- [77] WILSON, E. L., TAYLOR, R. L., DOHERTY, W. P., AND GHABOSSI, J. Incompatible displacement models. In *Numerical and computer models in structural mechanics*.
- [78] ZHENG, Y., LEWIS, R. W., AND GETHIN, D. T. Three-dimensional unstructured mesh generation .1. fundamental aspects of triangulation and point creation. *Computer Methods in Applied Mechanics and Engineering* 134, 3-4 (1996), 249–268.
- [79] ZIENKIEWICZ, O. C., AND TAYLOR, R. L. *The finite element method*, 4th ed., vol. I. McGraw-Hill, London ; New York, 1989.
- [80] ZIENKIEWICZ, O. C., AND TAYLOR, R. L. *The finite element method*, 4th ed., vol. II. McGraw-Hill, London ; New York, 1991.
- [81] ZIENKIEWICZ, O. C., AND TAYLOR, R. L. The finite element patch test revisited - a computer test for convergence, validation and error estimates. *Computer Methods in Applied Mechanics and Engineering* 149, 1-4 (1997), 223–254.