

Polyhedral Finite - Element Approximants in 3D Solid Mechanics

By

MILI SELIMOTIĆ

B.S. (University "Dzemail Bijedic" in Mostar, Yugoslavia) 1991

M.S. (University of California, Davis, CA) 2004

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

CIVIL AND ENVIRONMENTAL ENGINEERING

in the

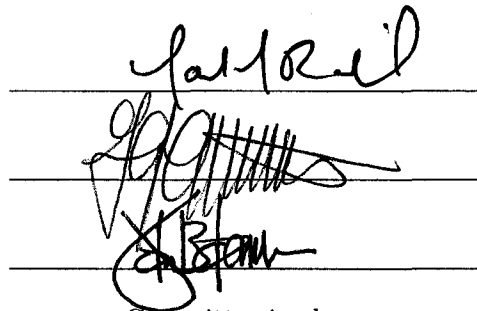
OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

Three handwritten signatures are written on three horizontal lines. The top signature is 'J. Selimotic'. The middle signature is 'J. G. [unclear]'. The bottom signature is 'J. B. [unclear]'.

Committee in charge

2008

UMI Number: 3329665

INFORMATION TO USERS

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 bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send 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.

UMI[®]

UMI Microform 3329665

Copyright 2008 by ProQuest LLC.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 E. Eisenhower Parkway
PO Box 1346
Ann Arbor, MI 48106-1346

Dedication

To my parents,
my sons, Berin and Daren,
my aunt Gara,
and my City - Mostar.

Contents

1	Introduction and Review of Galerkin Approximation Methods in Solid Mechanics	1
1.1	The conventional Finite Element Method (FEM)	2
1.2	Toward geometrically flexible finite elements	3
1.3	Meshless (Particle) Methods	6
1.3.1	General characteristics	6
1.3.2	Types of meshless methods	8
1.3.3	Disadvantages of meshless methods	13
1.4	The Natural Element Method (NEM)	18
1.5	Meshless Finite Element Method	20
1.6	The Generalized Finite Element Method (GFEM)	22
1.7	The Extended Finite Element Method (XFEM)	25
1.8	Enforcement of the Incompressibility Condition	27
1.9	Proposed Alternative Computational Approaches	29
1.9.1	The Variable Element Topology Finite Element Method (VETFEM)	29
1.9.2	The Discrete Data Polyhedral FEM (DDPFEM)	30
1.10	Dissertation organization	31
2	Finite Element Formulation	33
2.1	Continuum Mechanics Boundary Value Problem Formulation	34
2.2	Weak Form of Equilibrium and Discretization	35
2.3	The Conventional FEM and Galerkin Approach	38
2.4	Main Characteristics and Constraints of the Conventional FEM	40
2.5	Introduction to the VETFEM	44
3	3D Element Formulation of a Finite Element Method with General Polyhedral Element Geometry	46
3.1	Introduction	46
3.2	VETFEM Element Formulation	48
3.2.1	Shape Functions	48
3.2.2	Determination of Shape Function Coefficients	49

3.2.3	Shape Function Smoothness Optimization	55
3.2.4	Numerical Evaluation of Integrals of Monomial Terms	56
3.2.5	Integration Rule	58
3.2.6	Kinematic Enhancement	60
3.2.7	Element Level Algorithm	63
3.2.8	Convergence of the VETFEM	64
3.3	Computational Examples	66
3.3.1	Patch Tests	67
3.3.2	Experimental Analysis of the Rate of Convergence	71
3.3.3	3D Elastic-Plastic Problem	75
4	A Discrete Data Polyhedral Finite Element Method (DDPFEM)	78
4.1	Introduction	78
4.2	Constraints and Interior Smoothness Optimization	81
4.3	Integration Rule	85
5	Future Work	91
	Bibliography	95

List of Figures

2.1	Motion of the body relative to a stationary Cartesian coordinate system	34
2.2	Isoparametric coordinate mapping	41
3.1	Geometrical illustration of integrands in equation (3.7).	52
3.2	Piecewise linear function f_{ab} on the facet boundary $\partial\Gamma_b$	53
3.3	Placement of integration points for 3D VETFEM elements	59
3.4	Domain Patched with Eight VETFEM Elements	68
3.5	Domain Patched with Fifteen VETFEM Elements	69
3.6	Domain Patched with Four Voronoi Cells	70
3.7	Domain Patched with Six Voronoi Cells	70
3.8	1600 element VET mesh of a three-dimensional linearly elastic domain. The body was subjected to a uniform normal traction on its top surface with its bottom surface fixed.	73
3.9	Log-log plots of error in displacement magnitude and in vertical-direction normal stress at the points indicated in Figure 3.8. At each of 5 levels of refinement, 4 different VET analysis were performed and compared with the reference solution.	74
3.10	Octant of the Cube Patched with VET and hex elements	77
3.11	Normalized Major- and Minor- axis Deformation vs. Load Level	77

List of Tables

3.1	Distribution of element geometries for the VET mesh shown in Figure (3.10).	76
-----	---	----

Polyhedral Finite - Element Approximants in 3D Solid Mechanics

Abstract

Two novel finite-element-type based approximation schemes for solid-mechanics applications are presented in a three dimensional setting. The first, the Variable Element Topology Finite Element Method (VETFEM) retains all the powerful characteristics of the conventional Finite Element Method (FEM), while providing a level of simplicity and flexibility in the spatial discretization of the domain that compares to that of particle methods. The VETFEM employs elements that can take the form of general polyhedra, which are subject only to mild geometric restrictions in contrast to the conventional FEM. As such, the VETFEM can be especially useful for problems that involve complex geometry, adaptive remeshing or crack propagation.

The second novel method presented here, the Discrete Data Polyhedral Finite Element Method (DDPFEM), unlike other finite-element-based methods, does not define the shape functions pointwise on the element domain. Instead, the DDPFEM formulation involves values of the shape functions at the element nodes, and values of their gradients at the element integration points only. The method extends the applicability of arbitrary polyhedral elements to spatial discretizations that admit a greater degree of concavity in the elements, compared to VETFEM elements.

Convergence of the VETFEM is discussed and analyzed through an experimental analysis, and overall behavior observed through computational examples, including patch tests.

Chapter 1

Introduction and Review of Galerkin Approximation Methods in Solid Mechanics

A typical second order, elliptic boundary value problem in solid mechanics can be mathematically described by partial differential equations (PDE) which, in most cases, cannot be solved exactly. The need to analyze and solve everyday engineering problems has therefore resulted in a wide range of numerical approximation techniques that can be used to numerically model these problems and facilitate their solution. Over the last several decades, there has been a growing tendency to use model-based simulation to analyze different materials and their use in various engineering applications. Coupled together with laboratory testing, model-based simulation can provide new insights into

the behavior of materials under different loading and boundary conditions (i.e. propagation of cracks, existing or introduced). Model-based simulation is a process that, in general, consists of model construction (which involves the theoretical justification for the problem formulation), domain discretization (including the manner of specifying boundary conditions), numerical analysis, and interpretation of results. This process allows for progressively improved analysis and design of materials and structures through continuous refinement of the numerical model and adjustment of the design parameters. Advances in computer technology have decreased the computational effort to solve the resulting systems of equations, making model-based simulation even more attractive and prevalent across all sectors of engineering. The use of robust methodologies and algorithms can further decrease simulation runtime even while enlarging the range of problem types that can be addressed, eliminating computational cost as the overriding concern in the analysis of large problems. Modern efforts are also consistently directed towards automating the process of domain discretization and mesh regeneration, and to visual interpretation of the solution.

1.1 The conventional Finite Element Method (FEM)

The conventional, displacement based Finite Element Method (FEM) is currently the leading computational tool for solving problems in solid mechanics. Powerful features of the conventional FEM include an accurate and explicit definition of the boundary of the domain; convenient application of boundary conditions; a sparse and banded

global equation structure; and a high level of accuracy relative to the computational effort. These powerful properties of the conventional FEM derive, in large part, from the isoparametric formulation of standard elements on a simple parent domain. However, this type of element formulation also imposes a strict set of rules on the generation of suitable meshes. If proper meshes can be constructed, however, the conventional FEM can handle arbitrary geometry and boundary conditions, and constitutive models. Furthermore, a rich mathematical framework exists to address convergence and error estimation strategies. However, problems involving frequent remeshing, e.g. extreme deformations or crack extension, require significant effort, especially in human terms. In the conventional FEM, a mesh is used to construct the trial space for approximating the solution, as well as to perform quadrature. The quality of the approximation depends crucially on the properties of the trial space and, on the accuracy of the numerical quadrature.

1.2 Toward geometrically flexible finite elements

In the general case, FEM meshes are unstructured, and therefore suited for problems with complex geometry and material heterogeneity. Preferred element types are quadrilateral (in 2D) and hexagonal (in 3D) elements. However, many FEM approaches rely on triangular and tetrahedral elements in 2D and 3D respectively in problems involving complex geometries. These types of elements are sometimes used extensively, primarily to avoid formation of poorly shaped elements, which leads to solution errors.

The treatment of transitions between mesh blocks of differing density represents an area of considerable interest. In [34] and [35], a “uniform strain approach” was used to improve the standard master and slave boundary methodology in order to connect dissimilar finite element meshes in 2D and 3D, respectively. In [60], a formulation of multi-nodal elements that can be used in transition regions was described, extending the work presented in [42] and [17] to 2D polygonal elements containing more than four nodes. The element formulations proposed in these contributions are constant strain elements that use one-point quadrature, and are therefore appealing from the point of view of computational efficiency. However, constant strain elements that have more than three nodes (four nodes in 3D) experience force-free displacement modes. To eliminate these spurious modes, some form of mode stabilization must be applied in order to address rank deficiency in the global stiffness matrix. To prevent singularities in the global stiffness matrix, element stiffness matrices are corrected for elements having more than three nodes by calculating and adding a stabilization matrix that accounts for the lack of flexural stiffness in the constant gradient assumption.

In a different approach described in [81], an isoparametric transformation is used where parent elements are regular polygons and physical elements are generated as Voronoi cells. Natural neighbor shape functions (Sibsonian and Laplace) are defined on the parent element whose nodes lie on the circumcircle of the polygon, and are therefore natural neighbors to any point inside the parent element. Since the mapping is affine, the shape functions remain linear on the boundary of a distorted (but convex) physical

element, ensuring C^0 continuity on the domain Ω and, therefore, the conformity of the method. The method showed notably better accuracy than in the Natural Element Method (NEM) and in most other meshless methods, due to its conformity and the fact that the shape function supports coincide with the integration domains. The method is also less sensitive to locking with incompressible materials than the conventional FEM. This approach can be extended to polyhedral meshes in 3D [49], [48] where Laplace C^0 shape functions were used on polyhedral elements in physical coordinates. However, the nodes of these polyhedra have to be located on the same circumsphere, which represents a significant geometric restriction. (By comparison, this kind of geometric restriction does not apply to the Variable Element Topology Finite Element Method (VETFEM) described later in this dissertation). At the time of the development of the method, the evaluation of the weak form integrals over Voronoi cells was not completely developed, and two possible ways to perform numerical integration were suggested: a subdivision of either parent or physical element into triangles and applying well known Gauss quadrature rules, or applying cubature rules on the canonical element as a regular n-gon. The VETFEM, described later, resolves this issue by constructing an efficient quadrature rule for placement of the integration points within arbitrary polyhedra while assigning weights to each integration point in the element [65], [66], [67].

1.3 Meshless (Particle) Methods

Meshing is an expensive and extensive effort that includes grid generation, mesh adaptation to the domain geometry, element (cell) connectivity, grid motion and separation (to model fracture), and other aspects. For complex domains, this effort can become the major part of the expense of an analysis, and can require several iterations. Adaptive remeshing procedures for simulations of impact/ penetration problems, explosion/ fragmentation problems, fluid-structure interaction problems, etc. are likewise very demanding. The difficulties associated with these tasks also include mapping of the state variables from the old mesh to the new mesh. This process usually introduces numerical errors. It is for these reasons that the mesh forming rules of the conventional FEM impose significant burdens on human and computational resources. These complexities have led many researchers to pursue alternative approximation procedures that do not require a finite-element-like spatial discretization of the problem domain. The result of these efforts was the development of particle (meshless) methods [5], [49].

1.3.1 General characteristics

The idea of meshless Galerkin methods first emerged in the early 1990s. Meshless methods do not use a mesh, or use it minimally for numerical integration purposes, with weaker requirements on the mesh topology. Many different names are used for meshless methods in the literature, but they differ mostly in their method of shape function construction. Regular particle distributions are constructed using particles inside the

domain Ω , but the support for some of the particle shape functions could be partially outside of Ω . The domain of influence of a node (i.e. the support of the associated shape function) is represented by a sub-domain over which the subject function is non-zero, and whose size is relatively small comparing to the rest of the domain. The most commonly used subdomains are discs (2D) or balls (3D). The overlap between the shape function supports can be large, which makes these shape functions almost linearly dependent. The size of the support of these particle shape functions plays an important role, since the bandwidth of the global linear system of equations depends on it. It is shown in [5] that if shape functions possess certain properties, they will have good approximation properties as well. Since a wide variety of particle shape functions can be constructed, it is important to properly select an appropriate class of shape functions for the application. The best choice will have the smallest value of the usual Sobolev norm interpolation error. Different particle shape functions include smooth particle hydrodynamics functions for problems in fluid dynamics; moving least squares shape functions in the context of data fitting with respect to irregularly disturbed particles in higher dimensions; translation-invariant reproducing kernel particle shape functions in the context of solutions to partial differential equations, etc.. These functions depend on the choice of a weighting function, where different weight function will generate a different class of such functions.

There are two basic methods of discretization that generate meshless methods. The collocation method is used in Smooth Particle Hydrodynamics (SPH) to discretize the

strong form of a partial differential equation [53] directly. Here, the discrete equations are not applied on boundary nodes; the equations are enforced on the interior nodes only. The second general discretization approach is the Galerkin method that is used in the Element Free Galerkin Method (EFGM), Partition of Unity Finite Element Method (PUFEM), Reproducing Kernel Particle Method (RKPM), and hp clouds methods. This approach requires formation of a weak or variational form of the problem first. The implementation of the Galerkin method in particle methods is similar to the conventional FEM.

1.3.2 Types of meshless methods

- The SPH is probably the oldest meshless method (Lucy 1977, Gingold and Monaghan, 1977). In 1982, Monaghan invoked the notion of a kernel approximation. The SPH appears to be one of the most flexible meshless methods, and is easy to implement. The method uses collocation at the nodes to build a discrete set of governing equations, and does not require a background mesh to perform quadrature. Numerical integration takes a simple form (i.e. trapezoidal rule in 1D). The kernel (smoothing) function is required to possess a normality property, be positive on a subdomain of Ω (Ω_I), and zero otherwise in order to enable the approximation to be generated from a local representation. Commonly used weight functions are exponential, cubic spline and quartic spline, and are constructed to possess C^2 continuity. SPH shape functions are constructed as products of the

weighting function (kernel) and some geometric measure of the subdomain (Ω_I), and, like in the majority of particle methods, are not true interpolants of nodal displacements [15]. The kernel estimation technique used in SPH is equivalent to use of Shepard interpolants (lowest order form of moving least squares shape functions) that are known for inadequate precision, leading to poor accuracy of the method and a requirement for large number of nodes in order to improve its performance. The continuous form of the SHP kernel exhibits 0^{th} (constant) and linear consistency in the case when the weighting function is symmetric about the origin. However, linear consistency of the discrete form is not achieved. Despite this shortcoming, the SPH method can often be made to provide a satisfactory solution to some second order PDEs.

- Moving least squares (MLS) approximants are constructed by multiplying polynomial basis functions with coefficients that are functions of the spatial position. Consistency of order k is achieved if the basis is complete in polynomials of order k (i.e. if the basis is linearly complete, shape functions will satisfy linear consistency). Other functions can be used as well (i.e. singular functions for problems with singular solutions). In essence, any function included in the basis can be reproduced exactly by MLS approximants. Moving least squares approximants possess high rates of convergence, and since MLS functions can be easily constructed and produce any degree of regularity, even $C^\infty(\Omega)$, methods that use these shape functions (EFGM) [16], [10] do not exhibit volumetric locking and

perform well even for an irregular placement of nodes. However, MLS functions are not non-negative and do not interpolate the data. Another difficulty arises in constructing MLS functions relating to an enlargement in the function support, which occurs if an increase in the polynomial order that the MLS functions can represent is desired. Also, some nodal arrangements exist that can break down the algorithm used to construct MLS functions. Moving least squares approximants are used in some meshless methods to construct test and trial spaces [15], [53], employing then a Galerkin method to find the best solution. Such an approach by Nayroles, Touzot, and Villon in 1992 led to the development of the Diffuse Element Method (DEM) [59], that uses a distribution of nodes and a boundary description to develop Galerkin equations. Interpolants take the form of polynomials which are fit to the nodal values by a least square approximation. Belytschko, Lu, and Gu in 1994 [16] refined and modified the method, thereby introducing the Element Free Galerkin Method (EFGM).

- The EFGM employs moving least squares approximants to construct trial and test functions for the variational problem statement [16]. Dependent variables and their derivatives are continuous on the domain, and high resolution of localized steep gradients can be obtained. The EFGM includes certain terms intended to improve the accuracy of the derivatives, and generally must employ Lagrange multipliers to enforce essential boundary conditions. The use of Lagrange multipliers complicates the solution process, since special solvers that do not take

advantage of positive definiteness must be used. This can be remedied, to some extent, by utilizing perturbed Lagrangian or penalty methods. For numerical integration, the EFGM uses a background mesh, and requires a large number of integration points to perform quadrature. In [61], an alternative approach is developed that borrows from the Arbitrary Lagrangian-Eulerian (ALE) formulation for the EFG method. This formulation allows for continuous relocation of a limited number of nodes (i.e. around the crack tip in dynamic fracture problems), that are superposed onto a coarse Lagrangian mesh within the domain. The idea is to reduce the necessity for very fine discretization that would otherwise be required. The DEM and the EFGM are consistent and stable, but more computationally expensive than either SPH or conventional FE methods. Computational costs of an implicit EFGM in 2D exceed a low-order finite element solution by a factor of 4 to 10, and even more in 3D.

- The Reproducing Kernel Particle Method (RKPM), as well as SPH, are developed using a continuous reproducing kernel approximation, whereas the DEM and EFGM, like the conventional FEM, are derived through the discrete approach. The RKPM is based on the theory of wavelets [55] in which the interpolation function is represented by a combination of the dilation and translation of a single wavelet, or window function. This numerical approach is more efficient in computer implementation than the DEM and EFGM. The RKPM is generally regarded as representing an improvement comparing to the SPH kernel method.

This improvement results from the addition of the boundary correction term to the reproducing kernel approximation. Since the correction function and window (weight) function can be chosen to be smooth, the solution and its derivatives are continuous throughout the entire domain, unlike in the conventional FEM.

- The Partition of Unity Finite Element Method (PUFEM) subdivides the domain Ω into overlapping subdomains Ω_I , each associated with function $\Phi_I(\mathbf{x})$ that possesses the properties where $\Phi_I(\mathbf{x}) \neq 0$ only on the subdomain Ω_I , and where $\sum_I \Phi_I(\mathbf{x}) = 1$ on the domain Ω . Since the partition of unity property is identical to the order-zero consistency condition, all bases that are useful in Galerkin methods must be partitions of unity. Duarte and Oden in [41] used the partition of unity concept more generally, and in conjunction with moving least squares shape functions, to obtain the so-called hp clouds method. To improve conditioning, they used higher order terms employing higher order Legendre polynomials in the basis. The PUFEM shape functions that are based on Lagrange polynomials additionally satisfy the Kronecker-delta property. However, the extension to multidimensional problems has difficulties, because the construction of the Lagrange interpolant requires an ordering of nodes in all directions. The PUFEM can be made to include a priori known analytical information about the local behavior of the solution in finite element space [56]. The PUFEM constructs a global conforming finite element space out of given local approximation spaces, separating the issues of interelement continuity and local approximability. As such, the method is

especially useful for problems in which the usual finite element polynomial spaces do not offer an acceptable solution unless the mesh size parameter is very small, or the polynomial degree is very large, leading to high computational costs in either case. As with the majority of particle methods, the absence of a mesh and an inability to tie shape functions with an underlying mesh results in issues relating to numerical integration. Efficient imposition of essential boundary conditions is also an issue. The results of the method also greatly depend on the choice of the basis of the PUFEM space. It can be said that the partition of unity concept underlies moving least squares and kernel approximations, while providing a greater degree of flexibility. As with MLS approximants, a considerable disadvantage of the PUFEM is the computational time associated with calculating shape functions and their derivatives.

1.3.3 Disadvantages of meshless methods

The two main weaknesses of meshless methods [49] that impact their applicability, especially to three dimensional problems, are difficulty with essential boundary conditions, and numerical integration.

- *Imposition of essential boundary conditions :*

Since particle shape functions do not satisfy the Kronecker-delta property, the introduction of essential (Dirichlet) boundary conditions [49], [5] presents a major issue associated with meshless formulations. Different approaches have been taken

in order to deal with this problem in meshless methods [5]. For example, the main idea of the penalty method is to perturb the variational principle. Another method is the Lagrange multiplier method, first applied in this context by Babuska in 1973. This method depends on the relationship between the approximating space and the space of Lagrange multipliers, and leads to better rates of convergence compared to the penalty method. Here, a reaction force is introduced on the displacement boundary Γ_u that is complementary to the unknown displacement u . This method is the most accurate for imposing Dirichlet boundary conditions, and is useful for small problems where cost is immaterial. However, the discrete equations are no longer positive definite, and a stability problem arises if shape functions are not carefully chosen. The collocation method adds a constraint equation at certain points of the boundary $\partial\Omega$ to the stiffness matrix. Combinations of meshless methods and the conventional FEM have also been proposed, which use classical finite elements in the neighborhood of the boundary $\partial\Omega$ and particle shape functions whose supports do not intersect the boundary [50]. This approach is particularly appealing where a finite-element-like mesh is used for quadrature anyway.

The Displacement Constraint Equations (DCE) method [87] is presented as an exact method to impose essential boundary conditions. Boundary conditions are imposed as a displacement constraint on the discrete equations resulting from the Galerkin approach in meshless methods. The unknowns are split into two

sets, those on the boundary and the remaining unknowns, splitting all matrices in $\mathbf{KU} = \mathbf{F}$ into two. There are other methods that permit a direct imposition of essential boundary conditions in meshless methods allowing the approximation of the displacement field to be kinematically admissible, such as: the Nitsche and related methods, the characteristic function method, transformation methods [25] and boundary singular kernel method [26].

- *Numerical integration and volumetric locking :*

In meshless methods, linear completeness is achieved, and if a method with linearly complete approximants does not pass the patch test, the reason is likely to be in the manner that the numerical integration is performed. Numerical integration is a key issue in ensuring the convergence of Galerkin methods. Construction of the stiffness matrix and the load vector is where meshless methods pay the price for the absence of the mesh. The problem arises because the supports of the basis functions do not correspond with the integration cells, which can be either regular or finite-element-like. Instead, basis function supports generally overlap. An additional issue is that the shape functions are usually not polynomials, but rational functions where Gauss quadrature is not sufficiently accurate. Different techniques have been developed and proposed to address the quadrature issue, and to analyze the error of Gauss integration in Galerkin meshless methods. Gauss quadrature in Galerkin meshless methods adds a considerable complexity to the solution process and affects the accuracy and convergence, while direct nodal in-

tegration, although computationally more efficient, generally leads to numerical instabilities due to vanishing shape function derivatives at nodes. It also can exhibit poor accuracy. The most significant achievement of nodal integration (SPH, EFGM) is that it eliminates the need for a background mesh, making the method truly meshless.

Dolbow and Belytschko [37] studied the error incurred with Gauss integration on a background mesh, and the fact that integration rules require many Gauss integration points to adequately evaluate the weak form [49]. They proposed a meshing approach to minimize the integration error, and showed that a significant error is produced when integration regions do not correspond to shape function supports. Randles et al. [62] proposed a stress point method to improve quadrature in Smooth Particle Hydrodynamics with collocation by introducing two sets of points for domain discretization, with the intent of calculating derivatives away from locations where shape functions have zero derivatives. In [20], a way to improve the accuracy of nodal integration, namely by calculating coefficients of correction terms at discrete nodes that satisfy a linear patch test condition, was presented. In [10], a stabilized EFGM is presented to address the problem of spatial instability (formation of spurious, zero-energy modes of deformation) that are caused by underintegration of the weak form in the EFGM. The proposed method adds the square of the residual in the equilibrium equations to the potential energy functional as a stabilization term. The stabilized EFGM eliminates near-singular

modes in many cases, and achieves a reasonable rate of convergence. In [38], a selective reduced integration scheme for the EFGM was derived in order to eliminate volumetric locking. In this formulation, the deviatoric part of the weak form is integrated with full quadrature, while the dilatational part is integrated with nodal quadrature, where nodal quadrature weights are obtained from the Voronoi cell volumes corresponding to the tessellation of the grid. A stabilized nodal integration scheme in [27] improves efficiency, accuracy and convergence by introducing strain smoothing via an assumed strain method. The requirement that a linear displacement vector exactly satisfies the discrete equilibrium equations is imposed first as a set of integration constraints necessary to enforce essential boundary conditions. The strain smoothing, while satisfying the integration constraints, does not evaluate derivatives of the shape functions at nodes, and therefore eliminates spurious modes.

In summary, it can be said that the greatest strength and at the same time the greatest weakness of particle methods is that they do not involve any explicit connectivity among the nodes. On the one hand, there are no topological or geometric constraints of the kind exhibited by the conventional FEM, but on the other hand, this feature results in awkward influence of the boundary, complicated imposition of boundary conditions, and typically a much denser global system of algebraic equations than for similar FEM calculations. Effective implementation of meshless methods is the key to their success. It can be said that these methods still require considerable further development before they

can match the broad applicability of the conventional FEM in computational mechanics. The enhancement of the speed and robustness of particle methods represent the greatest challenges in their future development.

The following sections outline numerical approaches that, in certain aspects, differ from “classical” particle methods as well as from the conventional FEM and, therefore, fit somewhere in between. These methods are presented here in more detail.

1.4 The Natural Element Method (NEM)

The NEM [80], [86],[45] is another Galerkin computational scheme for solving elliptic PDEs, where both trial and test function spaces are constructed using natural neighbors interpolants that are smooth (C^∞) everywhere except at the nodes, where they are C^0 continuous. A displacement based Galerkin procedure can be implemented to obtain a discrete system of linear equations. The notion of natural neighbors was first introduced for data fitting and smoothing purposes, where the concept of second order Voronoi cells was used to introduce natural neighbors and natural neighbor coordinates. Construction of the interpolant is based on the Voronoi diagram (cells) that represents a unique discretization of the domain, in contrast to Deluney Tessellation, which is non-unique in degenerate cases, and is otherwise sensitive to geometric perturbations of nodal locations. The main difference between the NEM and other meshless methods is that the NEM constructs interpolants, while other meshless methods use approximants that do not possess the Kronecker-delta property. The method employs mainly two

algorithms to compute the natural neighbor interpolant: Sibsonian and non-Sibsonian (Laplace). Sibson shape functions are defined as the ratio of the area (volume) overlap of intersecting neighboring regions (Voronoi cells). Determination of these intersections is a known problem in computational geometry, but still a practical issue today. Non-Sibsonian shape functions require computing the Lebesgue measure of one order order less than the Sibson interpolant ($d - 1$ in \mathbf{R}^d), and therefore appears to be more computational friendly and easier to implement. Interpolants that are a linear combination of Sibsonian and non-Sibsonian interpolants also represent valid natural neighbor interpolants. The advantage of the non-Sibsonian interpolant is its computational efficiency and its ability to impose essential boundary conditions in a FEM-like manner on convex and non-convex boundaries. Since the shape functions form a partition of unity, the enrichment of the NEM trial functions can be achieved and used to analyze problems such as crack propagation, where knowledge of local behavior exists [79].

As in other meshless methods, evaluation of the weak form integrals in the NEM is an area of concern, and the problem of numerical integration is still not completely solved. The method typically uses Gauss quadrature with a background cell structure (mesh). The support of the shape functions is not the same as this background quadrature mesh (Delunay triangles in 2D). In order to remedy this problem, different approaches have been attempted. A so-called “local” approach in [45] is based on the decomposition of the nodal shape function supports into simpler geometrical regions (triangles that use Gaussian integration and isoparametric transformation of circular segments onto a

unit square are applied). This approach requires a large number of integration points in order to achieve reasonable results, which adversely affects computational cost. In [86], a stabilized, conforming nodal integration scheme with strain smoothing is proposed to address the issue of numerical integration, control instabilities, and improve the convergence. Nodal quadrature does not use background cells for integration, which leads to poor results in meshless Galerkin methods primarily because the shape function derivatives at the nodes are zero in most meshless methods, or are undefined as in the NEM. This approach is based on an assumption of a modified strain field at each node, and the enforcement of an integration constraint, i.e. satisfaction of discrete equilibrium equations when a linear displacement field is imposed. The method shows better results, and is especially compatible with the NEM since most of the computational features are related to Voronoi diagrams, which are already computed to construct the shape functions, making this technique more computationally efficient. However, the behavior of the three dimensional NEM with respect to numerical integration over tetrahedra has been seen to be similar to the two dimension case, and numerical experiments do not pass the patch test [30]. Coupling of the NEM with the conventional FEM is simple [31] and without the need for a blending domain or modified shape functions.

1.5 Meshless Finite Element Method

The notion of the meshless FEM was introduced to bridge the gap between meshless methods and the FEM. In [49], true meshless methods are defined as numerical

approaches where shape functions depend only on the nodal position, and where the connectivity of nodes depends exclusively on the total number of nodes. The conventional FEM is generalized to satisfy the above mentioned two criteria characterizing meshless methods. Standard Delauney tessellation makes meshless methods that use it sensitive to the nodal geometry, and can lead to creation of tetrahedra (in 3D) with zero volume. In [49], the extended Delauney tessellation is proposed that eliminates this problem and creates unique polyhedra with all nodes laying on the same Voronoi sphere, along with tetrahedra on the rest of the domain. This leads to a unique domain partition for a given nodal distribution, and is realized in bounded time. In order to define shape functions with strict continuity between neighboring polyhedra, the non-Sibsonian interpolation is used. Non-Sibsonian shape functions possess some FEM-like properties, including the Kronecker-delta property, and their supports are the natural neighborhood of the respective node. In comparison to this method, where polynomial shape functions on polyhedral elements can be used only for some special cases, the VETFEM demonstrates that polynomial shape functions can be used on polyhedral elements of any shape while still satisfying weak continuity, leading to the convergence of the method.

Numerical integration in the meshless FEM is performed on tetrahedra using only one Gauss point for evaluating integrals. Polyhedral elements are subdivided into tetrahedra for quadrature purposes only, and one Gauss point per tetrahedra is used. The imposition of boundary conditions is improved compared to other meshless methods,

due to the Kronecker-delta property of the non-Sibsonian shape functions. However, the boundary definition is still a problem and has an error comparable with other meshless methods. The method has improved the evaluation of integrals and offers more stable and smooth solutions compared to other meshless methods, as well as easier mesh generation compared to the conventional FEM.

1.6 The Generalized Finite Element Method (GFEM)

Polynomial functions have good approximation properties, but for certain classes of problems where the solution is not smooth, the need arises for different, more effective classes of functions that are not polynomials. One of the objectives in variational meshless methods is to be flexible and allow the inclusion of special “handbook” functions in trial and test spaces [5]. However, a high order of shape function continuity (when shape functions are not polynomials) decreases the convergence rate, which worsens with introduction of discontinuities [49]. The GFEM represents an extension of the conventional FEM, first proposed by Babuska (1994) for elliptic problems where the conventional FEM shows arbitrarily slow rates of convergence. The method was later called the partition of unity method, particle - partition of unity method, method of finite spheres, cloud method, extended FEM and GFEM.

The GFEM is based on the partition of unity concept and can be effected by a simple (i.e. uniform) mesh with “hat” functions to complete the partition. The GFEM expands the approximation space by including functions that are obtained either ana-

lytically or numerically using knowledge about the properties of the localized features of the solution within the problem domain such as voids, cracks, etc., and as such offers certain advantages compared to the conventional FEM. These special “handbook” functions with local support can then be multiplied with “hat” functions to construct conforming approximations and to improve the accuracy and convergence [75]. For partition of unity in the GFEM, particle shape functions can be used, and “handbook” functions might be employed in order to satisfy Dirichlet boundary conditions, avoiding, therefore, the need to use penalty or other methods to enforce essential boundary conditions. Shape functions created in this way belong to the Hilbert space $H_1(\Omega)$ and can therefore be used in a weak form method.

While the conventional FEM employs only one mesh to construct the interpolant and apply quadrature, the GFEM utilizes two meshes, the approximation mesh used to construct basis functions, and an integration mesh to evaluate weak form integrals. The approximation mesh exhibits wide flexibility, in the sense that it might be partially or completely independent of the domain, and can arbitrarily overlap the domain boundaries. The integration mesh, for example constructed using the Fast Meshing Algorithm [76], is applied to each element of the approximation mesh independently in order to account for the domain geometry, and to apply quadrature. As in any numerical method, an accurate and effective quadrature scheme to approximate the elements of the stiffness matrix is essential for the success of the method. Numerical integration represents a problem, especially when integrating over the areas that are

not simple triangles (tetrahedrons in 3D), and/or when singular functions are included in local approximation spaces. In these cases, standard quadrature schemes yield poor results. Even though the algorithm is shown to be robust, a rather complex set of rules is involved in subdivision of cells, and since the method was applied to 2D problems, it would be expected that these rules expand significantly for 3D settings. Also, system of linear equations may be singular and has a non-unique solution except under certain circumstances [5].

The enrichment of the approximation by known functions does improve results, as expected. The method achieves its desired goals in its flexibility to construct trial spaces, while allowing the approximant to use all available information about the solution. The main claim of the method is that it can be easily applied on domains where the use of the conventional FEM is basically prevented by topological and geometric constraints on the FEM mesh (enormous number of cracks, holes, corners, etc.). The GFEM, as well as meshless methods, allow use of non-polynomial and non-smooth shape functions, which makes them more suitable for certain types of problems than the conventional FEM. As shown later in this dissertation, the VETFEM addresses these shortcomings of the conventional FEM and provides suitable meshes and an effective, much less involved integration rule without the need for a separate (integration) mesh.

1.7 The Extended Finite Element Method (XFEM)

The XFEM has its origins in the work of N. Moes [58] relating to the use of step functions to model material discontinuities. The XFEM can be regarded as a particular instance of the PUFEM or GFEM, where the finite element mesh does not need to conform to the internal boundaries (cracks, material interfaces, voids, etc.) and hence a single mesh suffices for modeling. The main advantage of the method over meshless methods is that the sparsity and symmetry of the stiffness matrix, characteristic for the conventional FEM, is preserved.

In [58], for problems related to crack propagation, a methodology was developed that allows the crack to be represented independently of the mesh, avoiding the necessity for remeshing to model crack extension. Here, a standard displacement based approximation was enriched near the crack tip by incorporating known near-tip asymptotic fields. Discontinuous fields away from the tip were introduced through a partition of unity, which facilitates incorporation of the enrichment functions into the finite element approximation. The quadrature scheme used subdivision of the elements cut by the crack into sub-elements whose boundaries align with the crack geometry, and with the requirement that the integration is applied to both sides of the discontinuity for nodes enriched with the discontinuous function. Stress intensity factors computed in this way showed a satisfactory accuracy, even when a relatively coarse mesh was used. However, a small drawback is that the method introduces additional degrees of freedom associated with shape functions on the subset of nodes that include the enrichment,

introducing therefore a greater computational effort.

In the XFEM, issues with computational geometry are actively present, and burdensome in many aspects (determining the cut elements, subdivision of elements cut by a discontinuity, etc.). In [78], a numerical integration scheme was presented that uses Gauss-Legendre quadrature. Hexahedral elements with finite element shape functions and without enrichment were integrated using standard $2 \times 2 \times 2$ quadrature, while the elements with enriched degrees of freedom used a $6 \times 6 \times 6$ integration rule. In [39], in order to apply quadrature, the crack discontinuity is subdivided into segments to account for the interaction between the crack and mesh. This illustrates the computational effort to resolve intersections between the discontinuity and elements cut by the crack.

In [74], a coupling of the Level Set Method (LSM) and the XFEM was proposed. The LSM, as a numerical technique for tracking the motion of interfaces, is used to represent the crack location, facilitate determination of nodes to be enriched, and assist in defining enrichment functions, while the XFEM is employed to compute stresses and displacement fields in order to determine the rate of the crack growth. In order to avoid difficulties associated with branch functions used for the near-the-tip enrichment in the tip element, the parent domain of the partially cracked tip element is divided into two parts: the cracked and the uncracked part. Only the former part is enriched by the sign function ($sign(\mathbf{x})$) [88]. The sign function was used instead of the step function because of its symmetry. The treatment of the crack tip involves the use of asymptotic near-tip fields, which require care in integration. The matching of these asymptotic

fields with the step function used to model the remainder of the discontinuity can also be problematic. In [52], a methodology involving superimposed quarter point elements (overlaid patch) is proposed to address the above mentioned problems in the vicinity of the tip, while the remainder of the discontinuity treated by the XFEM.

The XFEM, as an extension of the conventional FEM, while employing certain enrichments in localized areas of the domain, still depends on the isoparametric transformations from the parent domain, and therefore inherits issues related to this mapping that exist with the conventional FEM. The VETFEM, however, precludes the need for such mapping, and provides a desirable flexibility and an efficient generation of suitable polyhedral meshes that conform exactly to the problem geometry.

1.8 Enforcement of the Incompressibility Condition

It has long been observed that low order, displacement based finite elements behave poorly near the incompressible limit, which plays an important role in real world engineering problems. So-called volumetric locking causes a deterioration in both the accuracy and rates of convergence of the solution [8], [82]. Locking is exhibited by standard bi-linear and tri-linear elements near the incompressible limit, and occurs as a result of the inability of low-order interpolants (polynomials) to adequately represent general volume preserving displacement fields. However, low-order polynomials are generally preferred for other reasons, particularly for large deformation problems. There exist various finite element formulations for effective use near the incompressible limit

[72]. These approaches include the use of higher order elements [82], and use of mixed methods in which different approximations are implemented for the displacement and pressure fields [47].

For non-linear problems, the enforcement of incompressibility is even more demanding and development of such low-order polynomial elements is not an easy task [71]. In [33], four node quadrilateral and eight node hexahedral elements that are based on splitting the deformation gradient \mathbf{F} are proposed to improve the behavior of the FEM near the incompressible limit, and especially for strain localization problems, metal forming simulations, etc.. This element formulation is based on the concept of multiplicative deviatoric (volume preserving) and volumetric (dilatational) split of the deformation gradient as $\mathbf{F} = \mathbf{F}_d \mathbf{F}_v$, in the conjunction with replacement of the compatible \mathbf{F} with an assumed modified counterpart \mathbf{F}_m . The deformation gradient \mathbf{F} at the integration point of interest is split first into its deviatoric and volumetric parts, as is the deformation gradient at the element centroid \mathbf{F}_0 . Specifically, $\mathbf{F}_v \equiv (\det \mathbf{F})^{1/3} \mathbf{1}$, and $\mathbf{F}_d \equiv (\det \mathbf{F})^{-1/3} \mathbf{F}$. The modified gradient \mathbf{F}_m is then composed of the deviatoric part of \mathbf{F} and the volumetric part of \mathbf{F}_0 . This modified deformation gradient is then used to compute stresses at the integration point of interest. In this formulation, the strain driven format of algorithms for integration of inelastic constitutive equations is maintained, so that the formulation can be easily incorporated into existing codes. In comparison with conventional kinematic formulations, some extra operations are required by this approach. For computing the internal force vector, the only difference

with respect to the conventional elements is the replacement of \mathbf{F} with \mathbf{F}_m , which is minimal from the computational standpoint. However, calculating the tangent stiffness matrix requires determining an additional part of the matrix that is generally unsymmetric and requires an unsymmetric solver to be used at the global level.

1.9 Proposed Alternative Computational Approaches

1.9.1 The Variable Element Topology Finite Element Method (VET-FEM)

The VETFEM was first introduced by Rashid and Gullett (2000, 2002) [65], [66] in a 2D setting and represents an alternative to the conventional FEM, and its extensions (GFEM, XFEM), as well as to particle methods. The VETFEM possesses all of the powerful features of the conventional FEM and, at the same time, approaches meshless methods with respect to its flexibility in the spatial discretization of the domain. The VETFEM eliminates the strict set of rules on the element geometry and topology imposed by the conventional FEM by using elements that can be general polyhedra, and can have any number of nodes in any arrangement. The VETFEM 3D element formulation represents one of the main contributions of this dissertation.

The advantage of the VETFEM compared to the conventional FEM arises from the manner in which the basis functions are constructed. The conventional FEM basis functions are constructed by employing the isoparametric transformation of canonical

shape functions defined over a simple parent element. This transformation, however, requires that a fairly strict set of topological and geometric restrictions be imposed on the elements in the mesh. VETFEM elements instead construct shape functions directly in the physical coordinates of the element, in the form of low-order polynomials, and do not require any isoparametric transformation. Through a solution process described in Chapter 3, the coefficients of these polynomials are constrained to ensure linear variation of shape functions over the element edges, weak continuity over the inter-element boundaries (facets), and optimal smoothness over the element interior, as well as satisfaction of the Kronecker-delta property.

1.9.2 The Discrete Data Polyhedral FEM (DDPFEM)

The DDPFEM represents a further variation on the theme of polyhedral finite elements. The preliminary development presented here represents a second significant contribution of this dissertation. The main difference between this method and the VETFEM described above lies in the manner in which the shape functions and their gradients are calculated on the element domains. While the conventional FEM, particle methods, and the VETFEM all construct shape functions that are defined pointwise on the problem domain, the DDPFEM attempts to calculate values of shape functions and their gradients only at discrete locations (i.e. at nodes and/ or integration points, as required), without defining shape functions pointwise on the element domains. The discrete values of the shape functions and their gradients are constrained in order for

the results to be compatible with the given values at the domain boundary, and are otherwise required to be smooth, in a certain sense, on the element interior. The advantages of the DDPFEM with respect to the conventional FEM are comparable to those of the VETFEM stated above, and are described in more detail in Chapter 4 of this dissertation.

1.10 Dissertation organization

Chapter 2 introduces a typical boundary value problem in solid mechanics in a Total Lagrangian setting, which is mathematically described by an elliptic partial differential equation. The theoretical requirements that represent the basis of any finite element method are then outlined. To provide a point of departure the conventional FEM is presented, with an emphasis on its main characteristics and practical limitations. At the end of Chapter 2, a brief description of the proposed finite-element-like numerical approach (VETFEM) is given.

Chapter 3 describes the main contribution of this dissertation - formulation and development of the 3D Variable Element Topology Finite Element Method (VETFEM). The VETFEM as an alternative FEM is introduced, and its differences compared to the conventional FEM are emphasized. The VETFEM 3D element formulation is presented by forming element shape functions as low-order polynomials in the physical coordinates of the problem. Then, a set of constraints is imposed on the shape function coefficients in order to satisfy, among other properties, a weak continuity condition at inter-element

interfaces (facets). The remaining freedom in these coefficients is used to optimize the inter-element compatibility and smoothness on the element interior. Finally, computational examples, including patch tests, and a convergence demonstration, are presented to illustrate the performance of this method.

Chapter 4 outlines the Discrete Data Polyhedral FEM (DDPFEM), in which multiple constrained minimization processes are used to calculate values of the shape functions and their gradients at discrete locations (interior integration points) without defining shape functions pointwise on the interior of the elements.

Finally, Chapter 5 concludes the dissertation with a discussion of future research and possible improvements of the approximation techniques presented herein. These future efforts will primarily focus on the ability of the numerical models to impact demanding industrial applications involving large deformations, very complex geometries, and/ or crack propagation that require robust remeshing and material state remapping. Accordingly, Chapter 5 also includes a brief discussion of mesh-to-mesh solution remapping.

Chapter 2

Finite Element Formulation

To describe a typical boundary value problem (BVP) in solid mechanics, let us consider a body that moves through space, and at time t occupies a region Ω_t that is called the configuration at t . In the Total Lagrangian (TL) formulation, the initial configuration Ω_0 is used as the reference configuration throughout the motion, while in the Updated Lagrangian (UL) formulation the choice of the reference configuration is “updated” to become the configuration occupied by the body Ω_t after each increment of the motion, as illustrated in Figure (2.1). Furthermore, we will assume that the body moves relative to a stationary Cartesian coordinate system, and that the motion is arbitrarily large. We further suppose that the body is made of a material whose stress is governed by an arbitrary nonlinear local constitutive model. In the subsequent analysis, a *Total Lagrangian (or material) formulation* of the problem is adopted, where material points are identified by their locations in a fixed reference configuration.

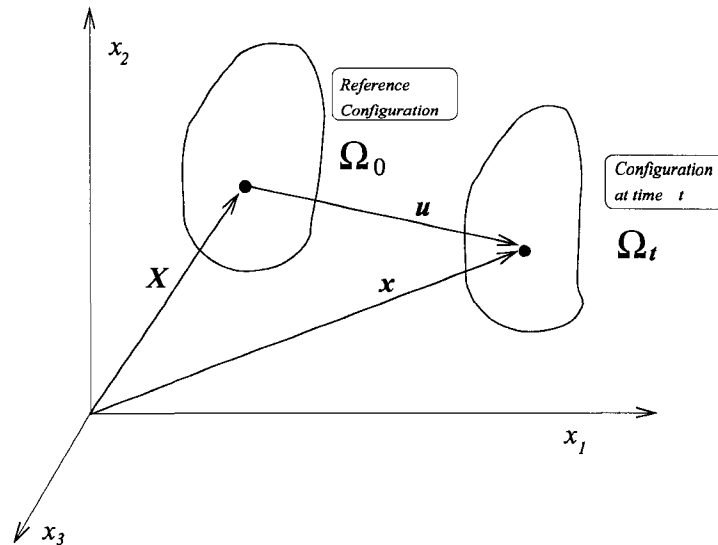


Figure 2.1: Motion of the body relative to a stationary Cartesian coordinate system

2.1 Continuum Mechanics Boundary Value Problem Formulation

Adopting a Total Lagrangian approach, on the global reference configuration the equilibrium of the body can be expressed in the so-called *strong form* as:

$$P_{ij,j} + \rho b_i = 0, \quad \mathbf{X} \in \Omega_0 \quad (2.1)$$

where \mathbf{P} is the first Piola-Kirchhoff stress tensor that represents forces acting on a deformed configuration but per unit reference (undeformed) configuration area, and is a function of the deformation gradient $\mathbf{F} = \partial \mathbf{x} / \partial \mathbf{X}$ and possibly its history; $\mathbf{x} = \mathbf{X} + \mathbf{u}$; and $\rho \mathbf{b}$ is the body force per unit reference volume. The displacement vector $\mathbf{u}(\mathbf{X}, t)$ enters the equilibrium equation (2.1) through the appropriate strain-displacement relations and constitutive (stress-strain) relations, where large strain and large displacement

are admitted.

In order to formulate the boundary value problem of quasistatic nonlinear solid mechanics, the field equations must be supplemented by suitable boundary conditions. I.e., the equilibrium statement (2.1) must be satisfied subject to given displacement $\bar{\mathbf{u}}$ on the portion of the boundary $\partial_u\Omega_0$, and given Piola traction vector $\bar{\mathbf{p}}$ on the portion of the boundary $\partial_t\Omega_0$. These boundary conditions can be mathematically stated as:

$$\mathbf{u} = \bar{\mathbf{u}}, \quad \mathbf{X} \in \partial_u\Omega_0$$

$$\mathbf{p} = \bar{\mathbf{p}}, \quad \mathbf{X} \in \partial_t\Omega_0$$

$$\text{where } \partial\Omega_0 = \partial_u\Omega_0 \cup \partial_t\Omega_0 \text{ and } \partial_u\Omega_0 \cap \partial_t\Omega_0 = \emptyset$$

2.2 Weak Form of Equilibrium and Discretization

In general, the body can undergo large displacements and large strains, and the constitutive relations are nonlinear. Therefore, the relation (2.1) cannot be solved directly to arrive at a solution for the displacements that are continuous in time. Our goal here is to evaluate equilibrium positions of the body at discrete points in time (i.e. $0, \Delta t, 2\Delta t, \dots$, where Δt is the time increment). Accordingly, in the solution strategy, we will assume that the solutions for the static and kinematic variables for time steps from 0 (i.e. $t = 0$) to time t have been obtained, and that the solution process for the next required equilibrium position corresponding to time $t + \Delta t$ is typical and would be applied repeatedly until the complete solution path is known.

In order to accomplish this task, various algorithms have been developed for obtain-

ing approximate solutions to the problem outlined above. The Finite Element Method (FEM) discretizes the problem domain (reference configuration) into subdomains (elements) and requires a different formulation that is called *weak form or variational form* of the boundary value problem. This formulation involves a less-strict continuity requirement on the displacement field.

Proceeding in this direction, define first a set S of displacement-field trial solutions \mathbf{u} and a set V of test functions \mathbf{v} on the problem domain Ω_0 , as follows:

$$S = \{\mathbf{u} \mid \mathbf{u} \in H^1(\Omega_0)^0, \mathbf{u} = \bar{\mathbf{u}}, \mathbf{X} \in \partial_u \Omega_0\}$$

$$V = \{\mathbf{v} \mid \mathbf{v} \in H^1(\Omega_0)^0, \mathbf{v} = \mathbf{0}, \mathbf{X} \in \partial_u \Omega_0\}$$

where $\partial_u \Omega_0$, as previously stated, is the part of the boundary of the domain Ω_0 with specified (known) displacements $\bar{\mathbf{u}}$. Here, S represents candidate or trial solutions that are required to satisfy given boundary conditions, and to belong to the Sobolev space of order one on Ω_0 (i.e. $\in H^1$ on the domain). This function space contains functions that are square-integrable on the domain Ω_0 (i.e. belong to $L_2(\Omega_0)$) and possess weak derivatives of order one that also belong to $L_2(\Omega_0)$.

The class of functions V contains the weighting functions or variations. This set is very similar to the trial solutions, except that we require the homogeneous counterpart to the displacement boundary conditions on $\partial_u \Omega_0$.

The weak form of equilibrium can then be expressed as:

$$\int_{\Omega_0} v_i (P_{ij,j} + \rho b_i) dV = 0 \quad \forall \mathbf{v} \in V. \quad (2.2)$$

Applying the 3D divergence theorem and the boundary condition $p_i = \bar{p}_i$ on the part of the boundary $\partial_t \Omega_0$ with specified (known) traction $\bar{\mathbf{p}}$, the problem can be formulated as follows:

find $\mathbf{u} \in S = \{\mathbf{u} \mid \mathbf{u} \in H^1(\Omega_0)^0, \mathbf{u} = \bar{\mathbf{u}}, \mathbf{X} \in \partial_u \Omega_0\}$ such that

$$\int_{\Omega_0} v_{i,j} P_{ij} dV = \int_{\partial_t \Omega_0} v_i \bar{p}_i dA + \int_{\Omega_0} \rho v_i b_i dV \quad \forall \mathbf{v} \in V. \quad (2.3)$$

where $\mathbf{v} \in V = \{\mathbf{v} \mid \mathbf{v} \in H^1(\Omega_0)^0, \mathbf{v} = \mathbf{0}, \mathbf{X} \in \partial_u \Omega_0\}$

Comparing equations (2.1) and (2.3), it can be shown that, if the displacement field $\mathbf{u}(\mathbf{X})$ is sufficiently continuous (e.g. if $\mathbf{u} \in C^2(\Omega_0)$), then the strong form and the weak form implicate the same solution. However, the weak form is much more convenient for purposes of an approximate numerical solution.

An approximate solution to the BVP outlined above is obtained by enforcing the weak form of equilibrium for a finite-dimensional subspace of V and at discrete points in time. In order to accomplish this, an Updated Lagrangian approach is taken, where all static and kinematic variables are referred to the known configuration at time t . I.e., the reference configuration is the known, beginning-step configuration Ω_t and we seek an incremental displacement $\hat{\mathbf{u}}$ such that:

$$\hat{\mathbf{u}} = \mathbf{x} - \bar{\mathbf{x}}, \quad \text{where:}$$

$\bar{\mathbf{x}}$ – configuration (material point locations) at the beginning of the time step

\mathbf{x} – configuration at the end of the time step

\mathbf{X} – global reference configuration position

The increment $\hat{\mathbf{u}}$ in each time (load) step is determined through iterations until a pre-specified convergence criterion is satisfied.

The weak form (2.3) is applied in each time step, and the residual evaluated in each iteration within the step to arrive at the suitable incremental displacement $\hat{\mathbf{u}}$:

$$R = \int_{\Omega_{t_k}} v_{i,j} P_{ij}^{t_{k+1}} dV - \int_{\partial_t \Omega_{t_k}} v_i \bar{p}_i^{t_{k+1}} dA - \int_{\Omega_{t_k}} \rho v_i b_i^{t_{k+1}} dV \quad (2.4)$$

Here, Ω_{t_k} represents the reference (beginning-step) configuration in the Updated Lagrangian setting, and stresses and tractions are evaluated at the end of the step. The residual (2.4) is expanded in a Taylor series about a known (initial guess or the previous iteration value) incremental displacement, which is truncated at the linear term. The linearized residual equation is then solved for an adjustment to the displacement increment. This procedure is repeated until the convergence criterion is satisfied. The resulting incremental displacement $\hat{\mathbf{u}}$ is added to the total displacement, and the solution is advanced to the next time (load) step. The incremental nodal displacements in one time step can be projected forward to provide an initial guess for the next step.

2.3 The Conventional FEM and Galerkin Approach

In a finite element analysis, we discretize the body into an assemblage of finite elements, with the elements sharing facets and their associated nodal points on the element boundaries. In the displacement based Finite Element Method used in solid mechanics, a displacement field $\mathbf{u}(\mathbf{x})$ is expressed as a linear combination of linearly

independent basis functions:

$$\mathbf{u}(\mathbf{x}) = \sum_{a=1}^N \mathbf{u}_a \phi_a(\mathbf{x}) \quad (2.5)$$

$$\text{where: } \mathbf{u}(\mathbf{x}) = \mathbf{0} \iff \mathbf{u}_a = \mathbf{0}, \quad a = 1, 2, \dots, N.$$

Basis functions $\phi_a(\mathbf{x})$ associated with each node a have the following properties:

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

$$\phi_a(\mathbf{x}_b) \neq 0 \quad \text{only on elements containing node } a$$

$$\phi_a(\mathbf{x}) \text{ are continuous on the problem domain } B \text{ (i.e. } \phi_a \in C^0(B))$$

In the expression above and subsequently, B generically stands for the configuration Ω_{t_k} that is being discretized. It can be seen that coefficients \mathbf{u}_a in equation (2.5) are actually the nodal values of $\mathbf{u}(\mathbf{x})$:

$$\mathbf{u}(\mathbf{x}_b) = \sum_{a=1}^N \mathbf{u}_a \phi_a(\mathbf{x}_b) = \sum_{a=1}^N \mathbf{u}_a \delta_{ab} = \mathbf{u}_b$$

In conjunction with the weak form of equilibrium (2.3), the problem can be formulated in the following way: Determine coefficients \mathbf{u}_a in (2.5) so that the displacement field $\mathbf{u}(\mathbf{x})$ satisfies the weak form (2.3), for a suitable space of test functions.

The displacement field (2.5) can also be expressed in the following form:

$$\mathbf{u}(\mathbf{x}) = \sum_{a \in \eta_o} \mathbf{u}_a \phi_a(\mathbf{x}) + \sum_{a \in \eta_u} \bar{\mathbf{u}}_a \phi_a(\mathbf{x})$$

where:

η_u - the set of nodes on the part of the boundary $\partial_u B$ with known

(specified) displacement ($\mathbf{x}_a \in \partial_u B, a \in \eta_u$)

η_o - all other nodes (i.e. $\eta_o = \{1, 2, \dots, N\} \setminus \eta_u$)

A Galerkin method is obtained if the finite-dimensional space of test functions is given by:

$$v(\mathbf{x}) = \sum_{a \in \eta_o} v_a \phi_a(\mathbf{x}), \quad v_a \text{ arbitrary.} \quad (2.6)$$

The weak form (2.3) combined with the expression (2.6) for test functions results in the following discretized form of equation (2.3):

$$R_{ia} = \int_B \phi_{a,j} P_{ij} dV - \int_{\partial_t B} \phi_a \bar{p}_i dA - \int_B \rho \phi_a b_i dV = 0, \quad a \in \eta_o.$$

or in vector form:

$$\mathbf{R}_a = \int_B \mathbf{B}_a^T \mathbf{P} dV - \int_{\partial_t B} \phi_a \bar{\mathbf{p}} dA - \int_B \rho \phi_a \mathbf{b} dV = 0, \quad a \in \eta_o. \quad (2.7)$$

Here, \mathbf{R}_a is a vector that contains contributions from displacement boundary conditions on $\partial_u B$, traction boundary conditions on $\partial_t B$, the body force \mathbf{b} , and the stress divergence term; and \mathbf{B}_a is a 6×3 matrix associated with each node $a \in \eta_o$ containing derivatives $\phi_{a,i}$ of the shape function ϕ_a associated with node a . It can be seen that this is the finite-dimensional form of the residual equation (2.4) that is applied in each time step during the incremental analysis outlined in Section 2.2.

2.4 Main Characteristics and Constraints of the Conventional FEM

The main characteristics of the conventional FEM emerge from the isoparametric formulation of standard FEM elements on a simple, parent domain. The isoparametric

mapping is illustrated in Figure (2.2). In the conventional FEM, shape functions are

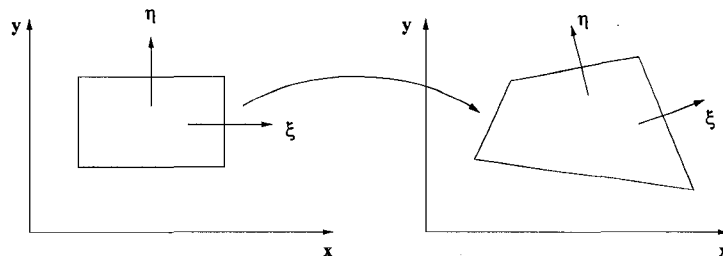


Figure 2.2: Isoparametric coordinate mapping

defined over a simple, parent domain Ω . Then the isoparametric (coordinate) transformation is applied to transform integrals over the physical element domain Ω^m to integrals over the parent domain Ω :

$$\int_{\Omega^m} f(x_1, x_2, x_3) dV = \int_{\Omega} f[x_1(\xi_1, \xi_2, \xi_3), x_2(\xi_1, \xi_2, \xi_3), x_3(\xi_1, \xi_2, \xi_3)] J(\xi_1, \xi_2, \xi_3) d\xi_1 d\xi_2 d\xi_3 \quad (2.8)$$

A typical integral on the left side of equation (2.8) contains shape function derivatives with respect to physical coordinates x_i of the domain. To be able to evaluate various terms in the weak-form integrals, we need to calculate the strain-displacement transformation matrix \mathbf{B}_a where the element strains are obtained in terms of derivatives of element displacements with respect to the local (parent element) coordinate system. Because the element displacements are defined on the parent element, we need to relate derivatives with respect to x_1 , x_2 and x_3 to derivatives with respect to ξ_1 , ξ_2 and ξ_3 . In order to obtain derivatives with respect to the global coordinate system, i.e. $\frac{\partial}{\partial x_a}$, the chain rule of differentiation is employed in the following form:

$$\frac{\partial}{\partial x_k} = \frac{\partial \xi_a}{\partial x_k} \frac{\partial}{\partial \xi_a} = J_{ak}^{-1} \frac{\partial}{\partial \xi_a} \quad (2.9)$$

while the inverse relation is:

$$\frac{\partial}{\partial \xi_k} = \frac{\partial x_a}{\partial \xi_k} \frac{\partial}{\partial x_a} = J_{ak} \frac{\partial}{\partial x_a} \quad (2.10)$$

where J_{ak} is the *Jacobian operator* relating local (parent element) coordinate derivatives to the global coordinate derivatives:

$$J_{ak} = \frac{\partial x_a}{\partial \xi_k} = \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} & \frac{\partial x_1}{\partial \xi_3} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} & \frac{\partial x_2}{\partial \xi_3} \\ \frac{\partial x_3}{\partial \xi_1} & \frac{\partial x_3}{\partial \xi_2} & \frac{\partial x_3}{\partial \xi_3} \end{bmatrix} \quad (2.11)$$

The existence of equation (2.9) requires that the inverse of J_{ak} exists; which holds provided that the mapping from parent to physical element is one-to-one and onto.

It bears mentioning that except for very simple cases, volume and surface element integrals must be evaluated by means of numerical integration (Gauss–Legendre, Newton–Coates, Lobatto are among the most commonly used integration rules).

The isoparametric transformation allows shape functions to take a very simple mathematical form on the parent element domain (generally low-order polynomials), while

allowing for distorted physical element geometry. The isoparametric transformation requires that the Jacobian Determinant J , which represents the ratio of volume elements in physical and parent space (i.e. $J(\xi_1, \xi_2, \xi_3) = dV/d\xi_1 d\xi_2 d\xi_3$), and which relates derivatives of shape functions with respect to physical coordinates $\partial N_\alpha / \partial x_i$ to derivatives with respect to parent coordinates, is known. This transformation ultimately leads to rather complicated functions of ξ_1, ξ_2 , and ξ_3 as the integrand on the right side of equation (2.8). Isoparametric transformations (2.8) require that each element must be topologically identical to that of the parent element, in the sense that node-facet and facet-element associations must be the same. Along with this requirement, a significant geometric constraint on the degree of distortion of the element is also present. Failure to satisfy this constraint renders the isoparametric mapping non-invertible, which in turn causes the Jacobian to be singular, ultimately rendering the weak-form integrals invalid.

However, this type of element formulation is also the main source of the powerful properties of the conventional FEM: the boundary of the domain is precisely and explicitly defined; the domain is subdivided into elements that facilitate the description of complex geometries and account for different materials within the domain; a convenient implementation of boundary conditions (essential and natural) is obtained; and the compact support property of basis functions leads to a sparse and banded global equation structure. These powerful features have driven the development of a wide range of commercially available FEM software, and ultimately explain why the conven-

tional FEM is the most widely used computational technique for solving solid mechanics problems of technological interest.

2.5 Introduction to the VETFEM

The Variable Element Topology Finite Element Method (VETFEM) represents an alternative to the conventional finite element method as well as to particle (meshless) methods, the two classes of approximation techniques currently available for solid mechanics problems involving complex/ evolving geometry. The VETFEM seeks to retain all of the powerful characteristics of the conventional FEM and, at the same time, allow a level of flexibility in the spatial discretization that approaches that of particle methods. The VETFEM uses elements that are subject to geometric restrictions that are much milder than in the conventional FEM: the elements can be general polyhedra, and can have any number of nodes in any arrangement.

This substantial advantage of the VETFEM, and the main contribution of the work presented in the next chapter of this dissertation, flows from the manner in which basis functions are constructed. The conventional FEM basis functions arise from the isoparametric transformation of the shape functions defined over the parent element. This isoparametric transformation is the main source of the topological and geometric restrictions for the conventional method. On the other hand, VETFEM elements do not involve any isoparametric transformation, and there is no parent element. In the VETFEM, shape functions are constructed directly in the form of polynomials in the physical

coordinates of the element. The coefficients of these polynomials are obtained through a constrained minimization process that renders linear variation of shape functions over the element edges, weak continuity over the inter-element boundaries (facets), and optimal smoothness over the element interior. A full account of the three dimensional VET element formulation is given in the next chapter.

Chapter 3

3D Element Formulation of a Finite Element Method with General Polyhedral Element Geometry

3.1 Introduction

Chapter 2 briefly presented the conventional, displacement based Finite Element Method as the dominant computational tool for solving problems in solid mechanics. Among the powerful features of the conventional FEM are its ability to accommodate complex geometries with a straight forward and precise imposition of natural and essential boundary conditions, as well as a high level of accuracy relative to the computational effort. However, for constructing high-quality meshes, the conventional FEM imposes a strict set of rules that arise from the isoparametric coordinate transformations described in the previous chapter. Many complex solid mechanics problems in modern industry, such as crack propagation and metal cutting, require continuous remeshing of the problem domain through the solution process. It is in these instances that the mesh

forming rules of the conventional FEM represent a significant bottleneck in the mesh generation process, and a burden on human and computational resources.

The Variable Element Topology Finite Element Method (VETFEM) was first introduced in two dimensions by Rashid and Gullett (2000, 2002) [65], [66]. As with any Galerkin approximation method, the role of the VETFEM is to provide both a set of suitable basis functions on the spatial domain B , and a means of evaluating the weak-form integrals on the domain. The VETFEM is a true finite element method for second order elliptic problems, with the main distinction that its elements can take any polyhedral shape, with much milder non-convexity restrictions. These relaxed geometric requirements greatly simplify the automatic mesh generation task comparing to the conventional FEM.

In the following, the three dimensional VETFEM element formulation in a Lagrangian setting is presented for the variational boundary value problem of solid mechanics. VETFEM shape (basis) functions are described first, as low order polynomials in the element physical coordinates. Then, a set of constraints is imposed on the polynomial coefficients in order to satisfy a weak continuity condition at inter-element interfaces (facets), and to optimize the inter-element compatibility and smoothness on the element interior. Finally, computational examples, including patch tests, are presented to illustrate the performance of the method.

3.2 VETFEM Element Formulation

The problem domain B is subdivided into arbitrary polyhedral elements with polygonal facets, and nodes are placed exclusively at the vertices. As with the conventional FEM, VETFEM elements are used to construct shape functions and to develop and apply suitable numerical quadrature.

3.2.1 Shape Functions

In the VETFEM, shape functions (and therefore basis functions) are not defined on a parent element as in the conventional FEM. Instead, they take the form of low-order polynomials in the physical coordinates of the element. Globally defined basis functions $\Phi_a(\mathbf{x})$ associated with each node a ($a = 1, 2, \dots, n$) are composed from shape functions ϕ_a defined on a typical element as follows:

$$\phi_a = \sum_{1 \leq j \leq M} G_j^a q_j \quad a = 1, 2, \dots, n, \quad (3.1)$$

where:

G_j^a - shape function coefficients

q_j - monomials in x_1, x_2, x_3

$M = (m + 1)(m + 2)(m + 3) / 6$ - number of monomials in the shape function polynomial

m - order of the (complete) shape function polynomial

3.2.2 Determination of Shape Function Coefficients

The shape function coefficients G_j^a are determined in such manner that certain finite-element-like properties of the method are ensured. With this in mind, a set of constraints is imposed, first on the shape function coefficients so that the shape functions exhibit the Kronecker-delta property:

$$\phi_a(\mathbf{x}_b) = \sum_{1 \leq j \leq M} G_j^a q_j(\mathbf{x}_b) = \delta_{ab} \quad 1 \leq a, b \leq n \quad (3.2)$$

This constraint ensures that the solution is a true interpolant of the nodal displacements, which in turn, results in a convenient and straightforward implementation of kinematic boundary conditions as in the conventional FEM. It should be mentioned that the constraints (3.2) are always linearly independent, and, for sufficiently large polynomial order m , are also compatible.

A second set of constraints is imposed on the shape function coefficients in order to enforce so-called “weak continuity” of the shape functions at inter-element interfaces (two dimensional facets). The weak continuity condition relates to the fact that the shape functions are assembled as low-order polynomials in the physical coordinates and therefore strict continuity on the interelement facets cannot be achieved for general polyhedral element geometries. The VETFEM is therefore regarded as a so-called incompatible method. However, it can be shown that the overall method is still convergent under mesh refinement if the weak continuity condition is satisfied. The weak continuity condition takes the following form: the mean value of the two shape functions on either

side of a given facet associated with the adjoining elements equals the same quantity. This is accomplished by requiring that the mean of each shape function on a given facet equals the mean value of a function with an “ideal” variation on that facet.

In the two dimensional case, a linear function along facets (i.e. edges) is an appropriate choice for this “ideal” variation. In the three dimensional case, however, the choice of this variation is not so readily apparent. In the 3D setting, we choose the harmonic function that satisfies the Dirichlet boundary conditions of piecewise linear variation along the facet edges, consistent with the Kronecker-delta property, to represent the “ideal” variation (function). The weak continuity condition on a given facet is then imposed by seeking equality between the mean value of the shape function on the facet, and the mean value of this ideal variation. The details are as follows.

Let us consider an element with k polygonal facets Γ_b ($\Gamma_b \subset \partial\omega$), where $\partial\Gamma_b$ represents the facet boundary (i.e. edges). The “ideal” variation on Γ_b for a shape function ϕ_a associated with an element ω is then represented by:

$$\begin{aligned} \bar{\nabla}^2 \psi_{ab} &= 0, \quad \mathbf{x} \in \Gamma_b; \quad \psi_{ab} = f_{ab}, \quad \mathbf{x} \in \partial\Gamma_b; \quad a \in \partial\Gamma_b, \\ \psi_{ab} &= 0, \quad a \notin \partial\Gamma_b. \end{aligned} \tag{3.3}$$

where:

f_{ab} - piecewise linear function on $\partial\Gamma_b$, consistent with the Kronecker-delta property at the nodes ($f_{ab}(\mathbf{x}_c) = \delta_{ac}$), with piecewise linear variation on $\partial\Gamma_b$ between the nodes
 $\bar{\nabla}^2$ - 2D Laplacian in the plane of Γ_b

The weak continuity constraint can then be stated as:

$$\int_{\Gamma_b} (\phi_a - \psi_{ab}) da = 0 \quad 1 \leq a \leq n, \quad 1 \leq b \leq k \quad (3.4)$$

This constraint requires the calculation of the mean value of the function ψ_{ab} . An acceptable approximation to this mean value can be obtained without explicitly solving the boundary value problem (3.3) by considering the following:

$$(\xi_i \psi)_{,i} = 2\psi + \frac{1}{2} (\xi_j \xi_j \psi_{,i})_{,i}$$

where ξ_i , $i = 1, 2$ are Cartesian coordinates in the plane of a typical element facet Γ , with the origin at the facet centroid, and $\psi(\xi_1, \xi_2)$ is a typical function ψ_{ab} associated with the facet Γ .

Employing the divergence theorem, the integral $\int_{\Gamma} \psi da$ becomes:

$$\int_{\Gamma} \psi da = \int_{\partial\Gamma} \left(\frac{1}{2} \xi_i \psi n_i - \frac{1}{4} \xi_j \xi_j \psi_{,i} n_i \right) ds \quad (3.5)$$

where n_i are components of the outward unit normal to the facet boundary $\partial\Gamma$ in the plane of the facet Γ .

It is important to emphasize at this point that our goal is that the shape functions on the shared facets of two neighboring elements have the same mean value. Whether that mean value is compared to the mean value of the “ideal” variation, or to a close approximation to it, is immaterial - as long as the mean values of the shape functions are identical. In other words, we can compare the mean of the shape functions on shared facets with the same *approximation* of the mean of the “ideal” variation (3.3), without

any detrimental effects on the result. With this in mind, the second term in equation (3.5) can be neglected in favor of the first one, leading to the following weak continuity constraint:

$$\int_{\Gamma_b} \phi_a da = \frac{1}{2} \int_{\partial\Gamma_b} f_{ab} \xi_i n_i ds \quad 1 \leq a \leq n, \quad 1 \leq b \leq k \quad (3.6)$$

where f_{ab} is as defined in (3.3) above.

The right-hand side of (3.6) is easily evaluated once the geometry of the element is known in the following way:

$$\int_{\partial\Gamma_b} f_{ab} \xi_i n_i ds = \int_{\partial\Gamma_b} f_{ab} (x_i - x_{c_i}) n_i ds \quad (3.7)$$

where $\xi = \mathbf{x} - \mathbf{x}_c$, \mathbf{x}_c is the position vector of an arbitrary, fixed point C on the facet Γ_b (i.e. the facet centroid), and $\mathbf{n} = \boldsymbol{\lambda} \times \mathbf{n}_{fet}$, as defined in Figure (3.1) below,

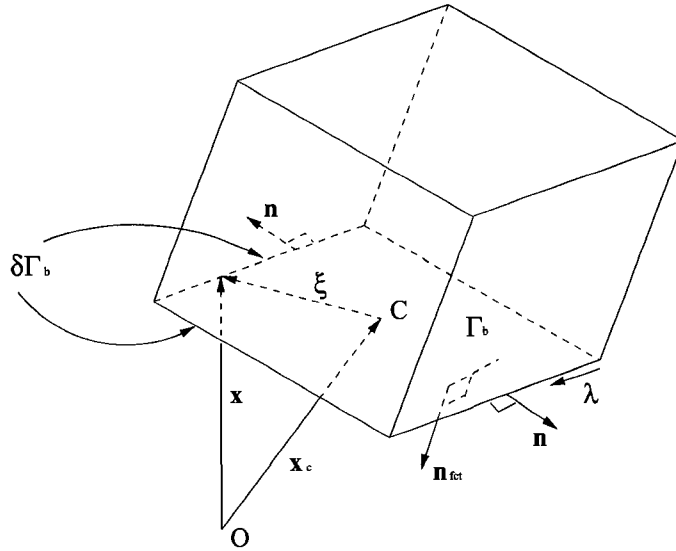


Figure 3.1: Geometrical illustration of integrands in equation (3.7).

Now, the integral on the right-hand side of (3.7) becomes:

$$\begin{aligned} \int_{\partial \Gamma_b} f_{ab} (x_i - x_{c_i}) n_i ds = \\ \sum_l \frac{n_i^l}{L^l} \lambda_j^l \left(\int_l x_i^l x_j^l ds - x_{e_j} \int_l x_i^l ds \right) - x_{c_i} \sum_l n_i^l \int_l f_{ab} ds = \\ \sum_l \frac{n_i^l}{L^l} \lambda_j^l \left(\int_l x_i^l x_j^l ds - x_{e_j} \int_l x_i^l ds \right) - \frac{1}{2} x_{c_i} \sum_l n_i^l L^l \end{aligned} \quad (3.8)$$

$$\int_l f_{ab} ds = \frac{L^l}{2}$$

only on the segments (edges) of $\partial\Gamma_b$ containing node a , and zero otherwise.

Equations (3.2) and (3.6) represent $(n+k)$ linear constraints on the M shape function coefficients G_j^a (3.1) for each shape function. An additional three constraints on the coefficients G_j^a in 3D have to be imposed to ensure convergence. These constraints have a very small impact on the final values of the coefficients, but are necessary when the stress-divergence term in (2.3) is evaluated by means of numerical quadrature. This term has to be integrated exactly in the case of a uniform stress field as a necessary condition for convergence of the method. Accordingly, the three additional constraints are as follows:

$$\int_{\omega} \phi_{a,j} dV = \int_{\partial\omega} \phi_a n_j da$$

The total of $(n + k + 3)$ linear constraints on the shape function coefficients G_j^a (3.1) for each shape function can be written in the following matrix form:

$$CG^a = b^a \tag{3.9}$$

where:

C - the $(n + k + 3) \times M$ constraint matrix

G^a - the $M \times 1$ vector of polynomial coefficients for shape function number a

b^a - the right hand side vector that depends only on the element geometry

Importantly, while the right-hand-side vector b^a is specific to the shape function ϕ_a , the coefficient matrix C is independent of the shape function number. Therefore, it must be factored only once per element.

3.2.3 Shape Function Smoothness Optimization

The number of monomial terms is chosen in a manner such that the number M of unknown coefficients G_j^a is larger than the number of constraints (i.e. $M \geq (n + k + 3)$), which typically results in shape functions having a polynomial order of three or four. The excess freedom is used to further improve the quality of the polynomial shape functions. Specifically, and consistent with the constraints indicated above, the shape function coefficients are chosen so that inter-element compatibility, smoothness on the element's interior, and variation of shape functions along element edges are optimized. This optimization is achieved through minimization of the positive semi-definite functionals F_a :

$$\begin{aligned}
 F_a = & \frac{\alpha}{|\omega|} \int_{\omega} (\nabla \cdot \nabla \phi_a)^2 dV + \frac{\beta}{|\partial\omega|} \int_{\partial\omega} (\bar{\nabla} \cdot \bar{\nabla} \phi_a)^2 dA \\
 & + \frac{1 - \alpha - \beta}{\Sigma|\partial\Gamma|} \sum_{\Gamma} \int_{\partial\Gamma} (\tilde{\nabla} \cdot \tilde{\nabla} \phi_a)^2 ds
 \end{aligned} \tag{3.10}$$

where the Laplacian in three, two and one dimension respectively provides measures of curvature of the scalar field ϕ_a on the element interior, facets, and edges:

$\nabla \cdot \nabla \phi_a$ - in 3D - (in this case domain ω)

$\bar{\nabla} \cdot \bar{\nabla} \phi_a$ - on a 2D surface (in this case $\partial\omega$ - the boundary of the domain ω)

$\tilde{\nabla} \cdot \tilde{\nabla} \phi_a$ - on a 1D line (in this case $\partial\Gamma$ - the boundary of the facet Γ)

and:

$\alpha, \beta > 0, \alpha + \beta < 1$ - coefficients to be chosen to ensure a desired blend of the three

integral terms in the functional above.

The gradient operators $\bar{\nabla}$ and $\tilde{\nabla}$ in 2D and 1D respectively are defined as follows:

$\bar{\nabla} = \nabla - \mathbf{n} \mathbf{n} \cdot \nabla$ - gradient operator on a 2D surface with normal \mathbf{n} .

$\tilde{\nabla} = \boldsymbol{\lambda} \cdot \nabla$ - gradient operator on a line with tangent $\boldsymbol{\lambda}$.

By substituting (3.1) into (3.10), the functionals F_a can be written as quadratic functions in the shape-function coefficients \mathbf{G}^a :

$$F_a = \mathbf{G}^{a^T} \mathbf{A} \mathbf{G}^a \quad (3.11)$$

The $M \times M$ matrix \mathbf{A} in (3.11) is positive semi-definite, and is independent of the shape function number a :

$$\begin{aligned} A_{ij} = & \frac{\alpha}{|\omega|} \int_{\omega} q_{i,mm} q_{j,nn} dV \\ & + \frac{\beta}{|\partial\omega|} \int_{\partial\omega} (q_{i,mm} - n_m n_n q_{i,mn}) (q_{j,kk} - n_k n_l q_{j,kl}) da \\ & + \frac{1 - \alpha - \beta}{\Sigma|\partial\Gamma|} \sum_{\Gamma} \int_{\partial\Gamma} \lambda_m \lambda_n q_{i,mn} \lambda_k \lambda_l q_{j,kl} ds \end{aligned} \quad (3.12)$$

The F_a are minimized with respect to their respective shape function coefficients G_j^a , subject to the linear constraints (3.9).

3.2.4 Numerical Evaluation of Integrals of Monomial Terms

Determination of the shape function coefficients G_j^a that account for the Kronecker-delta property at the nodes, weak continuity condition at inter-element interfaces, inter-element compatibility, smoothness on the element's interior, and optimal variation of shape functions along element edges involves evaluating integrals of a typical monomial

$q_j(\mathbf{x}) = x^a y^b z^c$ on the element interior ω , its boundary $\partial\omega$, and the facet edges $\partial\Gamma$.

The volume integral of the monomial $q_j(\mathbf{x}) = x^a y^b z^c$ over ω can be converted to a boundary integral using the 3D divergence theorem, as follows:

$$\begin{aligned} \int_{\omega} x^a y^b z^c dV &= \frac{1}{3} \int_{\omega} \left[\left(\frac{x^{a+1} y^b z^c}{a+1} \right)_{,1} + \left(\frac{x^a y^{b+1} z^c}{b+1} \right)_{,2} + \left(\frac{x^a y^b z^{c+1}}{c+1} \right)_{,3} \right] dV \\ &= \frac{1}{3} \int_{\partial\omega} \left[\left(\frac{x^{a+1} y^b z^c}{a+1} \right) n_1 + \left(\frac{x^a y^{b+1} z^c}{b+1} \right) n_2 + \left(\frac{x^a y^b z^{c+1}}{c+1} \right) n_3 \right] dA \end{aligned} \quad (3.13)$$

Here, $\mathbf{n} = (n_1, n_2, n_3)$ is the outward unit normal on the surface $\partial\omega$, and is constant over each element facet. A typical integral in this expression is:

$$\int_{\partial\omega} x^m y^n z^l dA = \sum_{\Gamma} \int_{\Gamma} q dA, \quad (3.14)$$

where q is used generically to denote a monomial in x, y , and z .

Evaluation of the area integral on the right-hand side in the equation above on a single facet proceeds as follows:

Define vectors: $\mathbf{v} = q(\mathbf{y} \times \mathbf{n})$, $\mathbf{y} = \mathbf{x} - \mathbf{c}$

where:

\mathbf{x} - spatial position of any point on the facet (plane) Γ

\mathbf{c} - spatial position of an arbitrary, fixed point on the facet Γ

\mathbf{n} - outward unit normal on the facet

Note that $\mathbf{y} \cdot \mathbf{n} = 0$ - since vector \mathbf{y} lies in the facet plane.

Applying Stokes' Theorem,

$$\int_{\Gamma} \mathbf{n} \cdot (\nabla \times \mathbf{v}) dA = \int_{\partial\Gamma} \mathbf{v} \cdot \boldsymbol{\lambda} ds$$

where: $\boldsymbol{\lambda}$ - unit vector tangent to the facet boundary (edges),

to the facet integral (3.14) above results in the following form for the integral $\int_{\Gamma} q \, dA$:

$$\int_{\Gamma} q \, dA = \frac{1}{2 + a + b + c} \left\{ \int_{\partial\Gamma} (\boldsymbol{\lambda} \times \mathbf{n}) \cdot (\mathbf{x} - \mathbf{c}) q \, ds + \int_{\Gamma} \mathbf{c} \cdot \nabla q \, dA \right\}$$

Considering that the right-hand side should not depend on the choice of the vector \mathbf{c} , and applying the 2D divergence theorem to the facet and its boundary, we obtain the final expression for the integral $\int_{\Gamma} q \, dA$:

$$\int_{\Gamma} q \, dA = \frac{1}{2 + a + b + c} \left\{ \mathbf{n} \cdot \int_{\partial\Gamma} \mathbf{x} \times \boldsymbol{\lambda} q \, ds + h \mathbf{n} \cdot \int_{\Gamma} \nabla q \, dA \right\} \quad (3.15)$$

where $h = \mathbf{c} \cdot \mathbf{n}$ is the perpendicular distance from the origin to the facet plane.

The above equation allows area integrals of each monomial q of order k to be calculated if area integrals of monomials of order $k-1$ are known. Thus, recursive recovery of integrals of all monomials of a given order leads to an efficient computational strategy.

In summary, line integrals of all monomials are evaluated on element edges using Gauss quadrature of a sufficient order to ensure exact values. Then, equation (3.15) is used to recursively integrate monomials on each facet and, finally, (3.13) is employed to integrate monomials on the interior of the element ω .

3.2.5 Integration Rule

In non-linear problems, the material state must be evaluated and stored at discrete points within each element before using numerical integration to form the stress divergence term in (2.3). The VETFEM provides a suitable quadrature rule for placement of the integration points within the element, and assignment of weights to each integration

point, as described below.

A typical 3D VETFEM element is subdivided into subregions, with one corresponding to each node of the element. The subregions are constructed as illustrated in the Figure (3.3): first the centroid of each facet is connected to the middle of each edge of the facet. Then, the edge mid-sides and the facet centroids are connected to the centroid of the element. The volume and the centroid of the resulting subregions can then be evaluated by integrating 1 and \mathbf{x} over the subregions using the approach described in the previous subsection. The integration points for the element then take the location of the centroids of the subregions, and the corresponding weights are taken to be the volumes of the subregions. This one-to-one correspondence between the element nodes

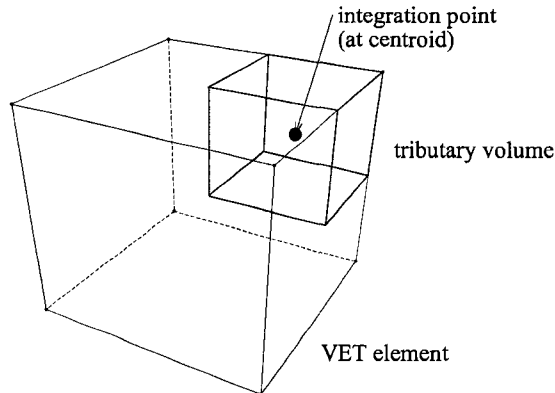


Figure 3.3: Placement of integration points for 3D VETFEM elements

and integration points has many advantages in facilitating the solution remapping that follows remeshing in problems that involve extreme deformations [64]. Another positive

feature of this integration rule is that an arbitrary linear function on the element is exactly integrated.

3.2.6 Kinematic Enhancement

It is known that low order, displacement based finite element methods behave poorly near the incompressibility limit when the incompressibility constraint is strictly enforced at the integration points. The enforcement of incompressibility is especially demanding when non-linear problems are analyzed. Volumetric locking is exhibited in the conventional FEM by standard bi-linear and tri-linear elements, as well as in VETFEM elements, as a result of the inability of low-order interpolants (polynomials) to adequately represent general volume preserving displacement fields. However, low-order polynomials are generally preferred because of their simplicity and algorithmic robustness, especially for large scale problems.

The severity of volumetric locking can be influenced by controlling the manner in which the dilatation (volume change) is sampled. In [47], an approach is taken whereby the element's quadrature rule is reduced in order, so as not to constrain the volume-preserving modes of deformation. This approach results in a significant reduction of the number of integration points, which makes it appealing from the perspective of computational efficiency, especially for explicit-dynamic problems. However, this approach usually produces certain unresisted deformation modes that require the addition of some form of artificial stiffness, in order to prevent unresisted distortion of the mesh.

In a different approach that is described here and in [67], the incremental deforma-

tion calculated in each time step is modified (or enhanced) at the integration points in such a manner that prevents volumetric locking in VETFEM elements. In particular, the algorithm takes a scaled form of the incremental deformation gradient $\hat{\mathbf{F}}$ as the input and computes a constant stretching rate tensor \mathbf{D} and an incremental rotation $\hat{\mathbf{R}}$ that are used to update the stress state.

For the purpose of outlining this approach in more detail, we define the incremental deformation gradient $\hat{\mathbf{F}}$, the beginning-step deformation gradient $\bar{\mathbf{F}}$, and the end-step (total) deformation gradient \mathbf{F} as:

$$\hat{\mathbf{F}} = \frac{\partial \mathbf{x}}{\partial \bar{\mathbf{x}}}, \quad \bar{\mathbf{F}} = \frac{\partial \bar{\mathbf{x}}}{\partial \mathbf{X}}, \quad \mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} = \hat{\mathbf{F}} \bar{\mathbf{F}}$$

Here, \mathbf{X} , $\bar{\mathbf{x}}$ and \mathbf{x} represent position vectors of a typical material point in the reference, beginning-step and end-step configurations. The incremental deformation gradient $\hat{\mathbf{F}}$ contains the kinematic information necessary to update stresses, and is computed in each time step at each integration point. The average incremental deformation for the element is:

$$\mathbf{f} = \frac{1}{|\bar{\omega}|} \int_{\bar{\omega}} \hat{\mathbf{F}} \, dv$$

This integral can be evaluated either by employing the divergence theorem, or using the quadrature rule for the element. Then, at each integration point, the modified incremental deformation gradient $\tilde{\mathbf{F}}$ is calculated as follows:

$$\tilde{\mathbf{F}} = \left(\frac{A}{\hat{J}} \right)^{1/d} \hat{\mathbf{F}}$$

where: $\hat{J} = \det \hat{\mathbf{F}}$, $A = \det \mathbf{f}$, and $d = 2$ or 3 is the dimensionality of the problem.

The modified incremental deformation gradient $\tilde{\mathbf{F}}$ is passed to the so-called *strongly objective kinematic algorithm* [63] to compute the aforementioned constant stretching rate tensor \mathbf{D} and the incremental rotation tensor $\hat{\mathbf{R}}$. With this approach, the deformation at each integration point is modified to assume the average dilatation for the entire element, while the deviatoric part remains locally unchanged. The constitutive equations are integrated and the stress computed over the step under the assumption of a constant stretching rate \mathbf{D} and zero vorticity \mathbf{W} (where $\mathbf{D} + \mathbf{W} = \mathbf{L}$ is the spatial velocity gradient). The final updated stress is then obtained by a pure stepwise rotation $\hat{\mathbf{R}}$.

The development and calculation steps of the *strongly objective kinematic algorithm* are presented in [63], where the number of floating point operations associated with each computation is given. The objectivity of an incremental algorithm is discussed and the notion of strong objectivity introduced in order to avoid coupling between the stretching and rotational components of the incremental deformation. Such coupling can result in significant errors in cases where the rotational increment is moderately large. The strongly objective kinematic algorithm described in [63] eliminates the possibility for this coupling by requiring that the resulting stretch remains unaffected when an arbitrary input deformation is changed by a rotation.

3.2.7 Element Level Algorithm

The shape functions on each VET element are given by polynomials in the physical coordinates of the problem. The coefficients appearing in these polynomials are set so that the Kronecker-delta property of finite-element-type shape functions is realized, along with inter-element compatibility in the mean on each facet, and consistency of the quadrature rule. The remaining freedom in the coefficients is utilized to further refine the inter-element compatibility, as well as to achieve favorable smoothness on the element interior.

The following sequence of steps constitutes the element-level calculations needed to form the polynomial shape functions:

- Set an appropriate polynomial order m for the shape functions. A simple and effective choice is to choose m to be the smallest value such that the total number of terms (monomials) M is greater than the number of constraints, i.e.

$$(m+1)(m+2)(m+3)/6 > n+k+3.$$

- Equations (3.2) and (3.6) constitute a total of $n+k$ linear constraints on the selection of coefficients \mathbf{G}^a for each shape function a . An additional 3 linear constraints on \mathbf{G}^a must be introduced to ensure that volume integrals of derivatives of each shape function are evaluated exactly. This ensures that equilibrium is satisfied for a uniform stress field. These $n+k+3$ constraints can be arranged in the linear system of equations (3.9).

- Factor the constraint equation using a generalized Gaussian elimination algorithm, resulting in the form: $\mathbf{G}^a = \mathbf{B}\mathbf{g}^a + \mathbf{d}^a$, where \mathbf{B} and \mathbf{d}^a are the outputs of the factoring operation. The vector \mathbf{g}^a consists of a subset of the variables in \mathbf{G}^a , and remains free.
- Form the system of equations $\mathbf{B}^T \mathbf{A} \mathbf{B} \mathbf{g}^a = -\mathbf{B}^T \mathbf{A} \mathbf{d}^a$, which results from minimizing F_a in (3.11) with respect to the free unknowns \mathbf{g}^a . The $M \times M$ matrix \mathbf{A} in the equation above appears in each functional (3.11) and is symmetric and positive semi-definite. It is emphasized here that the matrix \mathbf{A} and the constraint matrix \mathbf{C} are independent of the shape function number a and therefore result in only one factorization each per element.
- Factor the equation system by using a generalized Gaussian elimination algorithm to solve for \mathbf{g}^a .
- Recover the shape function coefficients from $\mathbf{G}^a = \mathbf{B}\mathbf{g}^a + \mathbf{d}^a$

3.2.8 Convergence of the VETFEM

Convergence of the method can be proven based on Stummel's generalized patch test [77]. This analytical approach is briefly explained below. The rate of convergence of the VETFEM is illustrated by means of a computational example presented in Section 3.3.

As described earlier in this chapter, VETFEM shape functions are constructed as low-order polynomials in the physical coordinates of the element, instead of using an

isoparametric transformation to a parent element as employed in the conventional FEM. In the conventional FEM, basis functions are of class C^0 (continuous on the problem domain), which makes the conventional FEM a conforming method. VETFEM shape functions, on the other hand, are determined as low-order polynomials and as such are only continuous in a “weak” sense across inter-element interfaces (facets). Strict (C^0) continuity of VETFEM functions is not possible for elements of general polyhedral shape, due to the finiteness of their polynomial order. Therefore, VETFEM must be regarded as a nonconforming method. The convergence of the method when applied to variational boundary value problems is an issue that therefore needs to be addressed.

In order for nonconforming methods to converge under mesh refinement, both the so-called *approximability property* and the *closedness property* of the function spaces produced by the methods have to be satisfied. Briefly and imprecisely stated, the *approximability property* requires that the expanding approximation space contains, under mesh refinement, the exact solution in the limit. The *closedness property* requires that, under mesh refinement, the approximate problem approaches the actual problem. The analytical means for establishing these two properties for nonconforming methods is the Generalized Patch Test (GPT) of Stummel (1979) [77]. This test, unlike the engineering patch test that uses results obtained from numerical experimentation to suggest (but not prove) overall good behavior of a method, is an analytical test that is applied to the function spaces themselves.

The *approximability property* requires that arbitrary polynomial functions of a suf-

ficiently high order can be represented exactly on each element. This means that, for second order boundary value problems such as those of solid mechanics, a linear function (polynomial order one) has to be exactly represented on the elements. At the same time, the *closedness property* is achieved if the discontinuities on the inter-element facets are sufficiently well-behaved. On the basis of Stummel's GPT, in 1987 Shi established the "F-E-M" test [70], that offers a sufficient condition for convergence of nonconforming finite element methods that satisfy approximability, and which is easier to apply than the original GPT. The "F-E-M" test contains two criteria. The "F" test that is applied only to the element facets, and the "E-M" test that involves only element interiors, and is required to be applied only if the "F" test fails. In Section 3.2.2, a set of constraints on the coefficients of the VETFEM shape functions was introduced in order to ensure that shape functions have zero mean jump across inter-element facets, thus satisfying Shi's "F" test for this method. We therefore expect the VETFEM to converge under mesh refinement. The expected behavior is confirmed computationally, in Section 3.3.2.

3.3 Computational Examples

In this section, the behavior of the 3D VETFEM is verified through use of several patch tests, as illustrated in the subsection below. Then, the convergence of the method is demonstrated on a 3D boundary value problem in elastostatic. Finally, a 3D elastic-plastic boundary value problem is presented, in which the results obtained by implementing the VETFEM are compared to the ones produced by the conventional

FEM.

3.3.1 Patch Tests

In the following patch tests, an arbitrary patch of elements is used to discretize a cubic domain. The domain is then subjected to boundary conditions (either displacement or traction) which implicate a uniform strain solution on the domain. The tests are passed if the nodal displacements obtained by the approximate solution technique are consistent with the exact solution, to within arithmetic precision.

A wide variety of element patches are possible, due to the arbitrariness of the element geometry in the method. Several mesh configurations used to perform 3D patch tests with the VETFEM are shown in Figures (3.4), (3.5), (3.6) and (3.7). It is noted that the meshes used in Figure (3.6) and Figure (3.7) consist of elements that were generated as Voronoi cells using the Voronoi Generator Program developed by Mien Yip, at UC Davis, Civil Engineering and Environmental Department. These Voronoi cells perform satisfactorily as VETFEM elements. Traction boundary conditions consistent with a uniform stress field were prescribed (i.e. the domain (cube) was subjected to uniform normal traction in the z direction on its top and bottom faces), along with sufficient displacement constraints to prevent rigid body motion. In each case, the computed nodal displacements were consistent with the exact solution, to within machine precision.

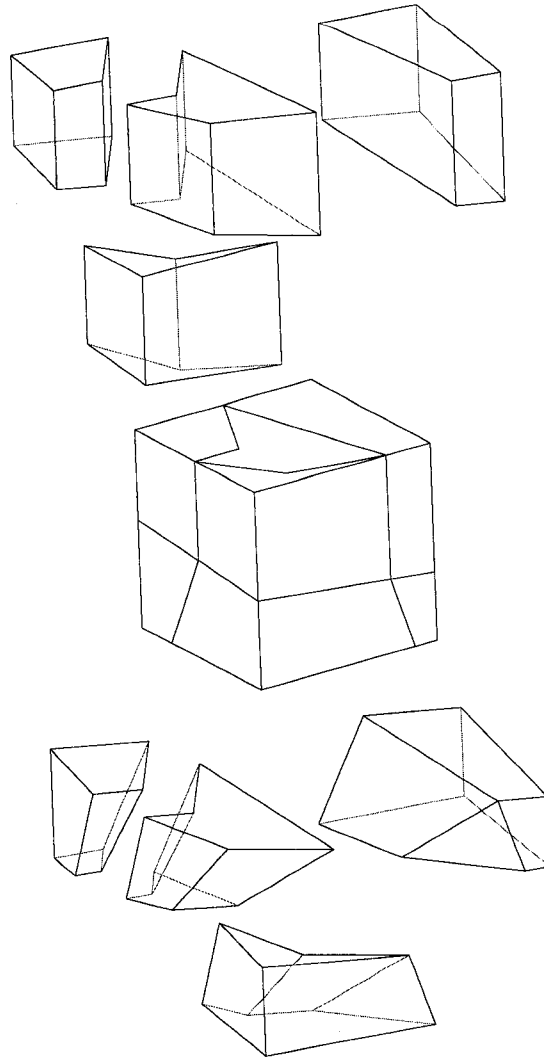


Figure 3.4: Domain Patched with Eight VETFEM Elements

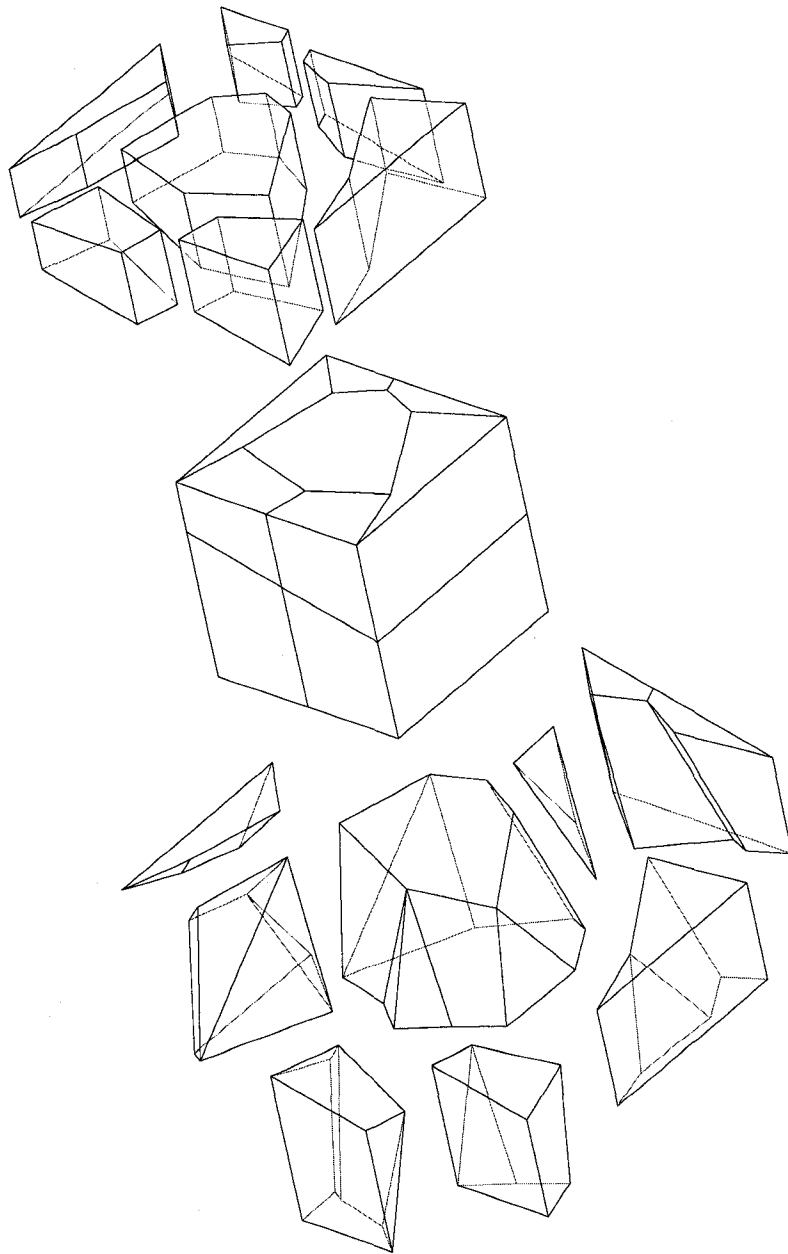


Figure 3.5: Domain Patched with Fifteen VETFEM Elements

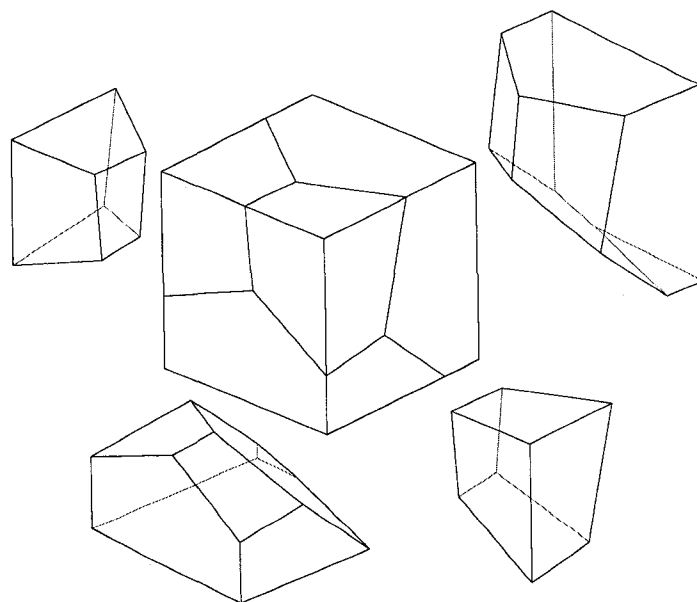


Figure 3.6: Domain Patched with Four Voronoi Cells

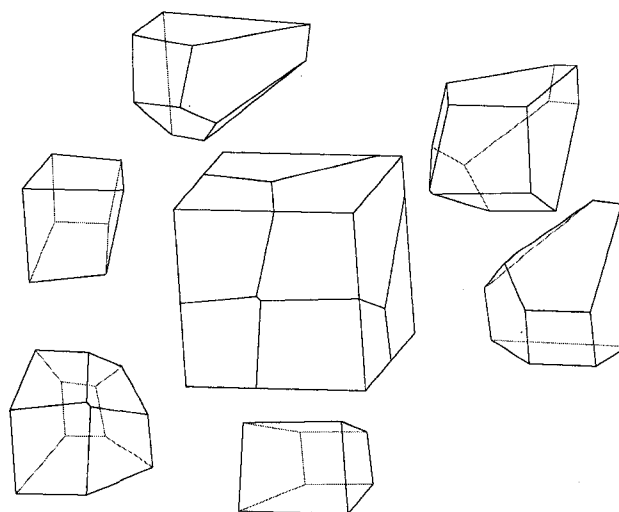


Figure 3.7: Domain Patched with Six Voronoi Cells

3.3.2 Experimental Analysis of the Rate of Convergence

In this Section, the convergence of the VETFEM method is demonstrated through numerical experimentation. Geometry and pertinent boundary conditions of the problem, as well as the loading on the problem domain, are described first. Then, the problem is analyzed under several levels of mesh refinement and various mesh realizations within each level of refinement. A reference solution was obtained by using a very fine mesh of VET elements near the problem boundary and conventional FEM elements (hexes) in the body interior. For each level of mesh refinement and realization, an error relative to the reference solution is calculated for both the displacement vector magnitude and the normal stress in the vertical direction at a fixed location in the domain. An “envelope” is then examined which bounds the error from above for all analysis.

The illustrative tri-dimensional problem in elastostatics involved the body shown in Figure 3.8. The body was generated first by forming a two-dimensional square mesh using a simple subdivision algorithm, and extruding it in the z-direction. Then, the mesh was cut by six planes (three at the top and three at the bottom). Finally, two cylindrical holes in the x and y directions were introduced. This task was performed using the “shave” algorithm created in the course of this research. Boundary conditions included fixed bottom surface and imposed uniform normal traction on the top surface of the body. The Poisson ratio was 0.3, and the displacement magnitude and the vertical-direction normal stress at the locations indicated in the Figure 3.8 were calculated. The values of displacement components at the specified location on the

top surface were obtained by linear interpolation of displacement components based on the twelve nearest-neighbor nodes, while the stresses were determined by interpolating based on twelve nearest-neighbor integration points.

The study was performed at five levels of mesh refinement, with approximately 700, 1600, 3000, 5100 and 8000 elements at each level respectively. Within each level, four different mesh realizations were obtained by slightly moving the mesh in the vertical direction before the cutting algorithm was applied. All meshes consisted of VET elements near the cutting surfaces and conventional FEM elements inside the body. However, for purposes of the element formulation, all elements were taken to be VET elements. The element aspect ratio was preserved within each level of refinement. The reference solution was determined as the average of four mesh realizations at the mesh refinement level that contained approximately 12000 elements. These meshes were obtained in the same manner as the meshes for the previously described five levels of refinement.

Figure 3.9 shows log-log plots of the magnitude of the displacement error and the stress error at the locations indicated in Figure 3.8, for all 20 analysis. In these plots, the element size was taken to be the diameter of the hex elements prior to the cutting operation. A line corresponding to the envelope of the error in the stress and displacement magnitude is also shown on the plots. The slopes are determined to be 2.2 for the displacement error and 1.4 for the error in vertical direction normal stress. It is clear that the results were in the expected range and comparable to similar studies performed by the conventional FEM with trilinear hex elements.

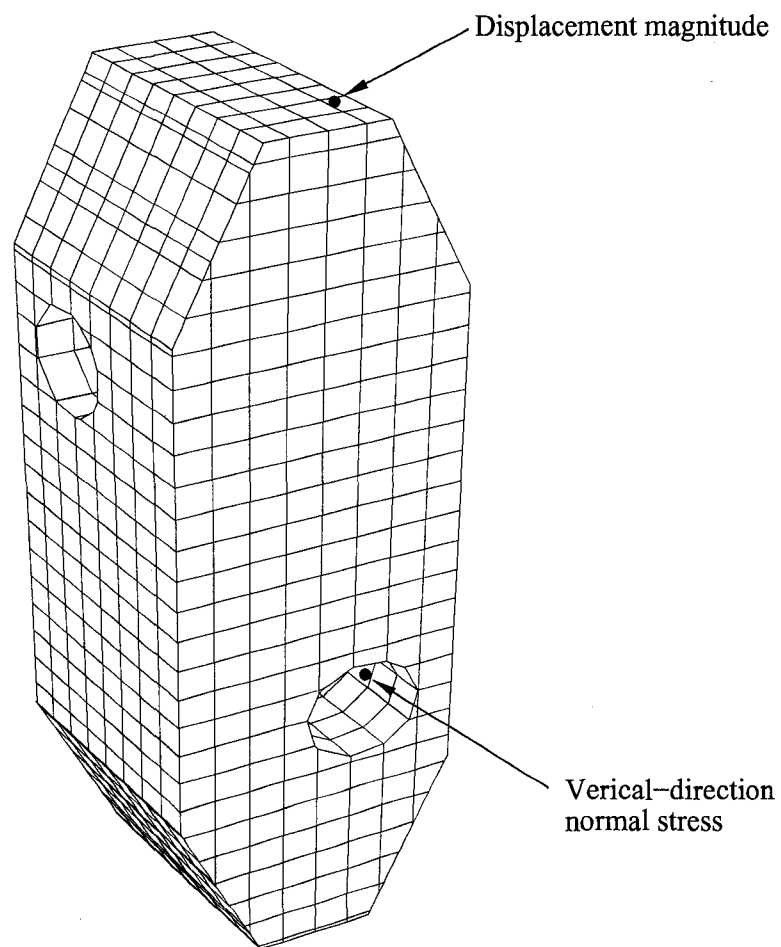


Figure 3.8: 1600 element VET mesh of a three-dimensional linearly elastic domain. The body was subjected to a uniform normal traction on its top surface with its bottom surface fixed.

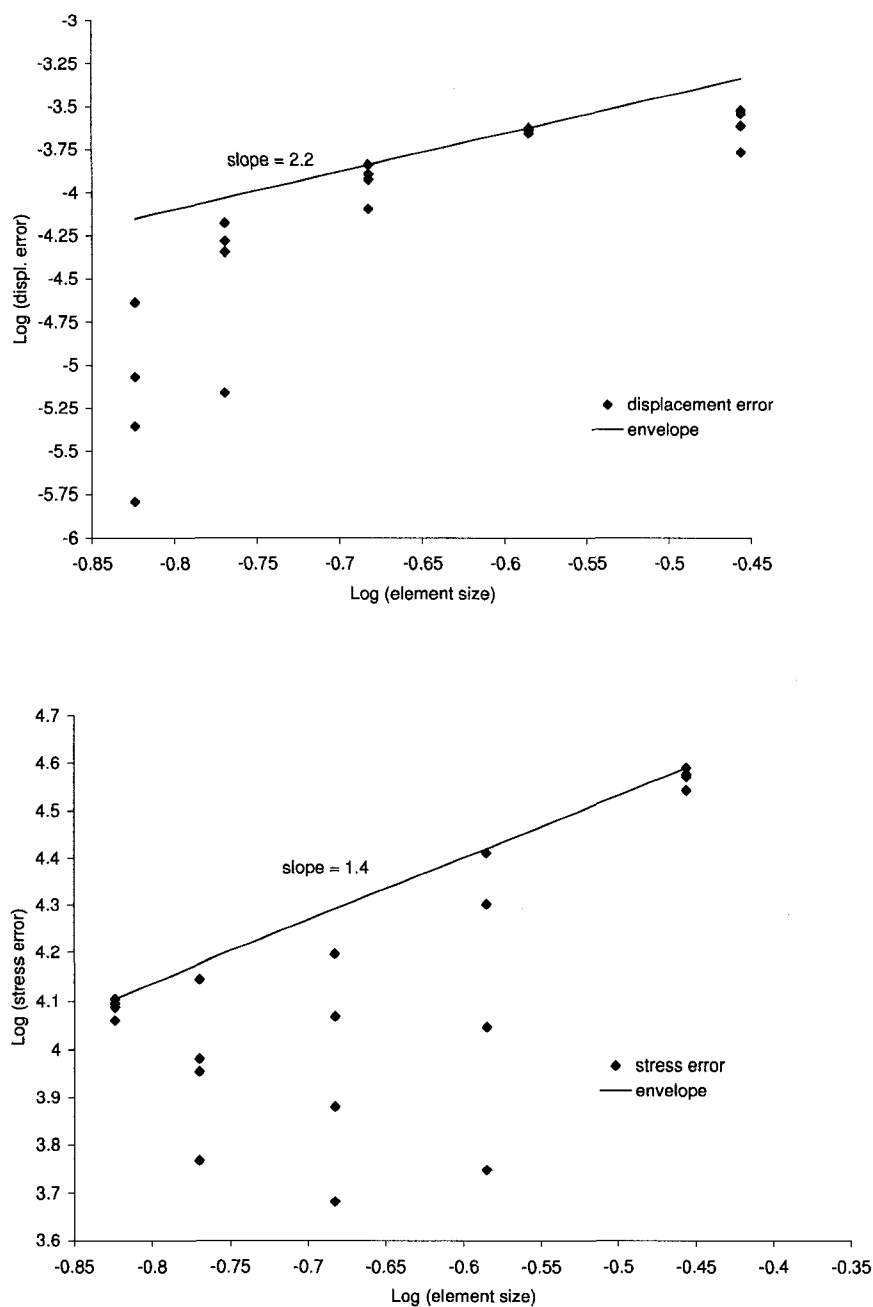


Figure 3.9: Log-log plots of error in displacement magnitude and in vertical-direction normal stress at the points indicated in Figure 3.8. At each of 5 levels of refinement, 4 different VET analysis were performed and compared with the reference solution.

3.3.3 3D Elastic-Plastic Problem

The 3D elastic-plastic boundary-value problem consisted of an elastic-plastic cube with an initially spherical void at its center. The cube was subjected to a uniform, normal traction on one pair of opposing sides. The analysis focused on one octant of the cube, with appropriate symmetry boundary conditions on three perpendicular planes that pass through the centroid of the cube. The material response was specified as the J_2 flow theory of elastoplasticity, with isotropic hardening in the following form:

$$Y = Y_0[1 + (e/\bar{e})^n]$$

In the above equation, Y is the current yield stress, Y_0 is the initial yield stress, e and \bar{e} are effective and reference plastic strain respectively, and n is the hardening exponent. The problem at hand was analyzed using values that are representative of a ductile aluminum alloy whose uniaxial stress-strain curve flattens out quickly after the initial yield: $Y_0 = 50$ ksi, $\bar{e} = 0.1$, $n = 0.15$, $E = 10,000$ ksi, $\nu = 0.3$.

Two analysis were conducted: one using 1694 VET elements, and the other with 1764 conventional eight-node hex elements (Figure (3.10)). The VET mesh was created by first generating a structured, spherically symmetric hex mesh. The desired final mesh shape was obtained by "chopping down" the initial hex mesh. The resulting mesh contained hex elements in the interior, and various polyhedral elements near the boundary as presented in Table (3.1) below:

Number of elements with:	Nodes	Facets
22 (1.3%)	4	4
232 (13.7%)	6	5
1184 (69.9%)	8	6
246 (14.5%)	10	7
10 (0.6%)	12	8

Table 3.1: Distribution of element geometries for the VET mesh shown in Figure (3.10).

In both analysis, the uniform-normal-traction boundary condition was applied monotonically over 62 increments, up to the final value of $1.928Y_0$. The solutions are compared in Figure (3.11), which shows the major- and minor-axis diameters of the void vs. load level for both the VET and the conventional FE analysis. As can be seen in the figure, these solution parameters are essentially equivalent over most of the loading range. The final values differ between the two analysis by 0.2% (minor axis) and 1.3% (major axis) near the end of the analyses, when extensive plastic yielding occurs. In general, the VET analysis exhibited slightly higher levels of deformation at a given load magnitude.

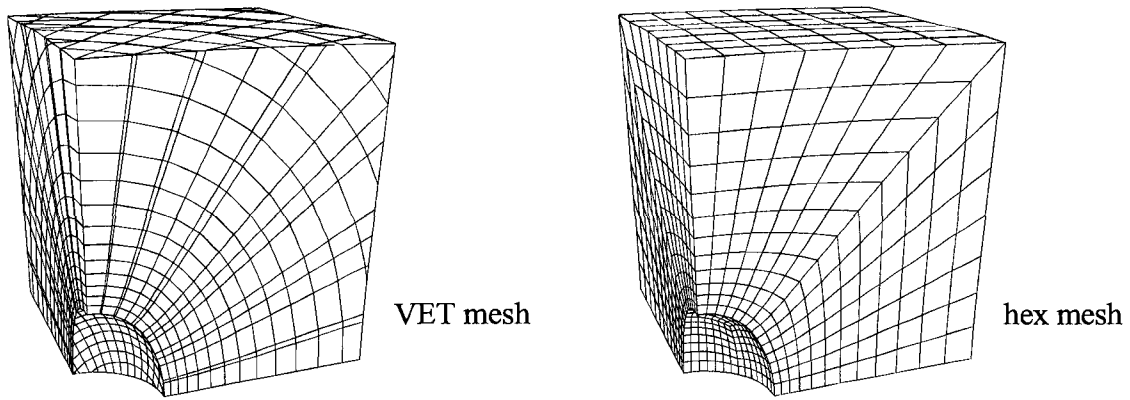


Figure 3.10: Octant of the Cube Patched with VET and hex elements

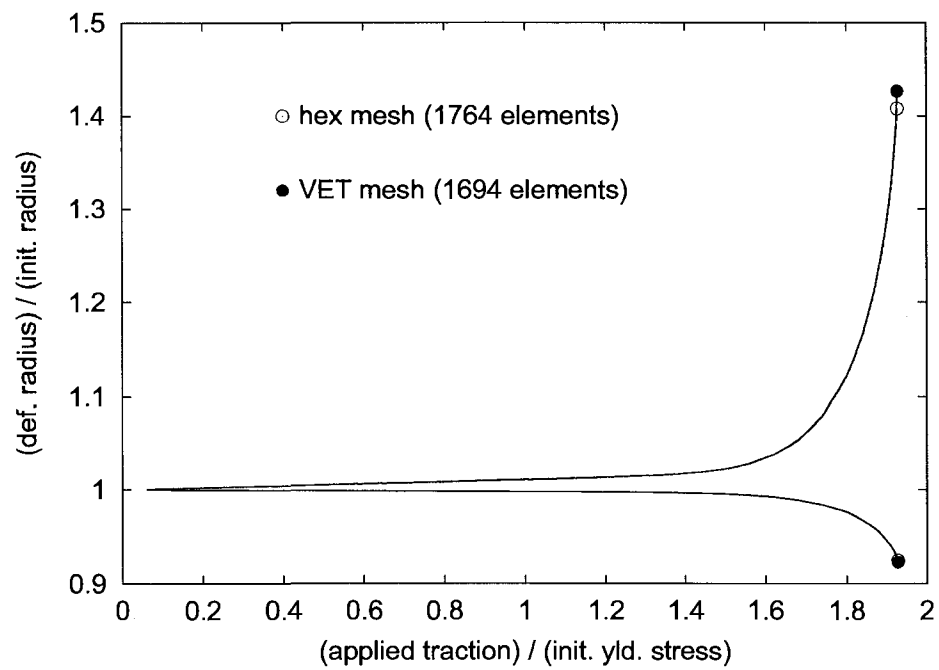


Figure 3.11: Normalized Major- and Minor- axis Deformation vs. Load Level

Chapter 4

A Discrete Data Polyhedral Finite Element Method (DDPFEM)

4.1 Introduction

The displacement based finite element approaches described in previous two chapters, namely, the VETFEM and the conventional FEM, share several powerful characteristics: both precisely and explicitly define boundary of the domain; both make it easy and convenient to impose boundary conditions; and both result in sparse and banded global equation structures. These methods also construct shape functions that are defined pointwise on each element domain, and possess a certain level of continuity (C^0 in the conventional FEM and "weak" continuity in the VETFEM that is enforced by the set of constraints on inter-element facets). Meshless (particle) methods, on the other hand, do not discretize the problem domain into elements in a finite-element-like manner. However, they still form basis functions on the problem domain by using some form of interpolation theory.

The developments in this chapter are based on the observation that, in order to solve typical boundary value problems in solid mechanics, it is necessary to determine the values of shape functions and their gradients *only at certain discrete locations (points) within the element*. Specifically, each of the aforementioned methods requires the values of shape functions at the nodes, and the values of their gradients at the element integration points. Additionally, we might also want to know the shape function values at the element boundary integration points (facets in 3D), in order to apply traction boundary conditions. The method presented here, in very preliminary form, called the Discrete Data Polyhedral FEM (DDPFEM), attempts to calculate these discrete values of shape functions and their gradients without defining shape functions pointwise on the elements. This approach clearly differs from any finite-element-like method described above and represents another significant contribution of this dissertation.

The advantage of the DDPFEM with respect to the conventional FEM is similar to that of the VETFEM described in Chapter 3 of this thesis, in the sense that topological and geometric restrictions inherent in the conventional FEM are not an issue because no isoparametric transformation to the parent element exists. Along these lines, arbitrary, polyhedral elements of the VETFEM type can be used to spatially discretize the problem domain. Furthermore, because of the unique approach that does not involve pointwise formulation of shape functions, a greater degree of concavity in the elements is possible, compared to the polyhedra used in the VETFEM formulation. As an extreme example, consider the case where facets (elements) have coincident nodes, but different

outward unit normals, such as in the analysis of arbitrary crack propagation. In the case of coincident nodes, no polynomial shape function form exists that would accommodate the Kronecker-delta property for distinct nodes with the same location. However, the DDPFEM does not use polynomials to construct shape functions pointwise, and instead pursues their discrete values only, avoiding this problem altogether.

As with any Galerkin approximation method, the DDPFEM provides both a set of suitable basis function values (only not pointwise, but at the discrete locations of the nodes and integration points) as well as a means of applying quadrature on the domain. In general, the approach of the method is to determine the values of shape functions and their gradients at the interior integration points, assuming that the values of the shape functions are known at the facet integration points. In a two dimensional setting, this would mean that, with known values of shape functions on element edges, the values of shape functions and their gradients are calculated on the interior of a planar 2D domain. In a three dimensional case, this approach for the two dimensional case would represent just the first step in which the values of shape functions and their gradients would be solved on the 2D planar facets of a 3D element. The second step would be completely analogous, and would consist of determining the values of shape functions and their gradients at the integration points in the interior of the element, with known (already calculated in the previous step) values on the element's boundary (facets).

4.2 Constraints and Interior Smoothness Optimization

To pursue the previously described process, we require that the interior, calculated values of shape functions φ and their gradients \mathbf{g} are compatible with the known values at the domain boundary. This requirement produces a set of constraints on the interior shape function values and their gradients:

$$\int_{\omega} (g_i - \varphi_{,i}) m_i dV = 0 \quad , \quad \mathbf{m} \text{ a vector-valued weight function}$$

Applying integration by parts and the divergence theorem results in:

$$\begin{aligned} \int_{\omega} [g_i m_i - (\varphi m_i)_{,i} + \varphi m_{i,i}] dV &= 0 \\ \int_{\omega} (g_i m_i + \varphi m_{i,i}) dV &= \int_{\partial\omega} \varphi m_i n_i da \end{aligned} \quad (4.1)$$

In addition to the above constraints, we desire that the shape function gradients be chosen in a manner that optimizes smoothness on the element interior. This optimization must be consistent with the above stated constraints, and is realized by minimizing the following functional:

$$\mathcal{F} \equiv \int_{\omega} g_i g_i dV . \quad (4.2)$$

This description briefly introduces the approach to formulating the DDPFEM shape functions. With this in mind, the problem can be precisely stated as follows:

Let g_i , and φ be a typical shape function gradient and shape function values. Let g_i^α , φ^α be discrete values of g_i , and φ associated with points within the element domain ω or its boundary $\partial\omega$. Some of these points are integration points (IPs) on ω , and others

are IPs on facets $\Gamma \subseteq \partial\omega$.

We wish to choose g_i^α , and φ^α so that a discretized version of the functional (4.2) is minimized, subject to discretized constraints (4.1). Now, let each IP be assigned an index α , and define sets of IPs relevant to each quadrature rule as follows:

B : set of IPs for quadrature on ω

b^k : set of IPs for quadrature on Γ_k , where: $\partial\omega = \bigcup_k \Gamma_k$

Next, we associate with each integer in B and b^k a real number - w , the corresponding quadrature weight. Then we have:

$B = \{(\alpha_1, w_1), (\alpha_2, w_2), \dots\}$; with a similar expression for b^k

where α_i , and w_i represent global IP index and corresponding weight respectively.

We could say that B and b^k define the integration rules on the domain interior and boundary.

The discrete forms of the equations (4.2) and (4.1) above are used to state the problem as follows:

Minimize the functional \mathcal{F} :

$$\mathcal{F} = \sum_{(\alpha, w) \in B} g_i^\alpha g_i^\alpha w \quad (4.3)$$

subject to constraints:

$$\sum_{(\alpha, w) \in B} [g_i^\alpha m_i^{\alpha I} w + \varphi^\alpha m_{i,i}^{\alpha I} w] = \sum_k \sum_{(\alpha, w) \in b_k} \varphi^\alpha (m_i^{\alpha I} + \varepsilon_i^{\alpha I}) n_i^{\alpha w} . \quad (4.4)$$

Here, $I = 1, \dots, N$ is the number of constraints; $m_i^{\alpha I}$ is a polynomial vector-valued test (weight) function number I , evaluated at the integration point number α .

Next, it is noted that, for g_i^α and φ^α consistent with a fixed and arbitrary linear function, the expression (4.4) should hold for all $m_i^{\alpha I}$. Since it is generally impractical to devise integration rules that guarantee this for all polynomial $m_i^{\alpha I}$, the adjustment vector-valued polynomial functions $\varepsilon_i^{\alpha I}$'s are introduced at the boundary integration points, and are determined through a constrained minimization process as indicated below.

Minimize the expression:

$$\int_{\partial\omega} \varepsilon_i \varepsilon_i \, da \quad (4.5)$$

subject to constraints:

$$\begin{aligned} \int_{\omega} (m_i + \varepsilon_i)_{,i} \, dV &= \int_{\omega} m_{i,i} \, dV \\ \int_{\omega} [(m_i + \varepsilon_i) x_j]_{,i} \, dV &= \int_{\omega} (m_i x_j)_{,i} \, dV, \end{aligned}$$

which emerge from the requirement that (4.4) holds for arbitrary linear (φ, g_i) .

Applying the divergence theorem, the constraints become:

$$\begin{aligned} \int_{\partial\omega} \varepsilon_i n_i \, da &= \int_{\omega} m_{i,i} \, dV - \int_{\partial\omega} m_i n_i \, da \\ \int_{\partial\omega} \varepsilon_i x_j n_i \, da &= \int_{\omega} (m_i + x_j m_{i,i}) \, dV - \int_{\partial\omega} m_i x_j n_i \, da \end{aligned} \quad (4.6)$$

All integrals in the expressions (4.5) and (4.6) above are evaluated using the quadrature rules B and b^k . It should be noted that the only term that appears in equations (4.4) and (4.6) is $\varepsilon_i n_i$. Therefore, we can simply replace $\varepsilon_i n_i$ with another number, say ε , while replacing (4.5) with $\int_{\partial\omega} \varepsilon^2 \, da$.

The first twelve $m^{\alpha I}$'s constitute the full set of constant and linear vector-valued test

functions. For these test functions, the correction term $\varepsilon = 0$, because integration rules can always be formulated that exactly integrate arbitrary linear functions on arbitrary polygons and polyhedra. The higher-order $\mathbf{m}^{\alpha I}$'s are chosen to introduce the correct number of linearly independent expressions in φ^α , ($\alpha \in B$), see (4.4).

To illustrate the selection of second and higher order m_i 's, first consider that the following linearly independent \mathbf{m} 's all produce $m_{i,i} = x_1$:

$$\frac{1}{2} (x_1^2, 0, 0), \quad (0, x_1 x_2, 0), \quad (0, 0, x_1 x_3).$$

Also, note that many m_i 's produce $m_{i,i} = 0$ as the result. With these observations in mind, the best way to choose the m_i 's can be illustrated by the following example:

Let us seek a vector-valued function whose derivatives $m_{i,i}$ equal x_1 , which yields the x_1 - moment of φ in (4.4). Let $m_i = a_\alpha p_i^\alpha$, where p_i^1, p_i^2, p_i^3 are the above three vectors, all of which give $m_{i,i} = x_1$. The problem of choosing the corresponding \mathbf{m} can be stated as:

Determine a_α so that the expression:

$$\int_\omega \sum_\alpha (a_\alpha)^2 p_i^\alpha p_i^\alpha dV \quad , \quad \text{is a minimum}$$

subject to:

$$\sum_\alpha a_\alpha = 1$$

This simple constrained quadratic minimization problem has a unique solution that can be written in closed form. It requires the calculation of integrals of monomials on ω , which was described in Chapter 3. Now, in the constraint equations (4.4) the φ^α values on the boundary are known. These values are used in the expression to form the nonzero

right-hand side of (4.4). The interior, unknown values of φ^α , together with the values of g_i^α , remain unknown in the system of equations. The calculation proceeds in a manner identical to that described in 3.2.7, by which the constraints (4.4) are employed to eliminate some unknowns, followed by the minimization of (4.3) to uniquely determine all φ and g_i values at interior integration points.

4.3 Integration Rule

A numerical integration rule on an arbitrary domain ω can in general be expressed by the relation:

$$\int_{\omega} f(\mathbf{x}) dV = \sum_{\alpha} w_{\alpha} f(\mathbf{x}_{\alpha}) \quad (4.7)$$

Requiring that an arbitrary linear function $f(\mathbf{x}) = \mathbf{a} \cdot \mathbf{x} + b$ be exactly integrated, the above expression (4.7) requires:

$$\begin{aligned} \mathbf{a} \cdot \int_{\omega} \mathbf{x} dV + b \int_{\omega} dV &= \sum_{\alpha} w_{\alpha} (\mathbf{a} \cdot \mathbf{x}_{\alpha} + b) \quad , \quad \forall \mathbf{a}, b \\ \mathbf{a} \cdot \left[|\omega| \bar{\mathbf{x}} - \sum_{\alpha} w_{\alpha} \mathbf{x}_{\alpha} \right] + b \left[|\omega| - \sum_{\alpha} w_{\alpha} \right] &= 0 \quad , \quad \forall \mathbf{a}, b \end{aligned} \quad (4.8)$$

where w_{α} represents the weight at the integration point α , and $\bar{\mathbf{x}}$ is the centroid of ω .

In order for the above relations to hold, both parenthesis must be zero:

$$\sum_{\alpha} w_{\alpha} = |\omega| \quad \text{and} \quad \sum_{\alpha} w_{\alpha} (\bar{\mathbf{x}} - \mathbf{x}_{\alpha}) = 0$$

The expressions above constitute the four constraints that have to be satisfied in order for the integration rule to exactly integrate any linear function. These constraints can

also be written as:

$$\begin{aligned}\sum_{\alpha} w_{\alpha} \mathbf{x}_{\alpha} &= |\omega| \bar{\mathbf{x}} \\ \sum_{\alpha} w_{\alpha} &= |\omega|\end{aligned}\tag{4.9}$$

Now, suppose that we want our integration rule to accurately integrate another class of functions besides linear functions. Specifically, let $f = f_{\alpha}$ be a piecewise constant function on a so-called “tributary region” $\mathbf{x} \in \omega_{\alpha}$, where $\omega = \bigcup_{\alpha} \omega_{\alpha}$, and let e be the relative error as defined in the expression below:

$$\begin{aligned}e &\equiv \left| \sum_{\alpha} w_{\alpha} f(\mathbf{x}_{\alpha}) - \int_{\omega} f(\mathbf{x}) dV \right| / \|f\| \\ &= \left| \sum_{\alpha} w_{\alpha} f_{\alpha} - \sum_{\alpha} |\omega_{\alpha}| f_{\alpha} \right| / \|f\| \\ &= \left| \sum_{\alpha} f_{\alpha} (w_{\alpha} - |\omega_{\alpha}|) \right| / \left[\sum_{\alpha} |f_{\alpha}|^2 |\omega_{\alpha}| \right]^{1/2}\end{aligned}\tag{4.10}$$

where:

$$\|f\|^2 = \sum_{\alpha} |f_{\alpha}|^2 |\omega_{\alpha}| \text{ is the } L^2 \text{ norm of the piecewise constant function } f.$$

It can be seen that the error e above completely vanishes in the case when the weights w_{α} correspond to the volumes $|\omega_{\alpha}|$ of the tributary regions. This satisfies the linear-exactness criterion (4.9) if the integration point locations coincide with the locations of the centroids of the tributary regions. However, in certain instances, we would like to have the flexibility to place the integration points in locations that do not correspond to the centroids of the tributary regions. For example, in a 3D setting, we would prefer to place integration points on a planar facet along the edges of the facet to the

extent possible, and not within the facet interior, in order to avoid having to define the shape functions themselves at many interior points of the facet. Therefore, we wish to minimize the error (4.10). This leads to another constrained minimization process described below, in order to minimize the error e and satisfy the four constraints (4.9) that pertain to the integration of linear functions on the domain.

Let us define an average error E over the set of functions f with norm c :

$$E = \frac{1}{c} \int_{\|f\|=c} e \, d\Omega \quad (4.11)$$

Also, let Λ be an N -dimensional Euclidean space with coordinates $\{f_1, f_2, \dots, f_N\}$. A one-to-one correspondence exists between points in Λ and functions f . Then, the domain Ω above represents a hypersphere contained in Λ . Now, we write the L^2 norm of the piecewise constant function f as a dot product:

$$\begin{aligned} \|f\|^2 = \langle f, f \rangle &= \int_{\omega} f^2 \, dV = \int_{\omega} \left(\sum_{\alpha} f_{\alpha} \zeta_{\alpha} \right) \left(\sum_{\beta} f_{\beta} \zeta_{\beta} \right) dV \\ &= \int_{\omega} \sum_{\alpha} f_{\alpha}^2 \zeta_{\alpha}^2 \, dV = \sum_{\alpha} f_{\alpha}^2 |\omega_{\alpha}| \end{aligned} \quad (4.12)$$

where ζ_{α} is the indicator function that has value of 1 on ω_{α} and 0 everywhere else.

Now, let us define the parameters ξ_{α} and η_{α} as:

$$\begin{aligned} \xi_{\alpha} &= f_{\alpha} |\omega_{\alpha}|^{1/2} && \text{no sum} \\ \text{and } \eta_{\alpha} &= g_{\alpha} |\omega_{\alpha}|^{1/2} && \text{no sum} \end{aligned} \quad (4.13)$$

We observe that the inner product $\langle f, g \rangle$ can be expressed as:

$$\langle f, g \rangle = \sum_{\alpha} \xi_{\alpha} \eta_{\alpha}$$

where ξ_α and η_α are Cartesian coordinates of N -dimensional Euclidean space \mathbb{R}^N , and every piecewise-constant function on ω can be put into unique correspondence with points in \mathbb{R}^N . The error e in the expression (4.10) now becomes:

$$e = \frac{1}{\|f\|} \left\{ \left[\sum_{\alpha} f_{\alpha} (w_{\alpha} - |\omega_{\alpha}|) \right] \left[\sum_{\beta} f_{\beta} (w_{\beta} - |\omega_{\beta}|) \right] \right\}^{1/2}$$

Substituting the definitions for ξ_α and η_α from (4.13), we have:

$$e = \frac{1}{\|f\|} \left\{ \sum_{\alpha} \sum_{\beta} \xi_{\alpha} \xi_{\beta} \frac{w_{\alpha} - |\omega_{\alpha}|}{|\omega_{\alpha}|^{1/2}} \frac{w_{\beta} - |\omega_{\beta}|}{|\omega_{\beta}|^{1/2}} \right\}^{1/2} \quad (4.14)$$

If we integrate the above expression (4.14) on the unit sphere in \mathbb{R}^N as indicated in (4.11), the symmetry of the sphere will cause all mixed products to cancel, and the only terms that remain will be the square terms, as follows:

$$E = \frac{1}{\|f\|} \int_{\xi \cdot \xi = 1} \sum_{\alpha} \xi_{\alpha}^2 \frac{(w_{\alpha} - |\omega_{\alpha}|)^2}{|\omega_{\alpha}|} d\Omega$$

Now, since the integral $\int_{\xi \cdot \xi = 1} \xi_{\alpha}^2 d\Omega$ is independent of the integration point number α , the average error E can be written as:

$$E \sim \sum_{\alpha} \frac{(w_{\alpha} - |\omega_{\alpha}|)^2}{|\omega_{\alpha}|} = \sum_{\alpha} \left\{ \frac{1}{|\omega_{\alpha}|} w_{\alpha}^2 - 2w_{\alpha} + |\omega_{\alpha}| \right\}$$

The third term in the bracket above is simply $|\omega|$ (the total volume), and is independent of the integration point number α . It can, therefore, be omitted for the purpose of minimizing the expression for E above. The second term (w_{α}) , after summing over α , also must equal the volume of the domain $|\omega|$, and can be dropped from the expression as well. The remaining form of the average error E then becomes:

$$E \sim \sum_{\alpha} \frac{1}{|\omega_{\alpha}|} w_{\alpha}^2 \quad (4.15)$$

Once the integration-point locations have been set, the uncoupled quadratic function (4.15) is minimized with respect to the weights w_α , subject to the constraints (4.9). This problem results in a small linear system of equations for the weights w_α .

In summary, the DDPFEM is being developed with the goal of providing a numerical tool with a greatly increased level of flexibility in analyzing problems involving complex and evolving geometry. The DDPFEM applies to conventional element geometries as well as to polyhedral elements of VET type, while allowing for an even greater degree of element concavity than that in the VETFEM. The main advantage of the DDPFEM lies in the manner in which the values of the shape functions and their gradients are determined. Unlike the conventional FEM and VETFEM, the DDPFEM does not construct basis functions pointwise on the domain. Instead, discrete values of shape functions and their gradients are calculated through a constrained minimization process that ensures their compatibility with the given values at the boundary, and smoothness over the element interior. Also, it can be seen that the values of the shape functions, as well as the quadrature rules, depend solely on the geometry of the element facets. Hence, the values of the shape functions on inter-element facets have to be equal for the neighboring elements, which automatically satisfies Shi's "F" test [70], as described in Section 3.2.8.

Further research is being conducted toward completing the DDPFEM integration rule described in Section 4.3 of this dissertation. An important part of this work is the calculation of the volumes of the "tributary regions" associated with the integration

points (and corresponding weights w_α). The research efforts in this direction will propose a different approach to the determination of the convex “tributary region” volumes and first moments in the sense that, instead of resolving the associated intersections geometrically, the problem is approximated by a constrained optimization problem, which can be efficiently solved computationally.

Chapter 5

Future Work

The conventional FEM is still the leading computational technique for approximately solving solid mechanics problems, and as a result there is a broad spectrum of related software available. In many cases triangular elements in 2D and tetrahedral elements for 3D regions are used, which typically exhibit poor behavior near the incompressibility limit. Quadrilateral 2D elements and hexahedral 3D elements are preferred in solid mechanics, but algorithms for mesh generation using these elements involve considerable logical complexity as well as human intervention.

The VETFEM, on the other hand, encompasses all of the powerful characteristics of the conventional FEM and, at the same time, imposes much weaker topological and geometric restrictions on the mesh design. The element-level calculations of the polynomial shape functions in the VETFEM are more extensive than those of the conventional FEM. However, these calculations increase only linearly with the size of the problem. It is also important to emphasize that the VETFEM is completely compatible with the standard FEM, and that both methods can be applied simultaneously to the same

problem in any spatial arrangement. The interface between the two (VETFEM and FEM) meshes does not require any special consideration. The DDPFEM introduced in Chapter 4 assigns values of shape functions and their gradients at discrete locations (interior integration points) of the problem domain without defining shape functions pointwise on the interior of the domain.

The need for generating quality three dimensional meshes, especially for problems involving complex and/ or evolving geometry such as crack propagation and metal cutting, engenders the need for a mesh-generating scheme that can capitalize on the flexibility of the VETFEM within an acceptable computing time, as well as with minimal or no burden on human resources. Further research effort in this direction seems to be prudent and justified. Recent research efforts in this direction by professor M.M. Rashid and T. Dinar, PhD. candidate at UC Davis, resulted in a novel and convenient mesh generation process where a structured hexahedral mesh is generated first in the bounding-box of the analysis domain. This mesh is then intersected with the domain boundary (facets) such that the exterior elements are cut, resulting in a mesh composed of general polyhedral elements near the boundary and hexahedral elements on the interior, which is suitable for the VETFEM.

Further, in the course of an analysis that involves large deformations and/ or evolving geometry, the mesh (elements) might become so distorted that the solution accuracy deteriorates rapidly, and the analysis ultimately terminates. It is in these instances that the deformed domain must be remeshed in order to proceed with the analysis. Concur-

rently with the mesh generation process, the transfer of information characterizing the material state from the old mesh to the new one prior to continuing the problem analysis also presents a current challenge in solving real-life 3D problems in solid mechanics (especially for non linear problems). The problem of material state remapping computationally consists of determining the intersections between so-called “tributary regions” associated with a point (i.e. node, integration point) in the new mesh, with regions in the old mesh that correspond with old mesh integration points or nodes. It can be seen that both the remeshing and material state data remapping tasks involve dealing with complicated computational geometry issues. Recent research by M.M. Rashid and T. Dinar at UC Davis proposes a different computational approach to the determination of the volumes of intersection of arbitrarily arranged convex “tributary regions”. Instead of geometrically resolving the associated intersections, the problem is approximated by a constrained optimization problem, which can be efficiently solved computationally.

In light of the discussion above, future work on improving and applying the methods described in this dissertation should focus on integrating the aforementioned tasks of successive mesh redesign and material state remapping into a single software platform. Our optimism that the presented methods can greatly facilitate the burdensome process of continuous remeshing of evolving problem domains is also based on the fact that conventional FEM meshes are entirely suitable for use as VETFEM meshes. Accordingly, the possibility of using the VETFEM in transition regions, between regions that can easily be meshed with conventional finite elements, is a particularly appealing feature

of the overall scheme. Even though the extension of the VETFEM to 3D engenders a restriction to planar element facets, the automatic generation of 3D VETFEM meshes is significantly simpler than for high quality conventional meshes (especially if hexahedral elements are required).

We firmly believe that the features inherent in the VETFEM have much more to offer in comparison to various new and modified finite-element-like approximation schemes, and are especially useful for problems that include very complex geometry, adaptive remeshing, or crack propagation. Therefore, as a part of our future endeavor, we will also work on improving the user interface with this software and attempt to familiarize various industries with all valuable characteristics and analytic possibilities of these developments.

Bibliography

- [1] P. Areias, T. Belytschko. Analysis of three-dimensional crack initiation and propagation using the extended finite element method. *International Journal for Numerical Methods in Engineering*, 63:760–788, 2005.
- [2] P. Areias, T. Belytschko. Non-linear analysis of shells with arbitrary evolving cracks using XFEM. *International Journal for Numerical Methods in Engineering*, 62:384–415, 2004.
- [3] S.N. Atluri, T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics.. *Computational Mechanics*, 22:117–127, 1998.
- [4] S. W. Attaway, M. W. Heinstein, and J. W. Swegle. Coupling of smooth particle hydrodynamics with the finite element method. *Nuclear engineering and design : an international journal devoted to the thermal, mechanical and structural problems of nuclear energy*, 150:199–205, 1994.
- [5] I. Babuska, U. Banerjee, J.E. Osborn. Survey of meshless and generalized finite element methods: a unified approach. *Acta Numerica*, 12:1–125, 2003.
- [6] I. Babuska, G. Caloz, J. Osborn. Special finite element methods in a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31:945–981, 1994.
- [7] I. Babuska, J.M. Melenk. The partition of unity method. *International Journal for Numerical Methods in Engineering*, 40:727–758, 1997.
- [8] I. Babuska, M. Suri. Locking effects in the finite element approximation of elasticity problems. *Numerische Mathematik*, 62, 1992.
- [9] Klaus-Jürgen Bathe. *Finite Element Procedures in Engineering Analysis*. Prentice Hall Inc., 1996. ISBN 0-13-301458 - 4.
- [10] S. Beissel, T. Belytschko. Nodal integration of the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139:49–74, 1996.

- [11] V.V. Belikov, V.D. Ivanov, V.K. Kontorovich, S.A. Korytnik, A. Yu Semenov. The non-Sibsonian interpolation: a new method of interpolation of the values of a function on an arbitrary set of points. *Computational Mathematics and Mathematical Physics*, 37(1):9–15, 1997.
- [12] T. Belytschko, T. Black. Elastic crack growth in finite elements with minimal remeshing. *International Journal for Numerical Methods in Engineering*, 45(5):601–620, 1999.
- [13] T. Belytschko, M. Fleming. Smoothing, enrichment and contact in the element-free Galerkin method. *Computers and Structures*, 71(2):173–195, 1999.
- [14] T. Belytschko, Y. Krongauz, J. Dolbow, C. Gerlach. On the Completeness of Meshfree Particle Methods. *International Journal for Numerical Methods in Engineering*, 43:785–819, 1998.
- [15] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless Methods: An Overview And Recent Developments. *Computer Methods in Applied Mechanics and Engineering*, 139:3–47, 1996.
- [16] T. Belytschko, Y. Y. Lu, and L. Gu. Element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.
- [17] T. Belytschko, J.S. Ong, W.K. Liu, J.M. Kennedy. Hourglass control in linear and nonlinear problems. *Computer Methods in Applied Mechanics and Engineering*, 43:254–276, 1984.
- [18] T. Belytschko and M. Tabbara. Dynamic fracture using element-free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 39:923–938, 1996.
- [19] W. Benz and E. Asphaug. Simulations of Brittle Solids using Smooth Particle Hydrodynamics. *Computer Physics Communications*, 87:253–265, 1995.
- [20] J. Bonet, S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulation. *International Journal for Numerical Methods in Engineering*, 47:1189–1214, 1999.
- [21] J. Braun, M. Sambridge. A numerical method for solving partial differential equations on highly irregular evolving grids. *Nature*, 376:655–660, 1995.
- [22] D. Bueche, N. Sukumar, B. Moran. Dispersive properties of the natural element method. *Computational Mechanics*, 25:207–219, 2000.
- [23] P. Chadwick. *Continuum Mechanics: Concise Theory and Problems*. Dover Publications, Inc., 1999.

- [24] J.S. Chen, W.K. Liu. Meshfree Methods: Recent advances and new applications. *Computer Methods in Applied Mechanics and Engineering*, 193(12-14):933–1321, 2004.
- [25] J. S. Chen, C. Pan, C. T. Wu, and W. K. Liu. Reproducing Kernel Particle Methods for Large Deformation Analysis of Nonlinear Structures. *Computer Methods in Applied Mechanics and Engineering*, 139:195–227, 1996.
- [26] J.S. Chen, H.P. Wang. New boundary condition treatments for meshless computation of contact problems. *Computer Methods in Applied Mechanics and Engineering*, 187:441–468, 2000.
- [27] J.S. Chen, C.T. Wu, S. Yoon, W. You. A stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.
- [28] H.-J. Chung, T. Belytschko. An Error Estimate in the EFG Method. *Computational Mechanics*, 21:91–100, 1998.
- [29] L. W. Cordes and B. Moran. Treatment of material discontinuity in the element-free Galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139:75–89, 1996.
- [30] E. Cueto, B. Calvo, M. Doblare. Modeling three-dimensional piece-wise homogeneous domains using the α -shape based Natural Element Method. *International Journal for Numerical Methods in Engineering*, 54(6):871–897, 2002.
- [31] E. Cueto, N. Sukumar, B. Calvo, M.A. Martinez, J. Cegonino, M. Doblare. Overview of recent advances in natural neighbor Galerkin methods. *Archives of Computational Methods in Engineering*, 10:307–384, 2003.
- [32] C. Daux, N. Moes, J. Dolbow, N. Sukumar, T. Belytschko. Arbitrary branched and intersecting cracks with the extended finite element method. *International Journal for Numerical Methods in Engineering*, 48:1741–1760, 2000.
- [33] E.A. De Souza Neto, D. Peric, M. Dutko, D.R.J. Owen. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal of Solids and Structures*, 33:3277–3296, 1996.
- [34] C.R. Dohrmann, S.W. Key, M.W. Heinstein. A method for connecting dissimilar finite element meshes in two dimensions. *International Journal for Numerical Methods in Engineering*, 48:655–678, 2000a.
- [35] C.R. Dohrmann, S.W. Key, M.W. Heinstein. Methods for connecting dissimilar three-dimensional finite element meshes. *International Journal for Numerical Methods in Engineering*, 47:1057–1080, 2000b.

- [36] C.R. Dohrmann, M. M. Rashid. Polynomial approximation of shape function gradients from element geometries. *International Journal for Numerical Methods in Engineering*, 53:945–958, 2002.
- [37] J. Dolbow, T. Belytschko. Numerical integration of the Galerkin weak form in meshfree methods. *Computational Mechanics*, 23:219–230, 1999.
- [38] J. Dolbow, T. Belytschko. Volumetric Locking in the Element Free Galerkin Method. *International Journal for Numerical Methods in Engineering*, 46(6):925–942, 1999.
- [39] J. Dolbow, N. Moes, T. Belytschko. An extended finite element method for modeling crack growth with frictional contact. *Computer Methods in Applied Mechanics and Engineering*, 190(51-52):6825–6846, 2001.
- [40] J. Dolbow, N. Moes, T. Belytschko. Discontinuous enrichment in finite elements with a partition of unity method. *Finite Elements in Analysis and Design*, 36:235–260, 2000.
- [41] C.A. Duarte, J.T. Oden. H-p clouds - an h-p meshless method. *Numerical Methods for Partial Differential Equations*, 12:673, 1996.
- [42] D.P. Flanagan, T. Belytschko. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17:679–706, 1981.
- [43] T.P. Fries, T. Belytschko. The intrinsic XFEM: a method for arbitrary discontinuities without additional unknowns. *International Journal for Numerical Methods in Engineering*, 68:1358–1385, 2006.
- [44] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics: Theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389, 1977.
- [45] D. Gonzalez, E. Cueto, M.A. Martinez, and M. Doblare. Numerical integration in natural neighbor Galerkin methods. *International Journal for Numerical Methods in Engineering*, 60:2077–2104, 2004.
- [46] Thomas J. R. Hughes. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, Inc., 2000.
- [47] T.J. R. Hughes. *The Finite Element Method*. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.
- [48] S.R. Idelsohn, N. Calvo, E. Onate. Polyhedralization of an arbitrary 3D point set. *Computer Methods in Applied Mechanics and Engineering*, 192(22-24):2649–2667, 2003.

- [49] S.R. Idelsohn, E. Onate, N. Calvo, F. Del Pin. The meshless finite element method. *International Journal for Numerical Methods in Engineering*, 58:893–912, 2003.
- [50] Y. Krongauz, T. Belytschko. Enforcement of essential boundary conditions in meshless approximations using finite elements. *Computer Methods in Applied Mechanics and Engineering*, 131:133–145, 1996.
- [51] P. Krysl, T. Belytschko. The Element Free Galerkin Method for Dynamic Propagation of Arbitrary 3-D Cracks. *International Journal for Numerical Methods in Engineering*, 44(6):767–800, 1999.
- [52] S.H. Lee, J.H. Song, Y.C. Yoon, G. Zi, T. Belytschko. Combined extended and superimposed finite element method for cracks. *International Journal for Numerical Methods in Engineering*, 59:1119–1136, 2004.
- [53] S. Li, W.K. Liu. Meshfree and particle methods and their applications. *Applied Mechanics Reviews*, 55(1):1–34, 2002.
- [54] W.K. Liu, T. Belytschko, H. Chang. An arbitrary Lagrangian-Eulerian finite element method for path-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 58:227–246, 1986.
- [55] W. K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical Methods in Engineering*, 20:1081–1106, 1995.
- [56] J.M. Melenk, I. Babuska. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.
- [57] N. Moes, T. Belytschko. Extended finite element method for cohesive crack growth. *Engineering Fracture Mechanics*, 69(7):813–833, 2002.
- [58] N. Moes, J. Dolbow, T. Belytschko. A Finite Element Method for Crack Growth without Remeshing. *International Journal for Numerical Methods in Engineering*, 46(1):131–150, 1999.
- [59] B. Nayroles, G. Touzot, and P. Villon. Generalizing the finite element method: Diffuse approximation and diffuse elements. *Computational Mechanics*, 10:307–318, 1992.
- [60] J.F. Peters, E. Heymsfield. Application of the 2-D constant strain assumption to FEM elements consisting of an arbitrary number of nodes. *International Journal of Solids and Structures*, 40:143–159, 2003.
- [61] J.-P. Ponthot, T. Belytschko. Arbitrary Lagrangian-Eulerian Formulation for Element-Free Galerkin Method. *Computer Methods in Applied Mechanics and Engineering*, 152:19–46, 1998.

- [62] P.W. Randles, L.D. Libersky, A.G. Petschek. On neighbors, derivatives, and viscosity in particle codes. In *Proceeding of ECCM Conference*. Munich, Germany, 31 August – 3 September 1999.
- [63] M. M. Rashid. Incremental kinematics for finite element applications. *International Journal for Numerical Methods in Engineering*, 36:3937–3956, 1993.
- [64] M. M. Rashid. Material State Remapping In Computational Solid Mechanics. *International Journal for Numerical Methods in Engineering*, 55:431–450, 2002.
- [65] M. M. Rashid and P. M. Gullett. On A Finite Element Method With Variable Element Topology. *Computational Plasticity Models, Software, and Applications*, 1:136–160, 1999.
- [66] M. M. Rashid and P. M. Gullett. Formulation of a finite element method with arbitrary element geometry. In *Modeling and Simulation-Based Life Cycle Engineering*, Chong K.P., Saigal S., Thynell S., Morgan H.S. (eds). Spon: London, 2002; 76-90.
- [67] M. M. Rashid and M. Selimotic. A Three - Dimensional Finite Element Method With Arbitrary Polyhedral Elements. *International Journal for Numerical Methods in Engineering*, 67:226–252, 2005.
- [68] Junuthula N. Reddy. *An Introduction to the Finite Element Method*. Series in Mechanical Engineering. McGraww-Hill, 2nd edition, 1993.
- [69] A. Yu Semenov, V.V. Belikov. New non-Sibson interpolation on arbitrary system of points in Euclidean space. In *15th IMACS World Congress*, vol. 2: Numerical Mathematics. Berlin, 1997; 237–242
- [70] Z.C. Shi. The F-E-M test for convergence of nonconforming finite element. *Mathematics of Computation*, 49:391–405, 1987.
- [71] J.C. Simo, F. Armero. Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering*, 33:1413–1449, 1992.
- [72] J.C. Simo, S. Rifai. A class of mixed assumed strain methods and the method of incompatible modes. *International Journal for Numerical Methods in Engineering*, 29:1595–1638, 1990.
- [73] F.L. Stazi, E. Budyn, J. Chessa, T. Belytschko. An extended finite element method with higher-order elements for curved cracks. *Computational Mechanics*, 31:38–48, 2003.
- [74] M. Stolarska, D.L. Chopp, N. Moes, T. Belytschko. Modelling crack growth by level sets in the extended finite element method. *International Journal for Numerical Methods in Engineering*, 51:943–960, 2001.

- [75] T. Strouboulis, K. Copps, I. Babuska. The generalized finite element method: an example of its implementation and illustration of its performance. *International Journal for Numerical Methods in Engineering*, 47(8):1401–1417, 2000.
- [76] T. Strouboulis, K. Copps, I. Babuska. The generalized finite element method. *Computer Methods in Applied Mechanics and Engineering*, 190:4081–4193, 2001.
- [77] F. Stummel. The generalized patch test. *SIAM Journal on Numerical Analysis*, 16:449–471, 1979.
- [78] N. Sukumar, N. Moes, B. Moran, T. Belytschko. Extended finite element method for three-dimensional crack modelling. *International Journal for Numerical Methods in Engineering*, 48 (11):1549–1570, 2000.
- [79] N. Sukumar, B. Moran, and T. Belytschko. The Natural Element Method in Solid Mechanics. *International Journal for Numerical Methods in Engineering*, 43:839–887, 1998.
- [80] N. Sukumar, B. Moran, A. Yu Semenov, and V. V. Belikov. Natural Neighbor Galerkin Methods. *International Journal for Numerical Methods in Engineering*, 50:1–27, 2001.
- [81] N. Sukumar, A. Tabarraei. Conforming polygonal finite elements. *International Journal for Numerical Methods in Engineering*, 61:2045–2066, 2004.
- [82] M. Suri. Analytic and computational assessment of locking in the hp finite element method. *Computer Methods in Applied Mechanics and Engineering*, 133:347–371, 1996.
- [83] S.P. Timoshenko, J.N. Goodier. *Theory of Elasticity*, (3rd edn). McGraw - Hill: New York, 1987.
- [84] G. Ventura, B. Moran, T. Belytschko. Dislocations by partition of unity. *International Journal for Numerical Methods in Engineering*, 62:1463–1487, 2005.
- [85] G.J. Wagner, N. Moes, W.K. Liu, T. Belytschko. The extended finite element method for rigid particles in Stokes flow. *International Journal for Numerical Methods in Engineering*, 51:293–313, 2001.
- [86] J.W. Yoo, B. Moran, J.S. Chen. Stabilized conforming nodal integration in the natural-element method. *International Journal for Numerical Methods in Engineering*, 60:861–890, 2004.
- [87] X. Zhang, X. Liu, M.-W. Lu, Y. Chen. Imposition of essential boundary conditions by displacement constraint equations in meshless methods. *Communications in Numerical Methods in Engineering*, 17:165–178, 2001.

- [88] G. Zi, T. Belytschko. New Crack-Tip Elements for XFEM and Applications to Cohesive Cracks. *International Journal for Numerical Methods in Engineering*, 57:2221–2240, 2003.
- [89] G.S. Zi, H. Chen, J.X. Xu, T. Belytschko. The extended finite element method for dynamic fractures. *Shock and Vibration*, 12(1):9–23, 2005.
- [90] Olgierd Cecil Zienkiewicz and Robert L. Taylor. *The Finite Element Method*, volume 1. McGraw - Hill Book Company, fourth edition, 1991.
- [91] Olgierd Cecil Zienkiewicz and Robert L. Taylor. *The Finite Element Method*, volume 2. McGraw - Hill Book Company, Fourth edition, 1991.