2009

Discrete gradient method in solid mechanics

Jing Qian
University of Iowa

DISCRETE GRADIENT METHOD IN SOLID MECHANICS

by

Jing Qian

An Abstract

Of a thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

May 2009

Thesis Supervisor: Jia Lu

ABSTRACT

The discrete gradient method is proposed as a novel numerical tool to perform solid mechanics analysis directly on point-cloud models without converting the models into a finite element mesh. This method does not introduce continuous approximation of the primary unknown field variables; instead, it computes the gradients of the field variables at a node using discrete differentials involving a set of neighboring nodes. The discrete gradients are substituted into Galerkin weak from to derive the algebraic governing equations for further analysis. Therefore, the formulation renders a completely discrete computation that can conduct mechanical analysis on point-cloud representations of patient-specific organs without resorting to finite element method.

Since the method is prone to rank-deficient instability, a stabilized scheme is developed by employing penalty that involves a minor modification to the method. The difference between nodal strain and subcell strain is penalized to prevent the appearance of zero average strain.

This dissertation delineates the theoretical underpins of the method and provides a detailed description of its implementation in two and three-dimensional elasticity problem. Several benchmark numerical tests are presented to demonstrate the accuracy, convergence, and capability of dealing with compressibility and incompressibility constraint without severe locking. An efficient method is also developed to automatically extract point-cloud models from medical images. Two and three-dimensional examples of biomedical applications are presented too.

Abstract Approved: _____

Thesis Supervisor

_____

Title and Department

_____

Date

DISCRETE GRADIENT METHOD IN SOLID MECHANICS

by

Jing Qian

A thesis submitted in partial fulfillment of the
requirements for the Doctor of Philosophy
degree in Mechanical Engineering
in the Graduate College of
The University of Iowa

May 2009

Thesis Supervisor: Jia Lu

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Jing Qian

has been approved by the Examining Committee for the thesis requirement for the Doctor of Philosophy degree in Mechanical Engineering at the May 2009 graduation.

Thesis Committee: _____
Jia Lu, Thesis Supervisor

_____
Weimin Han

_____
Sharif Rahman

_____
Colby C. Swan

_____
Shaoping Xiao

To my family

ACKNOWLEDGEMENTS

I would like to express my deep and sincere gratitude to my advisor, Professor Jia Lu. His inspirational suggestions and careful guidance have been of great value for me in graduate studies.

Many special thanks go to my wife Tingting Que, who shared happiness and difficult times with me. She has been a great source of strength for me all through this work.

A cordial appreciation goes to my best fiends Dr. Xuefeng Zhao and Dr. Yujiang Xiang. Their sincere friendships inspirit me to accomplish the work in this dissertation.

Finally, I am forever indebted to my parents and brother for their understanding, encouragement and the values they installed in me.

I dedicate this work to all of them.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1
INTRODUCTION

1.1 Motivation

Medical imaging technologies such as computed tomography (CT), magnetic resonance imaging (MRI) and ultrasonic imaging (UI) can provide anatomically accurate geometric information of internal organs. Pixel or voxel data from the medical images provide a point-cloud depiction for complicated anatomies that are difficult to describe in CAD geometry. Traditionally, a point-cloud model needs to be converted into finite element mesh to perform mechanical analysis. However, generating high quality meshes in complicated bodies remains a challenge although meshing generation tools have been significantly improved over the last decades. Therefore, the application of finite element method (FEM) has been limited in this filed.

Another technique, called voxel finite element method (VFEM), directly converts image voxels to eight node hexahedral elements [49, 54, 58, 101]. However, the jagged surfaces lead to inaccurate strain and stress at nodes corresponding to corners. Moreover, the un-smoothed geometry with sharp geometrical discontinuities might cause numerical problems such as convergence and accuracy [71]. To overcome this shortcoming, some techniques of smoothing surface were adopted [16, 22]. In addition, algorithms were developed to march cubes with tetrahedrons for smooth surface [48, 68, 78]. However, with such extra process, it is not possible directly to transfer CT voxels to hexahedral elements. On the other hand, the tetrahedral element may be of disadvantage in terms of accuracy.

To circumvent this restriction, several methodologies have been intensively pursued to perform image-based analysis directly on domains represented by point-clouds as happen in medical images without converting the model into a finite element mesh [30, 31, 35].

The first family of methods, which have gained great popularity, are meshfree method, e.g., [2, 10, 37, 67, 72]. The meshfree methods construct smooth approximations over a set of scattered nodes that have no particular topological connections or element structure required to construct approximations. Therefore, meshfree methods present significant advances over the finite element method in the issue of meshing generation and relating applications such as moving discontinuities and large deformation. However, except for a few formulations, most meshfree methods still face algorithmic difficulties related to the loss of interpolatory property in the approximation, or the requirement of higher-order quadrature rules in numerical integration. Some formulations such as smoothed particle hydrodynamics method [12, 50, 69, 74–77] does not satisfy the linear consistency requirement. Moreover, most meshfree Galerkin methods require background cell for integrating the weak form. The accuracy as well as the stability will be influenced by the selection of Gauss quadrature points [9, 38, 95, 96]. On the other hand, most meshfree methods involve the construction of implicit functions so that the numerical computation is much more expensive than explicit interpolation based finite element method.

Another family of methods that can render a fully meshfree computation are discrete difference methods including the finite difference methods [39, 63, 81, 86, 98]

and some finite volume methods [4, 42, 44, 55, 79], which have been used in numerical solution of differential equations. They are *fully discrete* in the sense that both the domain and the unknown are represented by discrete quantities. These methods do not construct continuous approximation of the unknown field. Instead, the unknown variables and its derivatives are discretized into either nodes or volumes or the so-called storage points.

Finite difference methods (FDM) or finite volume methods (FVM) have their own limitations when they are applied with point-cloud models. First, it is hard to solve high order partial differential equation using FDM because of the difficulty to approximate high order differential operators. Second, classic FDM or FVM works well just for structured mesh although there are some developments to extend such methods to unstructured grids [19–21, 56, 57, 62–64, 70, 83]. Algorithms working on irregular grids need special treatment to define the stencils and approximate differentials. Most existing works in this area were devoted to the selection of computation cell and difference schemes because computational cell must be carefully chosen to avoid singularity and losing of accuracy. In addition, finite difference methods were mostly carried out for strong form [81], the weak form formulations have not been adequately investigated [19, 42, 44, 82]. To develop a rigorous weak form discrete method, issues such as discretization, consistency, stability and patch tests need to be carefully addressed.

The motivation for my work in this dissertation arises partially from the emerging field of image-based, patient-specific biomechanical analysis. A discrete solver

that interfaces seamlessly with discrete point-cloud models would be more desirable. Therefore, the aims of this dissertation is to

1. develop a discrete solver in the framework of weak form;
2. provide a discrete method relying on automatic domain tessellation;
3. derive approximation of gradient over arbitrarily scatted nodes;
4. demonstrate the accuracy and efficiency of the new method;
5. implement the new methods with point-cloud models derived from medical images.

1.2 Outline of this thesis

The dissertation is organized as follow.

In Chapter 2, several tessellation approaches, which are employed in this dissertation, are introduced. Some existing discrete differential methods, newly proposed natural element method and nodal integration method are reviewed.

In Chapter 3, the fundamental concept of discrete gradient is introduced. The framework of the discrete Galerkin formulation throughout this dissertation is introduced. The linear consistency condition for approximation is deduced. The issue of patch test, which has been overlooked in previous contributions, is addressed.

In Chapter 4, Voronoi cell discrete gradient method is presented. The implementation of this method in linear elasticity, followed by the extension to geometrically nonlinear problems, is described. A proof that the method satisfies the linear

displacement patch test is provided .

Chapter 5 introduces a bacrycentric discrete gradient method. It begins with the introduction of barycentric division over domain, followed by a discussion on its properties. Proofs of linear consistency and satisfaction of patch test are presented.

In Chapter 6, we present the numerical results of some benchmark tests to validate the accuracy and convergence of newly proposed method in elasticity and finite strain formulations.

In Chapter 7, we present a stabilized discrete method over polygon mesh. The subcell strain and a modified from of strain energy are proposed to prevent spurious modes. The stabilization formulation under the framework of Galerkin weak form is deduced. A one-dimensional example is discussed to exhibit the existence of spurious mode. Several benchmark numerical tests are presented to demonstrate the properties of the method.

In chapter 8, the discrete stress analysis on point-cloud models in biomechanics is introduced. To fulfill the discrete analysis, we develop an efficient method to extract point-cloud model from medical images. Details of implementation are demonstrated in two and three dimensional examples.

In chapter 9, conclusion remarks are mode to summary the work in this dissertation.

CHAPTER 2
AN OVERVIEW ON DERIVATIVE APPROXIMATION METHODS

This chapter, we review some existing methods that approximate derivative over scatted nodes. Although different discrete approximations employ various algorithms, they share a fundamental feature, which the computational domain is first tessellated into a collection of cells that completely cover the domain and the created cells can define relationships among nodes. Therefore, in this chapter, we start with the introduction of tessellation of domain. Then, several differential approximation methods are shortly introduced.

2.1 Tessellations of a domain

In the finite element method, the elements provide a way of dividing the domain. Approximation techniques use explicit shape functions to rebuild field variables in each element formed by fixed nodes. This type of interpolation is termed stationary element based interpolation. However, when the spatial discretiztion lacks an element framework, the field variables and their derivatives at an arbitrary node have to be approximated by using a group of field nodes in a local domain. Hereby, those nodes included in the interpolation are considered as supporting nodes. In addition, the small local domain containing those supporting nodes is influence domain. The accuracy of interpolation upon the node of interest depends on all supporting nodes. Therefore, a suitable supporting domain should be chosen to ensure an efficient and accurate approximation. Evidently, different tessellations method of a domain may lead to various interpolating process. In our work, computational domain is required

to be automatically tessellated into a collection of non-overlapping cells that completely cover the domain. Mathematically, a domain Ω that originally contains N nodes is partitioned into N cells. If the cell associated with the node I is denote by Ω_I, then, obviously, $\Omega_I \cap \Omega_J = \emptyset$ if $I \neq J$, and $\cup_{I=1}^{N} \Omega_I = \Omega$. For subsequent use, we also introduce the following notations:

$nodes(\Omega_I)$: The nodal set of the cell I including node I;

N_I : $nodes(\Omega_I) \backslash I$, all supporting nodes of node I;

A_I : The area of cell Ω_I.

In future chapters, we mainly consider four kinds of tessellation: Voronoi diagram, Delaunay triangulation, quadrilateral and random polygons.

Voronoi diagram is a geometric construction that decomposes a domain in Euclidean space into non-overlapping cells according to distances among a given set of scattered nodes. For simplicity of exposition, we describe the tessellation in two-dimensional space. Given a set of nodes $\{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$ in a plane, the Voronoi cell $\mathcal{V}(I)$ associated with node $\mathbf{X}_I$ is defined as

$$\mathcal{V}(I) = \{\mathbf{X} \in \mathbb{R}^2 : \ d(\mathbf{X}, \mathbf{X}_I) < d(\mathbf{X}, \mathbf{X}_J) \ \forall J \neq I\}, \tag{2.1}$$

where d is the Euclidean distance. Each cell is obviously the convex polygon formed by the intersection of finitely many open half spaces, each being delimited by the perpendicular bisecting plane of the line segments connecting $\mathbf{X}_I$ and other nodes.

Delaunay triangulation is dual of Voronoi diagram. In computational geometry, a Delaunay triangulation for a set N of points in the plane is a triangulation $\mathcal{DT}(N)$ such that no point in N is inside the circumcircle of any triangle in $\mathcal{DT}(N)$. Boris Delaunay invented the triangulation in 1934. This is the original definition for two-dimensional spaces. However, it is possible to use it in three-dimensional spaces by using a circumscribed sphere in place of the circumcircle. Delaunay triangulations are often used to build meshes for the finite element method, because of the Delaunay triangulation maximize the minimum angle of all the angles of the triangles in the triangulation.

Quadrilateral and polygons tessellation can be considered as the expansion of triangular partition because they all can be divided into triangles.

2.2 Finite difference method and finite volume method

The finite difference methods [39, 63, 81] approximate the derivatives using algebraic expressions relating nodal values without providing information about the distribution of unknown variable among nodes. Most finite difference formulations follow a collocation procedure where the derivatives in the strong form are replaced by finite difference approximations [81]. An alternative approach, the idea of using discrete differential operators in a weak form or energy method, has been proposed in the early studies on computational structural mechanics. Bushell [19], Johnson [57], Jensen [56] and Pavlin [82] introduced weak form based finite difference implementations for plate and shell problems. Recently, Hao *et al* [53] proposed a moving particle finite element method, in which the gradient operator is directly approximated, and

is substituted to a weak form. Eymard *et al* [42,44] investigated a discrete weak-form formulation for the Laplace equation in the context of the finite volume methods.

In the 1960s, Tichonon and Samarskii *et al* [86, 98] introduced the concept of finite volume methods. "Finite volume" refers to the small volume surrounding each node point. The finite volume methods employ a volume average of the strong form to derive the discrete equations [4,42–44,55,79]. These methods are proved to be well suited to deal with physical conservation laws. Being free from an underlying continuous approximation, these methods do not need meshing, and are in general more efficient numerically. Some of the notable features of the finite volume methods are that it is applicable to arbitrary geometries and it may use structured or unstructured meshes [20, 21]. In addition, this method leads to robust schemes.

2.3 Discrete differential operators on non-uniform grid

The numerical computation of difference approximations on non-uniform grids is used for the solution of partial differential equations, or in data fitting. Although classic finite difference schemes are typically based on regular grids, techniques have been proposed to derive finite difference operators over arbitrarily scattered nodes since 1970s [19,56,57,62–64,70,83]. Jensen [56] and Liszka *et at* [62–64] took another path and proposed a generalized finite difference method (GFDM) on irregular grids through appropriate computational cell (star-shaped domain).

A number of applications in various second order boundary value problems had been discussed, and primary contributions focused on finite difference algorithms over irregular grids [83]. Jensen *et al* [56] used the Taylor series expansions to obtain finite

difference formulas approximating derivatives up to the second order. Baty and Villon *et al* [5] used least squares method, whereas Breitkopf *et al* [17, 18] adopted moving least squares approximation to solve elliptic problems on arbitrary irregular grids. Onate *et at* [39, 80, 81] presented a particle method called free finite point method (FPM) to solve convection-diffusion and fluid flow type problems. The approach was based on a weighted least square interpolation of point data and point collocation for evaluation the approximation integrals. Recently, Tonti [100], Cosmi [28] and Ferretti [45] developed discrete formulations in solid mechanics. Their works indicated that discrete formulations can anatomically preserve some key properties of the physical problem.

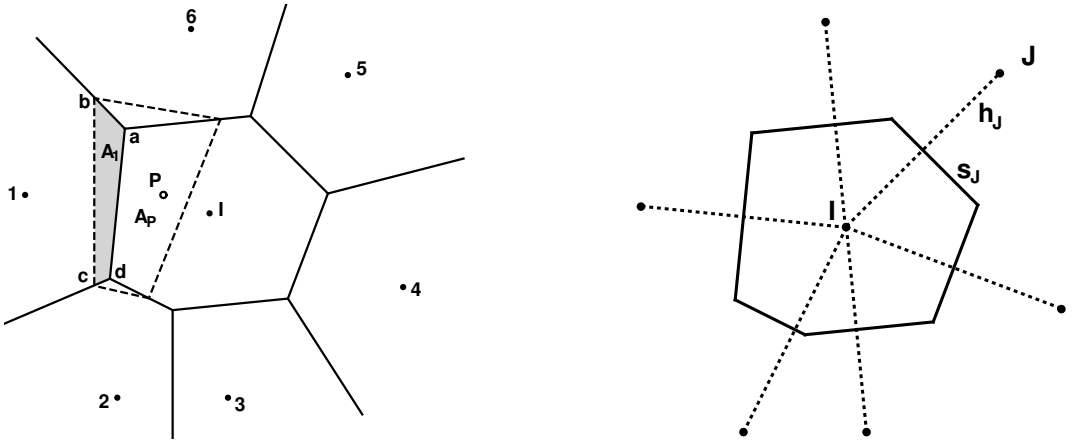
2.4 Natural neighbor based interpolation

The notions of natural neighbors and natural neighbor interpolation were introduced by Sibson *et at* [87] as a means for data fitting and smoothing. Consider a set of scattered nodes N and its Voronoi diagram $\mathcal{V}(N)$. If the Voronoi cell for node I and P has a common facet, then the node I is said to be a natural neighbor of node P.

The natural neighbor co-ordinates are used as the interpolating functions in natural neighbor interpolation or Sibson interpolation. As shown in Figure 2.1(a), the natural neighbor coordinates of P with respect to a neighbor I is defined by the ratio of area measure in $\mathbb{R}^2$:

$$\phi_I(\mathbf{x}) = \frac{A_I(\mathbf{x})}{A(\mathbf{x})}, \quad A(\mathbf{x}) = \sum_J^{N_I} A_J(\mathbf{x}),$$

where $A_I(x)$ is the area of overlap between the first-order Voronoi cells of node P and I, $A(\mathbf{x})$ is the area of the first-order Voronoi cell of P. Natural neighbor interpolation has primarily been used in the area of data interpolation and modelling of geophysical phenomena.



(a) (b)

Figure 2.1: Natural neighbor-based interpolants. (a) Sibson interpolant; (b) Laplace interpolant.

Belikov *et at* [7] proposed a new natural neighbor based interpolant referred as non-Sibsonian interpolant. Let $\mathcal{V}_I$ be the Voronoi cell associated with node I, t_{IJ} be the common facet that is shared by $\mathcal{V}_I$ and $\mathcal{V}_J$ and $|t_{IJ}|$ denotes the Lebesgue measure of t_{IJ}. If node I has n natural neighbor, then its non-Sibsonian shape function is defined as

$$\phi_I(\mathbf{x}) = \frac{\alpha_I(\mathbf{x})}{\sum_J^n \alpha_J(\mathbf{x})}, \quad \alpha_J(\mathbf{x}) = \frac{|t_J(\mathbf{x})|}{h_J(\mathbf{x})}, \tag{2.2}$$

where $h_J(\mathbf{x})$ is the Euclidean distance between node I and its natural neighbor as plotted in Figure 2.1(b). Therefore, $\alpha_I(\mathbf{x})$ is the weight function.

For Sibson and non-Sibsonian shape functions, they have several nice properties as following:

1. positivity $0 \leq \phi_I(\mathbf{x}) \leq 1$;
2. Kronecker-Delta property $\phi_I(\mathbf{x}_J) = \delta_{IJ}$;
3. partition of unity $\sum_I^n \phi_I(\mathbf{x}) = 1$;
4. linear consistency $\mathbf{x} = \sum_I^n \phi_I(\mathbf{x})\mathbf{x}_I$.

The natural element method (NEM) is a Galerkin method for the solution of partial differential equations developed by Sukumar [30, 31, 90, 92, 93]. The approximation of this method is based on Sibson or Non-Sibson interpolation [7, 8, 87] over natural neighbor co-ordinates.

Sukumar *et al* [90], Eymard *et al* [44] proposed a Voronoi cell finite difference method to discrete differential operators, in which Sukumar adopted the Voronoi cell and natural neighbors to study difference approximations for the diffusion operators.

Consider Laplace equation

$$\begin{aligned} &\mathcal{L}u(\mathbf{x}) = \nabla \cdot (\nabla u(\mathbf{x})) = f(\mathbf{x}) \quad \texttt{in } \Omega \\ &u(\mathbf{x}) = g(\mathbf{x}) \quad \texttt{on } \partial\Omega, \end{aligned} \tag{2.3}$$

where ∇ is the gradient operator, $\partial\Omega$ is the boundary of Ω. The discrete form of

above equation is

$$\begin{aligned} &\mathcal{L}_h u(\mathbf{x}_I) = \nabla \cdot (\nabla u(\mathbf{x}_I)) = f(\mathbf{x}_I) \quad \mathbf{x}_I \in \Omega \\ &u(\mathbf{x}_I) = g(\mathbf{x}_I) \quad \mathbf{x}_I \in \partial\Omega, \end{aligned} \tag{2.4}$$

Now we can proceed to find the discrete approximation for the Laplace operator at node I. The starting point is the balance law and Gauss's theorem.

$$(\mathcal{L}_h u)_I = \lim_{A_I \to 0} \frac{\int_{\partial A_I} \nabla u \cdot \mathbf{n} d\Gamma}{\int_{A_I} d\Omega} = \lim_{A_I \to 0} \frac{\int_{\partial A_I} \frac{\partial u}{\partial n} d\Gamma}{A_I}, \tag{2.5}$$

where A_I is the area of the Voronoi cell of node I.

On using a simple central difference approximation of the derivative of u normal to the Voronoi edge, the above equation can be written as:

$$(\mathcal{L}_h u)_I = \lim_{A_I \to 0} \frac{\int_{\partial A_I} \frac{\partial u}{\partial n} d\Gamma}{A_I} \approx \frac{1}{A_I} \sum_{J=1}^{m} \frac{u_J - u_I}{h_{IJ}} s_{IJ}, \tag{2.6}$$

where m is the number of natural neighbors for node I, h_{IJ} is the distance between node I and J, s_{IJ} is the length of the Voronoi edge associated with node I and J. Using Equation (2.6) and after some algebraic simplification, the approximation for Laplace operator is

$$\begin{aligned} &(\mathcal{L}_h u)_I = \frac{1}{A_I}[(\sum_{J=1}^{m} \alpha_{IJ} u_{IJ}) - \alpha_I u_I], \\ &\alpha_{IJ} = \frac{s_{IJ}}{h_{IJ}}, \quad \alpha_I = \sum_{J=1}^{m} \alpha_{IJ}, \quad A_I = \frac{1}{4} \sum_{J=1}^{m} s_{IJ} h_{IJ}. \end{aligned} \tag{2.7}$$

The above expression is consistent with the prescription introduced for the discrete Laplace on a random lattice.

As Sukumar mentioned in [91], the advantages of this approach are two-fold. First, natural neighbor provides a simple means to determine uniquely the neighbors for point. The Voronoi cells also provide a weight function for each neighbor. Second, natural neighbor-based interpolation is a local approximation that is well defined at all points in the domain. The computational costs involved in the construction of the interpolant are also minimal since only algebraic calculations are required in the evaluation of the discrete differential operators.

Sukumar *et al* [94] extended the natural neighbor method to convex polygons and named it as conforming polygonal finite elements. By referring the concepts of generalized barycentric coordinates [73] and mean value coordinate [46, 47], Sukumar proposed a conforming approximation using natural neighbor interpolant and extended the potential application of finite element to convex polygons of arbitrary order.

2.5 Nodal integration methods

Nodal integration methods were initially motivated by the need of eliminating the integration mesh in meshfree methods [6, 14, 25, 26, 51, 102, 103]. Some finite element methods, such as node-based average strain finite element methods [13,15,36] and smoothed finite element method [65,66], share the similar idea. Strictly speaking, nodal integration only evaluates strain or stress at the nodes and is a form of particle method. These particle methods offer a Lagrangian solution to very large deformation

problems.

Chen *et al.* [25, 26] introduced a nodal average strain computed from the numerical average of the continuum strain over Voronoi cell inclosing the node as shown in Figure 2.2(a). The nodal area A_I is computed over nodal domain Ω_I defined by Voronoi diagram, the nodal strain is computed as

$$(\boldsymbol{\varepsilon}^h)_I = \frac{1}{A_I} \int_{\partial\Omega_I} \frac{1}{2}(\mathbf{u} \otimes \mathbf{n} + \mathbf{n} \otimes \mathbf{u}) d\Gamma, \tag{2.8}$$

where $\mathbf{n}$ is the cell wall normal. The strain smoothing avoids evaluating derivatives of meshfree shape functions at nodes and thus eliminates spurious modes. The procedure was later extended to natural neighbor finite elements [103].

The average strain finite element methods compute average volumetric strains or strains at nodes based on surrounding element. It was motivated by the desire to improve the performance of lower order elements in the incompressibility limit. This method was believed that the nodal integration, in which the number of integration points exactly matches the number of nodes, offers an optimal equations-to-constraints ratio [61]. Existing studies focused mainly on linear triangle element (2D) and linear tetrahedron element (3D). In these elements, the strains are constant, and the nodal strain can simply be taken as a weighted average of the element strains. To use linear tetrahedral elements even in large strain incompressible situations, Bonet *et al.* [13] presented a simple technique that is based on the definition of nodal volumes leading to a nodal pressure that is averaged over the elements. Referring Figure

2.2(b), the set $elem(I)$ is defined to be the group of tetrahedral element common to nodal I. Therefore, the nodal cell volume attached to node I can be defined as

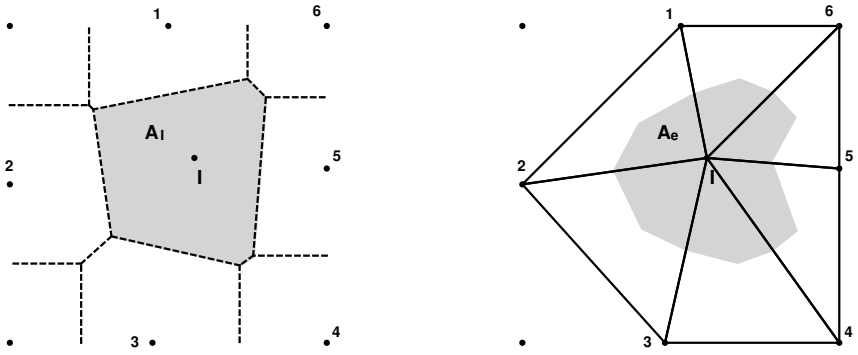
$$V_I = \sum_{e \in elem(I)} V_e,$$

where V_e is one fourth of the area of tetrahedral element e. Dohrmann [36] developed this idea and formulated the average small strain finite element methods by processing nodal average to the whole strain tensor rather than just the volumetric component. The displacement gradient of element e is $(\nabla^h \mathbf{u})_e$, and the strain is $(\boldsymbol{\varepsilon}^h)_e = (\nabla_s^h \mathbf{u})_e$, where ∇_s is the symmetric part. Therefore, the average nodal gradient $(\nabla^h \mathbf{u})_I$ and strain $(\boldsymbol{\varepsilon}^h)_I$ are defined as

$$(\nabla^h \mathbf{u})_I = \frac{1}{V_I} \sum_{e \in elem(I)} V_e \nabla^h \mathbf{u}_e \quad \texttt{or} \quad (\boldsymbol{\varepsilon}^h)_I = \frac{1}{V_I} \sum_{e \in elem(I)} V_e \boldsymbol{\varepsilon}_e^h. \tag{2.9}$$

The nodal averaging method requires the strain $\boldsymbol{\varepsilon}_e^h$ to be computed within each triangle element e first. Furthermore, Bonet [15] extend the formulation proposed by Dohrmann to the large strain regime by using the deformation gradient tensor as the main kinematic variable.

Recently, Krysl and Zhu [61] developed average strain formulation for a quadrilateral mesh, in which the nodal strain was derived from a variational setting. Puso and Solberg [85] studied the stability of nodal average strain linear tetrahedron, and proposed a stabilized formulation. Conceivably, the same way may also be applied to polygon elements [94].



(a) (b)

Figure 2.2: Voronoi cell and subcells.

The aspects of stability, consistency and explicit time integration in both element-based and meshfree-based nodal integration methods are discussed in a recent publication [84].

CHAPTER 3
DISCRETE GRADIENT GALERKIN METHOD

3.1 Discrete gradient and gradient interpolants

The idea to discrete gradient over cells usually is employed by finite difference methods or finite volume methods. Taylor expansion, least squares method and Green-Gauss method [33, 34] are candidates of such kind of methods. After partition of domain, approximations can be defined over cells. For example, the Green-Gauss method linearly reconstructs gradient as

$$\int_{\Omega} \nabla u da = \sum_{J \in N_I} \frac{3}{2}(u_I + u_J) \upsilon_{IJ}, \tag{3.1}$$

where $\upsilon_{IJ} = \int_{\mathbf{x}_{J-\frac{1}{2}}}^{\mathbf{x}_{J+\frac{1}{2}}} \mathbf{n} ds$ represents integrated normal along cell's edges, see Figure 3.1

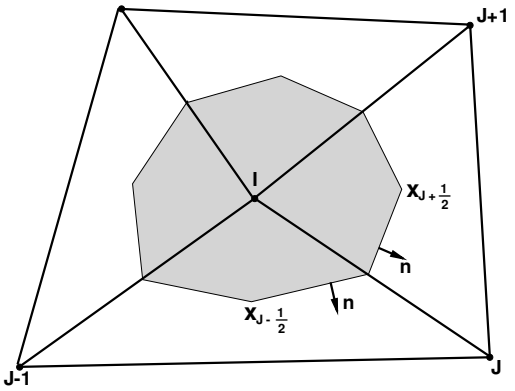


Figure 3.1: Nodal cell demarcated by median segments of triangles incident to the vertex I with cyclically indexed neighbors

Follow the same path, we will develop algorithms for computing average gradient over a finite cell by providing an interpolating formula of the form

$$(\nabla^h u)_I = \sum_{J \in nodes(\Omega_I)} \mathbf{R}^*_{IJ} u_J, \tag{3.2}$$

where $\mathbf{R}^*_{IJ}$ is the gradient interpolant associated with node parameters u_J. This derivation replies on geometric constructions to identify the computational cells firstly. It is evident that different cell definitions lead to various computation of gradient interpolants. Detailed schemes will be discussed in Chapter 4 and 5. Its computational structure is similar to node-based average strain finite element methods [13, 15, 36], nodal integration methods [6,25,26,103], and smoothed finite element method [65,66].

3.2 Linear consistency condition for gradient interpolants

For second order physical problems such as elasticity and heat conduction, the basic requirement on the discrete operator (3.2) is the linear consistency. It means that the discrete gradient can exactly reproduce the gradient of an arbitrary linear function. This linear consistency condition, sometimes, is called first-order consistency in $\mathbb{R}^3$ space. Consider the function $u^{lin} = a_0 + a_1 X + a_2 Y$, where a_0, a_1, a_2 are arbitrary constants. In vector form we write $u^{lin} = a_0 + \mathbf{a} \cdot \mathbf{X}$ where $\mathbf{a} = a_1 \mathbf{e}_x + a_2 \mathbf{e}_y$ and $\mathbf{X} = X\mathbf{e}_x + Y\mathbf{e}_y$. The linear consistency condition requires that $(\nabla^h u)_I = \nabla u^{lin} = \mathbf{a}$ when the nodal values are assigned according to $u_J = a_0 + \mathbf{a} \cdot \mathbf{X}_J$. Substituting the

nodal values into (3.2), we see that the linear consistency infers

$$(\nabla^h u^{lin})_I = \Big[\sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ}\Big] a_0 + \Big[\sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ}\otimes \mathbf{X}_J\Big]\mathbf{a} = \mathbf{a} \quad \forall a_0 \in \mathbb{R},\ \mathbf{a}\in\mathbb{R}^2. \tag{3.3}$$

Evidently, the condition is satisfied if and only if

$$\begin{aligned} \sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ} &= \mathbf{0}, \\ \sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ}\otimes \mathbf{X}_J &= \mathbf{I}. \end{aligned} \tag{3.4}$$

These are the discrete form of the *derivative consistency conditions* recorded in [59,60] and [53]. Here, "$\otimes$" denotes the standard tensor product and $\mathbf{I}$ is the second order identity tensor.

Remark 1. *Note that we can deduce* $R^*_{II} = -\sum_{J\in N_I} R^*_{IJ}$ *from* $(3.4)_1$ *and eventually write* $(3.4)_2$ *as*

$$\sum_{J\in N_I} \mathbf{R}^*_{IJ}\otimes \mathbf{R}_{IJ} = \mathbf{I},$$

where $\mathbf{R}_{IJ} = \mathbf{X}_J - \mathbf{X}_I$. *If we think* $\mathbf{R}_{IJ}$ *as the local "basis" incident on the node* I, *this relation suggests that* $\mathbf{R}^*_{IJ}$ *constitute a set of generalized dual basis. The notation of* $\mathbf{R}^*_{IJ}$ *is motivated by this observation.*

3.3 Discrete Galerkin formulation for linear elasticity

Consider the elasticity problems governed by the strong form:

$$\begin{aligned} &\nabla\boldsymbol{\sigma} + \rho\mathbf{b} = \rho\ddot{\mathbf{u}}, \quad \boldsymbol{\sigma} = \boldsymbol{\sigma}^T \quad \text{in } \Omega, \\ &\boldsymbol{\sigma} = \mathbb{D} : \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} = \frac{1}{2}(\nabla\mathbf{u} + \nabla^T\mathbf{u}) := \nabla_s\mathbf{u}, \end{aligned} \tag{3.5}$$

where $\boldsymbol{\sigma}$ is the Cauchy stress, $\boldsymbol{\varepsilon}$ is the strain tensor, ρ is the density, $\ddot{\mathbf{u}}$ is the acceleration, and $\mathbb{D}$ is the fourth-order elasticity tensor, assumed to be bounded, elliptic and symmetric. The subscript s indicates the symmetric part of the gradient. The boundary is a disjoint union of $\partial\Omega_{\mathbf{t}}$ and $\partial\Omega_{\mathbf{u}}$, over which the following boundary conditions are applied:

$$\begin{aligned} &\boldsymbol{\sigma}\mathbf{n} = \bar{\mathbf{t}} \quad \text{on } \partial\Omega_{\mathbf{t}} \\ &\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \partial\Omega_{\mathbf{u}}. \end{aligned} \tag{3.6}$$

Here $\mathbf{n}$ is the unit outwards normal vector on the boundary. Conventional Galerkin formulations are based on the continuum weak form

$$\int_\Omega \boldsymbol{\varepsilon}(\delta\mathbf{u}) : \mathbb{D}\boldsymbol{\varepsilon}(\mathbf{u})da + \int_\Omega \delta\mathbf{u}\cdot\rho\ddot{\mathbf{u}}da - \int_\Omega \delta\mathbf{u}\cdot\rho\mathbf{b}da - \int_{\partial\Omega_{\mathbf{t}}} \delta\mathbf{u}\cdot\bar{\mathbf{t}}ds = 0, \tag{3.7}$$

where $\delta\mathbf{u}$ is any kinematically admissible virtual displacement. Here, $(\cdot)$ and $(:)$ denote the vector inner product and the tensor contraction, respectively.

In the discrete format, the integration over the domain breaks into a sum over nodal cells. Over each cell, the strain is assumed constant and is computed

from the discrete gradient at the corresponding nodes. A nodal integration scheme is introduced, so that

$$\begin{aligned}
&\int_\Omega \boldsymbol{\varepsilon}(\delta\mathbf{u}) : \mathbb{D}\boldsymbol{\varepsilon}(\mathbf{u}) da \approx \sum_I A_I (\nabla_s^h \delta\mathbf{u})_I : \mathbb{D}\boldsymbol{\varepsilon}_I(\mathbf{u}), \\
&\int_\Omega \delta\mathbf{u}\cdot\rho\ddot{\mathbf{u}} da \approx \sum_I A_I \delta\mathbf{u}\cdot(\rho\ddot{\mathbf{u}})_I = \sum_I m_I \delta\mathbf{u}\cdot\ddot{\mathbf{u}}_I, \\
&\int_\Omega \delta\mathbf{u}\cdot\rho\mathbf{b}\, da + \int_{\partial\Omega_\mathbf{t}} \delta\mathbf{u}\cdot\bar{\mathbf{t}}\, ds \approx \sum_I \delta\mathbf{u}_I\cdot\mathbf{f}_I^{ext},
\end{aligned} \tag{3.8}$$

where $\boldsymbol{\varepsilon}_I$ is average strain for nodal cell, $m_I = A_I\rho$ and $\mathbf{f}_I^{ext} := \int_{\Omega_I}\rho\mathbf{b}\, da + \int_{\partial\Omega_\mathbf{t}\cap\partial\Omega_I}\bar{\mathbf{t}}$. This procedure may be regarded as a nodal integration scheme where A_I play the role of integration weights. The choice of weight is crucial for passing the patch test.

Over each nodal cell, the strain is approximated by the node-wise discrete gradient. The discrete gradient of the displacement $\mathbf{u}$, now a vector quantity and using the discrete kinematics, is:

$$\boldsymbol{\varepsilon}_I^h = (\nabla_s^h\mathbf{u})_I = \frac{1}{2}\sum_{J\in nodes(\Omega_I)} (\mathbf{u}_J\otimes\mathbf{R}_{IJ}^* + \mathbf{R}_{IJ}^*\otimes\mathbf{u}_J). \tag{3.9}$$

We write the strain vector

$$[\boldsymbol{\varepsilon}_I^h] = \sum_{J\in nodes(\Omega_I)} \mathbf{B}_{IJ}\mathbf{u}_J, \quad \text{where} \quad \mathbf{B}_{IJ} = \begin{bmatrix} R_{IJ}^{*1} & 0 \\ 0 & R_{IJ}^{*2} \\ R_{IJ}^{*2} & R_{IJ}^{*1} \end{bmatrix}. \tag{3.10}$$

Therefore, the cell-level stiffness matrix at the nodal cell I is

$$\mathbf{K}_I = A_I \mathbf{B}_{IK}^T \mathbf{D} \mathbf{B}_{IL} \quad K, L \in nodes(\Omega_I). \tag{3.11}$$

The gradient of the virtual displacement is computed similarly:

$$(\nabla^h \delta \mathbf{u})_I = \sum_{J \in nodes(\Omega_I)} \delta \mathbf{u}_J \otimes \mathbf{R}_{IJ}^*. \tag{3.12}$$

Inserting (3.9) and (3.12) into the right side of $(3.8)_1$ and re-arranging the terms, we can write

$$\sum_I A_I (\nabla_s^h \delta \mathbf{u})_I : \boldsymbol{\sigma}(\boldsymbol{\varepsilon}_I^h) = \sum_I \delta \mathbf{u}_I \cdot \mathbf{f}_I^{int},$$

A straightforward manipulation shows that

$$\mathbf{f}_I^{int} = \sum_{J \in nodes(\Omega_I)} A_I \boldsymbol{\sigma}(\boldsymbol{\varepsilon}_I^h) \mathbf{R}_{JI}^*, \tag{3.13}$$

where $\mathbf{R}_{JI}^*$ is the node-I interplant associated with node J. Therefore, the discrete system of equations can be expressed as

$$\mathbf{f}_I^{int} + m_I \ddot{\mathbf{u}}_I = \mathbf{f}_I^{ext} \qquad \forall I. \tag{3.14}$$

Eventually, the global equation is

$$\mathbf{K}\mathbf{d} + \mathbf{M}\ddot{\mathbf{d}} = \mathbf{F}^{ext},$$

where $\mathbf{K}$ is the global stiffness matrix assembled from cellular stiffness matrices, and $\mathbf{M}$ is the global diagonal mass matrix.

3.4 Discrete Galerkin formulation for finite strain

3.4.1 Fundamentals of continuum mechanics

In this section, we introduce the basic concepts of continuum mechanics and nonlinear elasticity.

We use $\mathbf{x}$ for the current position of a material point, and $\mathbf{X}$ for its reference position. Then the relationship between current position and reference position is

$$\mathbf{x} = \mathbf{X} + \mathbf{u}, \tag{3.15}$$

where $\mathbf{u}$ is the displacement. A fundamental measure of deformation is given by

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \quad \text{or} \quad \mathbf{F} = 1 + \frac{\partial \mathbf{u}}{\partial \mathbf{X}} \tag{3.16}$$

and subject to the constraint $J = \det \mathbf{F} > 0$ to ensure that material volume elements remain positive. The determinant of the deformation gradient maps a volume element in the reference configuration into the reference configuration

$$\mathrm{d}v = \det \mathbf{F} \mathrm{d}V$$

The right Cauchy-Green deformation tensor, $\mathbf{C}$, is introduced as $\mathbf{C} = \mathbf{F}^T\mathbf{F}$. Alternatively the Green strain tensor, $\mathbf{E}$, is introduced as $\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$, where $\mathbf{I}$ is

the second order identity tensor.

In current configuration, a common deformation measure is the left Cauchy-Green deformation tensor, $\mathbf{b}$, expressed as $\mathbf{b} = \mathbf{F}\mathbf{F}^T$. The Green-Almansi strain tensor, $\mathbf{e}$, may be expressed in terms of $\mathbf{b}$ as $\mathbf{e} = \frac{1}{2}(\mathbf{I} - \mathbf{b}^{-1})$.

In finite deformation problems, care must be taken to describe the configuration to which stress is measured. The Cauchy stress, $\boldsymbol{\sigma}$, and the Kirchhoff stress, $\boldsymbol{\tau}$, are measures defined with respect to the current configuration. They are related through the determinant of the deformation gradient as $\boldsymbol{\tau} = J\boldsymbol{\sigma}$. The second Piola-Kirchhoff stress, $\mathbf{S}$, is a stress measure with respect to the reference configuration and has components

$$\mathbf{S} = J\mathbf{F}^{-1}\boldsymbol{\sigma}\mathbf{F}^{-1}.$$

Therefore, the second Piola-Kirchhoff stress is related to the Kirchhoff stress through $\boldsymbol{\tau} = \mathbf{F}\mathbf{S}\mathbf{F}^T$. Finally, the first Piola-Kirchhoff stress, $\mathbf{P}$, is related to $\mathbf{S}$ through $\mathbf{P} = \mathbf{F}\mathbf{S}$.

In the current configuration, traction is given by $\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$. And in the reference configuration, traction is deduced from the first Piola-Kirchhoff stress through $\mathbf{t}_0 = \mathbf{P}\mathbf{N}$, where $\mathbf{N}$ is an unit outward pointing normal to the reference surface. Using the definition for traction and stresses, we obtain

$$\mathbf{F}^T\mathbf{n}\mathrm{d}s = J\mathbf{N}\mathrm{d}S \quad \text{and} \quad \mathrm{d}s = \sqrt{\mathbf{N}\cdot\mathbf{C}\mathbf{N}}\mathrm{d}S$$

to relate changes in the surface area and transformation of the norms.

3.4.2 Discrete deformation gradient in finite strain

In the discrete setting, the map $\mathbf{\Phi} : \mathbf{x} = \mathbf{\Phi}(\mathbf{X})$ is defined only at the nodes. The discrete deformation gradient at the node I is given by

$$\mathbf{F}_I^h := (\mathrm{Grad}^h\, \mathbf{\Phi})_I = \sum_{J \in nodes(\Omega_I)} \mathbf{x}_J \otimes \mathbf{R}^*_{IJ}, \tag{3.17}$$

where the interpolant $\mathbf{R}^*_{IJ}$ is defined relative to the reference geometry. Due to the identity (3.4), the discrete gradient in Equation (3.17) exactly preserves any homogeneous gradient, including as a special case the rigid body rotation. In fact, if $\mathbf{x}_J - \mathbf{x}_I = \mathbf{F}^c \mathbf{R}_{IJ}, \quad \forall\ J \in N_I$ and $\mathbf{R}^*_{II} = -\sum_{J \in N_I} \mathbf{R}^*_{IJ}$, we have

$$\mathbf{F}_I^h = \sum_{J \in N_I} (\mathbf{F}^c \mathbf{R}_{IJ}) \otimes \mathbf{R}^*_{IJ} = \mathbf{F}^c \Big[\sum_{J \in nodes(\Omega_I)} \mathbf{X}_J \otimes \mathbf{R}^*_{IJ} \Big] = \mathbf{F}^c. \tag{3.18}$$

A variation of the nodal positions induces a corresponding variation of the deformation gradient

$$\delta\mathbf{F}_I^h := (\mathrm{Grad}^h\, \delta\mathbf{\Phi})_I = \sum_{J \in N_I} \delta\mathbf{\Phi}_J \otimes \mathbf{R}^*_{IJ}. \tag{3.19}$$

The discrete *spatial gradient* of the displacement variation is introduced as

$$(\nabla^h \delta\mathbf{\Phi})_I = (\delta\mathbf{F}_I^h)(\mathbf{F}_I^h)^{-1}. \tag{3.20}$$

We can proceed to define discrete finite strains and their variations using the basic formulae (3.17), (3.19), and (3.20). For example,

$$\begin{aligned}
&\mathbf{C}_I^h = (\mathbf{F}_I^h)^T(\mathbf{F}_I^h) \\
&\delta\mathbf{C}_I^h = (\mathbf{F}_I^h)^T\left[(\nabla^h\delta\mathbf{\Phi})_I + (\nabla^h\delta\mathbf{\Phi})_I^T\right](\mathbf{F}_I^h) = 2(\mathbf{F}_I^h)^T(\nabla_s^h\delta\mathbf{\Phi})_I(\mathbf{F}_I^h).
\end{aligned} \tag{3.21}$$

3.4.3 Discrete weak form in finite strain

In the Lagrangian setting, the equilibrium is governed by the boundary value problem

$$\begin{aligned}
&\operatorname{Div}\mathbf{P} + \rho_0\mathbf{b} = \rho_0\ddot{\mathbf{u}}, \quad \mathbf{PF} = \mathbf{F}^T\mathbf{P}^T \quad \text{in } \Omega_0, \\
&\mathbf{PN} = \bar{\mathbf{p}} \quad \text{on } \partial\Omega_{0\mathbf{t}}, \\
&\mathbf{\Phi} = \bar{\mathbf{\Phi}} \quad \text{on } \partial\Omega_{0\mathbf{u}},
\end{aligned} \tag{3.22}$$

where $\mathbf{P}$ is the first Piola-Kirchhoff stress, $\mathbf{N}$ refers to the unit outwards normal vector on the boundary $\partial\Omega_0$, and the subscript 0 denotes quantities in the reference configuration. The continuum weak form of the boundary value problem is

$$\int_{\Omega_0} \operatorname{Grad}\delta\mathbf{\Phi} : \mathbf{P}(\mathbf{F})dA + \int_{\Omega_0} \delta\mathbf{\Phi}\cdot\rho_0\ddot{\mathbf{u}}dA - \int_{\Omega_0} \delta\mathbf{\Phi}\cdot\rho_0\mathbf{b}dA - \int_{\partial\Omega_{0\mathbf{t}}} \delta\mathbf{\Phi}\cdot\bar{\mathbf{p}}dS = 0. \tag{3.23}$$

Similar to the small strain case, the external nodal force $\mathbf{f}_I^{ext}$ is the resultant of the forces applied on the cell I,

$$\mathbf{f}_I^{ext} = \int_{V_I} \rho_0\mathbf{b}dA + \int_{\partial V_I\cap\partial\Omega_{0t}} \bar{\mathbf{p}}dS. \tag{3.24}$$

The discrete weak form can be stated as: find the current nodal positions $\mathbf{x}_I$, such that

$$\sum_I A_{0I}(\mathrm{Grad}^h\, \delta\mathbf{\Phi})_I : \mathbf{P}(\mathbf{F}_I^h) - \sum_I \delta\mathbf{\Phi}_I \cdot \mathbf{f}_I^{ext} = 0, \tag{3.25}$$

for any kinematically admissible variation of the nodal position, subject to the discrete kinematics relation (3.17) and the constitutive law $\mathbf{P} = \mathbf{P}(\mathbf{F}^h)$. Introduce the internal force vector $\mathbf{f}_I^{int}$ such that

$$\sum_I A_{0I}(\mathrm{Grad}^h\, \delta\mathbf{\Phi})_I : \mathbf{P}(\mathbf{F}_I^h) = \sum_I \delta\mathbf{\Phi}_I \cdot \mathbf{f}_I^{int}. \tag{3.26}$$

The discrete weak form gives a set of nonlinear equations

$$\mathbf{f}_I^{int} + m_I \ddot{\mathbf{u}}_I - \mathbf{f}_I^{ext} = \mathbf{0} \quad \forall I, \tag{3.27}$$

or, in the assembled global system,

$$\mathbf{F}^{int} + \mathbf{M}\ddot{\mathbf{d}} - \mathbf{F}^{ext} = \mathbf{0}. \tag{3.28}$$

3.5 Solution strategy: Newton's method

Newton's method is employed to solve the nonlinear equations. If we record the residual as $\mathbf{R} = \mathbf{F}^{ext} - \mathbf{F}^{int} - \mathbf{M}\ddot{\mathbf{d}}$. To implement a Newton method, it is necessary to linearize the residual equation

$$\mathbf{R}^{(k+1)} = \mathbf{R}^{(k)} + \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}|^{(k)} d\boldsymbol{\alpha}^{(k)} = \mathbf{0},$$

where $\boldsymbol{\alpha}$ is one the variables at time t_{n+1}. We define tangent matrix as

$$\mathbf{S}^{(k)} = -\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}$$

and solve

$$\mathbf{S}^{(k)} d\boldsymbol{\alpha}^{(k)} = \mathbf{R}^{(k)}.$$

The solution is updated using

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} + d\boldsymbol{\alpha}^{(k)}.$$

The tangent matrix for of the nonlinear equations may be expressed in terms of the incremental displacements, velocity or acceleration. For example, consider the case where the solution is parameterized in terms of increments of the displacement,

$$\mathbf{S} d\mathbf{d} = -\frac{\partial \mathbf{R}}{\partial \mathbf{d}} d\mathbf{d} - \frac{\partial \mathbf{R}}{\partial \mathbf{v}} \frac{\partial \mathbf{v}}{\partial \mathbf{d}} d\mathbf{d} - \frac{\partial \mathbf{R}}{\partial \mathbf{a}} \frac{\partial \mathbf{a}}{\partial \mathbf{d}} d\mathbf{d}, \tag{3.29}$$

where $\mathbf{v} = \dot{\mathbf{u}}$ and $\mathbf{a} = \ddot{\mathbf{u}}$. We do not show dependence on the iteration (k) for simplicity of notation. From above equation, we observe that

$$\mathbf{K} = -\frac{\partial \mathbf{R}}{\partial \mathbf{d}}, \quad \mathbf{C} = -\frac{\partial \mathbf{R}}{\partial \mathbf{v}}, \quad \mathbf{M} = -\frac{\partial \mathbf{R}}{\partial \mathbf{a}} \tag{3.30}$$

define the tangent stiffness, damping and mass, respectively. For materials employed in our work, the damping matrix $\mathbf{C}$ is set be zero.

For transient problem, the Newmark's method define following parameters during solution

$$\frac{\partial \ddot{\mathbf{d}}}{\partial \mathbf{d}} = \frac{1}{\beta \Delta t^2}, \quad \frac{\partial \dot{\mathbf{d}}}{\partial \mathbf{d}} = \frac{\gamma}{\beta \Delta t}. \tag{3.31}$$

Thus, the tangent matrix can be expressed as

$$\mathbf{S} = \mathbf{K} + \frac{\gamma}{\beta \Delta t}\mathbf{C} + \frac{1}{\beta \Delta t^2}\mathbf{M}. \tag{3.32}$$

3.5.1 The stiffness matrix

Linearization of the stress divergence term is often the most involved step in computations and should be considered in a reference configuration representation. The evaluation of the stiffness matrix $\mathbf{K}$ proceeds as follow. Invoking the following continuum mechanics result regarding the stress power

$$\delta \mathbf{F} : \mathbf{P} = \frac{1}{2}\delta \mathbf{C} : \mathbf{S} = \nabla_s(\delta \mathbf{\Phi}) : \boldsymbol{\tau}, \tag{3.33}$$

where $\mathbf{S} = \mathbf{F}^{-1}\mathbf{P}$ is the second Piola-Kirchhoff stress and $\boldsymbol{\tau} = \mathbf{F}\mathbf{S}\mathbf{F}^T$ is the Kirchhoff stress, the increment of stress power under incremental displacement $\Delta \mathbf{x}$ can be derived as

$$\Delta(\delta \mathbf{F} : \mathbf{P}) = \nabla_s(\delta \mathbf{\Phi}) : [\mathbb{C} + \mathbf{I} \boxtimes \boldsymbol{\tau}] : \nabla_s(\Delta \mathbf{x}), \tag{3.34}$$

where $\mathbb{C}$ is the spatial material tangent tensor, in components

$$C_{abcd} = 2F_{aA}F_{bB}\frac{\partial S_{AB}}{\partial C_{CD}}F_{cC}F_{dD}. \tag{3.35}$$

The operator $\cdot \boxtimes \cdot$ defines the Kronecker product of two second-order tensors, in components $[\mathbf{I} \boxtimes \boldsymbol{\tau}]_{abcd} = \delta_{ac}\tau_{bd}$. Carrying these results to the discrete setting, the increment of the stress power over the cell I (modulo the area factor) can be written as

$$\Delta((\mathrm{Grad}^h\, \delta\boldsymbol{\Phi})_I : \mathbf{P}(\mathbf{F}_I^h)) = (\nabla_s^h \delta\boldsymbol{\Phi})_I : [\mathbb{C} + \mathbf{I} \boxtimes \boldsymbol{\tau}] : (\nabla_s^h(\Delta\mathbf{x}))_I. \tag{3.36}$$

Therefore, the cell-level stiffness associated with cell I is

$$\mathbf{K}_{KL} = A_{0I}\mathbf{B}_{IK}^T\big[\mathbb{C} + \mathbf{I} \boxtimes \boldsymbol{\tau}\big]\mathbf{B}_{IL}, \quad K, L \in \{I\} \cup N_I, \tag{3.37}$$

where $\mathbf{B}_{IJ}$ is the strain-displacement matrix corresponding to the spatial basis $\mathbf{r}_{IJ}^*$, which is updated at every iteration step.

3.6 Necessary condition to satisfy patch test

The patch test [97] has been widely used to assess the performance of numerical formulations despite the controversy as to whether satisfying the patch test is necessary for convergence. In the continuum case, it is now understood that the use of linearly complete shape functions does not necessarily guarantee a linear exactness in the solution; in fact, additional requirements must be placed on the test function and numerical integration scheme in order to pass the patch test [25, 59, 60]. In this section, we will make a similar observation for the discrete methods.

We discuss this issue in the context of large strain formulation; the small strain case proceeds in the same manner. The linear patch test requires that a linear solution $\mathbf{x}_I = \mathbf{F}^c\mathbf{X}_I + \mathbf{a}$ exactly satisfies the discrete equilibrium equations (3.27) cor-

responding to a zero body force boundary static value problem with the essential and the natural boundary conditions prescribed according to the assumed linear motion. To satisfy this requirement, the discrete gradient must be linearly complete, so that any homogeneous deformation gradient is exactly recovered. In addition, the discrete equilibrium equations (3.27) must be identically satisfied for any homogeneous stress $\mathbf{P}^c$ under the application of boundary traction corresponding to the given homogeneous stress. The second requirement places restrictions on the discrete variation of deformation gradient $\delta\mathbf{F}^h_I$. Consider the general representation

$$\delta\mathbf{F}^h_I = \sum_{J\in nodes(\Omega_I)} \delta\mathbf{\Phi}_J \otimes \mathbf{R}^*_{IJ}. \tag{3.38}$$

Substituting this expression into Equation (3.26), setting $\mathbf{P} = \mathbf{P}^c$ and re-arranging the sums, we identify

$$\mathbf{f}^{int}_I = \sum_{J\in nodes(\Omega_I)} A_{0J}\mathbf{P}^c\mathbf{R}^*_{JI}. \tag{3.39}$$

According to Equation (3.24), in the absence of the body force, the external nodal force must either be zero for interior nodes, or equal the resultant of the boundary traction. Hence,

$$\mathbf{f}^{int}_I := \sum_{J\in nodes(\Omega_I)} A_{0J}\mathbf{P}^c\mathbf{R}^*_{JI} = \begin{cases} \mathbf{0} & \text{if } I \text{ is an interior node} \\ \sum_{K\in B_I} S_{IK}\mathbf{P}^c\mathbf{N}_{IK} & \text{if } I \text{ is a boundary node,} \end{cases} \tag{3.40}$$

here $\mathbf{P}^c$ is any homogeneous Piola-Kirchhoff stress, and $\mathbf{N}_{IK}$ is the unit normal vector

on the exterior edge indexed by K. Since $\mathbf{P}^c$ is arbitrary, this equality dictates

$$\sum_{J \in nodes(\Omega_I)} A_{0J}\mathbf{R}^*_{JI} = \begin{cases} \mathbf{0} & \text{if } I \text{ is an interior node} \\ \sum_{K \in B_I} S_{IK}\mathbf{N}_{IK} & \text{if } I \text{ is a boundary node.} \end{cases} \tag{3.41}$$

This is the discrete counterpart of the *integration constraint* derived by Chen et al. [25].

We proved that the condition (3.41) is necessary for any proposed discrete gradient operator to satisfy patch test.

CHAPTER 4
VORONOI CELL DISCRETE GRADIENT METHOD

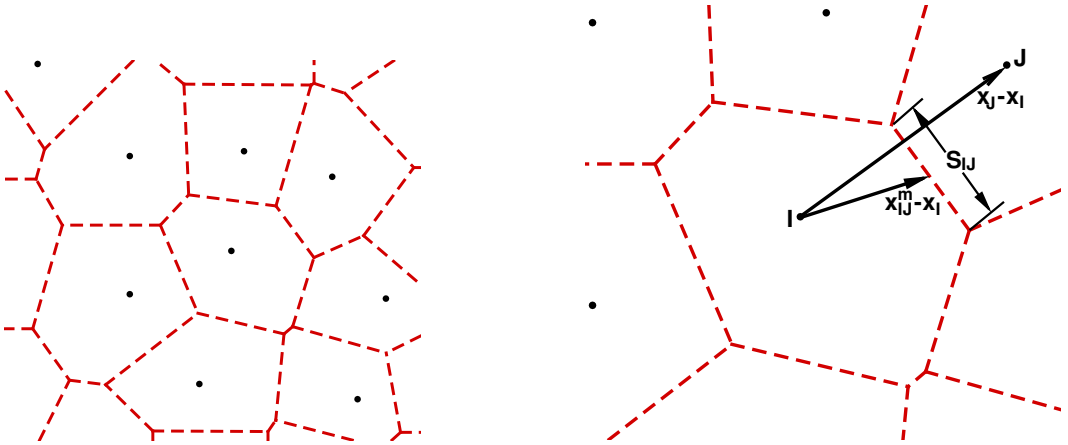
This chapter introduces a Galerkin formation that builds on discrete gradient defined on Voronoi cells. The formulation is presented in the context of two-dimensional elasticity; the theory, however, applies to any dimension. The Voronoi diagram is employed to identify the nodal support of the gradient operator and to partition the domain into a set of non-overlapping cells. Compared to schemes that rely solely on Euclidean distance to define the computational cell for discrete operators [56, 63, 64, 83], the use of the Voronoi diagram takes into account both distance and neighborliness, and yields a natural (and in some sense an optimal) choice of computational cells. The tessellation process is also fully automatic and does not require algorithmic parameters such as the size of influence region. The global weak form is evaluated using nodal integration, and thus, Gauss integration is eliminated. The gradient interpolant has the Kronecker delta property; hence, it is straightforward to introduce essential boundary conditions and to couple the method with the finite element method.

The chapter is organized as follows. In Section 4.1, the discrete gradient operator is introduced. Section 4.2 details the implementation of this method in linear elasticity, followed by a description of the extension to geometrically nonlinear problems. Section 4.3 discusses the issue of patch test and provides proof for the fact that the current formulation satisfies the linear displacement patch test.

4.1 Discrete gradient over Voronoi cell

4.1.1 Partition with Voronoi cell

In our application, the domain Ω is bounded. The boundary $\partial\Omega$ is modeled by closed planar line segments. We redefine the computational cell as the intersection $\mathcal{V}_I \cap \Omega$ where $\mathcal{V}_I$ is the original Voronoi cell, which could be unbounded. If $\mathcal{V}_I \cap \Omega = \mathcal{V}_I$, the cell is an interior cell, and the corresponding node is an interior node. Figure 4.1(b) shows an example of an interior cell. If $V_I \cap \Omega \subsetneqq \mathcal{V}_I$, the cell is identified as a boundary cell. The boundary of the cell I is define by $\overline{(\Omega_I \cap \Omega)} \backslash (\Omega_I \cap \Omega)$, which consists of closed straight edges. In the definition, the number of neighboring nodes is always the same as the number of edges for any interior cell, while the number of neighboring nodes is always less than the number of edges for any boundary cell. Every edge of an interior cell is an interior edge that is shared by two neighboring cells. Cell edges that fall on $\partial\Omega$ are called exterior edges; the set of exterior edges of the cell I is denoted by B_I. Due to the closeness, an interior node has at least three neighbors (in general, $n_d + 1$ neighbors, where n_d is the dimension of the space). The nodes on boundary $\partial\Omega$ are called boundary nodes. In this work, it is required that the nodal distribution on the boundary be sufficiently dense so that every two adjacent boundary nodes are natural neighbors to each other. For reasons that will become clear in Section 4.1.4, we impose an additional requirement on boundary cells, that an admissible boundary cell must have at least three natural neighbors. An example of admissible boundary cells is illustrated in Figure 4.2. In the following, we will denote the computation cell $\Omega_I \cap \Omega$ simply by Ω_I.



(a) (b)

Figure 4.1: Vectors defined over Voronoi cells. (a) An example of Voronoi tessellation; (b) Notations used in the chapter.

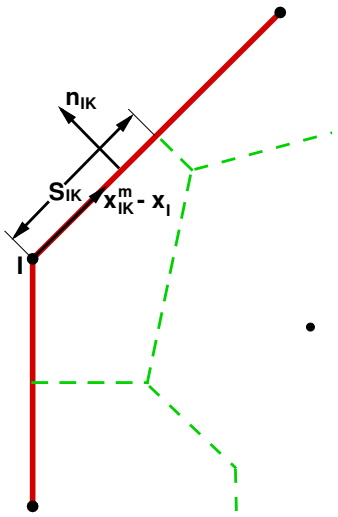


Figure 4.2: Illustrations of a boundary cells.

4.1.2 A tensorial identity for Voronoi diagram

The starting point for deriving the discrete gradient operator is the Gauss theorem

$$\int_{\Omega} \nabla \boldsymbol{\varphi} da = \int_{\partial\Omega} \boldsymbol{\varphi} \mathbf{n} ds, \tag{4.1}$$

where Ω is a domain in the Euclidean space, $\partial\Omega$ is its boundary, and $\mathbf{n}$ is the unit outwards normal on the boundary. For $\Omega \subset \mathbb{R}^2$, setting $\boldsymbol{\varphi} = 1$, $\boldsymbol{\varphi} = x$ and $\boldsymbol{\varphi} = y$ respectively, we have

$$\int_{\partial\Omega} \mathbf{n} ds = \mathbf{0}, \quad \int_{\partial\Omega} \mathbf{X} \otimes \mathbf{n} ds = A\mathbf{I}, \tag{4.2}$$

where A is the area of Ω, $\mathbf{X} = (X, Y)$, and $\mathbf{I}$ is the identity tensor. With reference to the Voronoi cell depicted in Figure 4.1(b), let S_{IJ} to be the length of the edge common to cell I and cell J, and let $R_{IJ} = \|\mathbf{X}_J - \mathbf{X}_I\|$. Applying the identities $(4.2)_1$ to the Voronoi cell I, and given that the normal vector of the edge IJ is $\mathbf{n}_{IJ} = \frac{\mathbf{X}_J - \mathbf{X}_I}{R_{IJ}}$, we obtain

$$\sum_{J \in N_I} \frac{S_{IJ}}{R_{IJ}} (\mathbf{X}_J - \mathbf{X}_I) = \mathbf{0}, \tag{4.3}$$

This identity is the basis of the non-Sibsonian interpolant introduced by Belikov et al. [7, 8]. Sukumar et al. [92] established a Galerkin meshfree method using the non-Sibsonian interpolation. Specifying Equation $(4.2)_2$ to the Voronoi cell and using the

trapezoidal rule on the boundary integral yield

$$\frac{1}{A_I}\sum_{J\in N_I}\frac{S_{IJ}}{R_{IJ}}\mathbf{X}_{IJ}^m\otimes(\mathbf{X}_J-\mathbf{X}_I)=\mathbf{I}, \tag{4.4}$$

where A_I is the area of the Voronoi cell Ω_I, and $\mathbf{X}_{IJ}^m$ is the middle point of the edge IJ. Note that the trapezoidal rule is exact in this particular case. From the vector identities in Equation (4.3) and (4.4) we can further conclude that

$$\frac{1}{A_I}\sum_{J\in N_I}\frac{S_{IJ}}{R_{IJ}}(\mathbf{X}_{IJ}^m-\mathbf{a})\otimes(\mathbf{X}_J-\mathbf{X}_I)=\mathbf{I} \tag{4.5}$$

for any vector $\mathbf{a}$. In particular, setting $\mathbf{a}=\mathbf{X}_I$, we obtain the desired tensorial identity

$$\frac{1}{A_I}\sum_{J\in N_I}\frac{S_{IJ}}{R_{IJ}}(\mathbf{X}_{IJ}^m-\mathbf{X}_I)\otimes(\mathbf{X}_J-\mathbf{X}_I)=\mathbf{I}. \tag{4.6}$$

Introducing vectors

$$\mathbf{R}_{IJ}^*=\frac{S_{IJ}}{A_I R_{IJ}}(\mathbf{X}_{IJ}^m-\mathbf{X}_I),\quad \mathbf{R}_{II}^*=-\sum_{J\in N_I}\mathbf{R}_{IJ}^*, \tag{4.7}$$

it follows from the definition (4.7) and the identity (4.6) that

$$\begin{aligned}\sum_{J\in nodes(\Omega_I)}\mathbf{R}_{IJ}^*&=\sum_{J\in N_I}\mathbf{R}_{IJ}^*+\mathbf{R}_{II}^*=\mathbf{0},\\ \sum_{J\in nodes(\Omega_I)}\mathbf{R}_{IJ}^*\otimes\mathbf{X}_J&=\sum_{J\in N_I}\mathbf{R}_{IJ}^*\otimes\mathbf{X}_J+\mathbf{R}_{II}^*\otimes\mathbf{X}_I=\mathbf{I}\end{aligned} \tag{4.8}$$

Equation $(4.8)_2$ can be written equivalently as

$$\begin{aligned}\sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ} &= \sum_{J\in N_I} \mathbf{R}^*_{IJ}X_J + \mathbf{R}^*_{II}X_I = \mathbf{e}_X,\\ \sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ} &= \sum_{J\in N_I} \mathbf{R}^*_{IJ}Y_J + \mathbf{R}^*_{II}Y_I = \mathbf{e}_Y,\end{aligned} \tag{4.9}$$

where ($\mathbf{e}_X$, $\mathbf{e}_Y$) are the basis vectors. The tensorial identity (4.6) or its equivalent in (4.8) plays a pivoting role in the ensuing development discrete gradient.

4.1.3 Discrete gradient operator at an interior node

Considering a continuous scalar function $u : \mathbb{R}^2 \mapsto \mathbb{R}$, we define the discrete gradient at an interior node I as

$$(\nabla^h u)_I = \sum_{J\in nodes(\Omega_I)} \mathbf{R}^*_{IJ}u_J = \sum_{J\in N_I} \mathbf{R}^*_{IJ}(u_J - u_I) = \sum_{J\in N_I} \mathbf{R}^*_{IJ}u_J + \mathbf{R}^*_{II}u_I. \tag{4.10}$$

This gradient is linearly complete in the sense that it can exactly reproduce the homogeneous gradient of any linear function. The linear completeness follows directly from the properties recorded in Equation (4.8), which is same as the conditions listed in Equation (3.4).

From the linear completeness, we can also estimate the truncation error. Let h_I be the characteristic length of the cell A_I, say $h_I = \max_{J\in N_I}(S_{IJ}, R_{IJ})$. Let us assume that the function $u(X, Y)$ admits the Taylor series expansion at $\mathbf{X}_I$, so that

$$u_J - u_I = u_{,X}(X_J - X_I) + u_{,Y}(Y_J - Y_I) + \mathscr{O}(h_I^2). \tag{4.11}$$

We say a family of Voronoi cell is regular if there are two constants $c_1, c_2 \in (0, \infty)$, such that $c_i S_{IJ} \leq R_{IJ} \leq c_2 S_{IJ}$ for every cell A_I in the family. Assuming the Voronoi diagram is from a regular family, an examination of $\mathbf{R}^*_{IJ}$ in Equation (4.7) shows that $\mathbf{R}^*_{IJ} = \mathscr{O}(h_I^{-1})$. Substituting the Taylor expansion into Equation (4.10), we find

$$(\nabla^h u)_I = \sum_{J \in N_I} \mathbf{R}^*_{IJ}[u_{,X}(X_J - X_I) + u_{,Y}(Y_J - Y_I) + \mathscr{O}(h_I^2)] = u_{,X}\mathbf{e}_X + u_{,Y}\mathbf{e}_Y + \mathscr{O}(h_I), \tag{4.12}$$

namely $(\nabla^h u)_I - \nabla u(\mathbf{X}_I) = \mathscr{O}(h_I)$. The first derivatives are exactly preserved due to the linear exactness.

4.1.4 Discrete gradient at a boundary node

The discrete gradient described above is based on the Gauss theorem (4.1). Essentially, the average of gradient over an Voronoi cell, evaluated from the boundary integral $\frac{1}{A_I}\int_{\partial\Omega} u\mathbf{n}ds$, is used to approximate the gradient of u at the corresponding node. A boundary cell defined in this chapter always has exterior edges that are not associated with neighboring nodes. In this case, the sum $\sum_{J \in N_I} \mathbf{R}^*_{IJ}(u_J - u_I)$ reflects an integral over a partial boundary, and thus the gradient formula needs to be modified.

Let B_I denote the set of exterior edges of a boundary cell A_I. With reference to the notations introduced in Figure 4.2, the vector identities in Equation (4.3) for

a boundary cell become

$$\begin{aligned}
&\sum_{J\in N_I}\frac{S_{IJ}}{R_{IJ}}(\mathbf{X}_{IJ}^m-\mathbf{X}_I)+\sum_{K\in B_I}S_{IK}\mathbf{n}_{IK}=\mathbf{0},\\
&\frac{1}{A_I}\sum_{J\in N_I}\frac{S_{IJ}}{R_{IJ}}(\mathbf{X}_{IJ}^m-\mathbf{X}_I)\otimes(\mathbf{X}_J-\mathbf{X}_I)+\frac{1}{A_I}\sum_{K\in B_I}S_{IK}(\mathbf{X}_{IK}^m-\mathbf{X}_I)\otimes\mathbf{n}_{IK}=\mathbf{I},
\end{aligned}\tag{4.13}$$

where S_{IK} is the length of the exterior edge indexed by K, and $\mathbf{n}_{IK}$ is the unit outwards normal of this edge. The gradient is defined indirectly by

$$(\nabla^h u)_I=\sum_{J\in N_I}\mathbf{R}_{IJ}^*(u_J-u_I)+\left[\frac{1}{A_I}\sum_{K\in B_I}S_{IK}(\mathbf{X}_{IK}^m-\mathbf{X}_I)\otimes\mathbf{n}_{IK}\right](\nabla^h u)_I. \tag{4.14}$$

That is,

$$\left[\mathbf{I}-\frac{1}{A_I}\sum_{K\in B_I}S_{IK}(\mathbf{X}_{IK}^m-\mathbf{X}_I)\otimes\mathbf{n}_{IK}\right](\nabla^h u)_I=\sum_{J\in N_I}\mathbf{R}_{IJ}^*(u_J-u_I). \tag{4.15}$$

Due to the relation in Equation $(4.13)_2$, the tensor inside the square bracket equals $\sum_{J\in N_I}\mathbf{R}_{IJ}^*\otimes(\mathbf{X}_J-\mathbf{X}_I)$. By our requirement of admissible boundary cell, a boundary cell must have at least two distinct neighboring nodes, implying that the matrix $[\sum_{J\in N_I}\mathbf{R}_{IJ}^*\otimes(\mathbf{X}_J-\mathbf{X}_I)]$ has a full rank. It follows that

$$(\nabla^h u)_I=\sum_{J\in N_I}\left[\sum_{K\in N_I}\mathbf{R}_{IK}^*\otimes(\mathbf{X}_K-\mathbf{X}_I)\right]^{-1}\mathbf{R}_{IJ}^*(u_J-u_I)\equiv\sum_{J\in N_I}\hat{\mathbf{R}}_{IJ}^*u_J+\hat{\mathbf{R}}_{II}^*u_I, \tag{4.16}$$

where

$$\hat{\mathbf{R}}^*_{IJ} = \left[\sum_{K\in N_I} \mathbf{R}^*_{IK} \otimes (\mathbf{X}_K - \mathbf{X}_I)\right]^{-1} \mathbf{R}^*_{IJ}, \qquad \hat{\mathbf{R}}^*_{II} = -\sum_{J\in N_I} \hat{\mathbf{R}}^*_{IJ}. \tag{4.17}$$

Equation (4.16) is the modified gradient operator for a boundary cell. The modified interpolant obviously satisfies

$$\sum_{J\in N_I} \hat{\mathbf{R}}^*_{IJ} + \hat{\mathbf{R}}^*_{II} = \mathbf{0}. \tag{4.18}$$

Furthermore,

$$\begin{aligned} &\sum_{J\in N_I} \hat{\mathbf{R}}^*_{IJ} \otimes \mathbf{X}_J + \hat{\mathbf{R}}^*_{II} \otimes \mathbf{X}_I = \sum_{J\in N_I} \hat{\mathbf{R}}^*_{IJ} \otimes (\mathbf{X}_J - \mathbf{X}_I) \\ =&\left[\sum_{K\in N_I} \mathbf{R}^*_{IK} \otimes (\mathbf{X}_K - \mathbf{X}_I)\right]^{-1} \left[\sum_{K\in N_I} \mathbf{R}^*_{IK} \otimes (\mathbf{X}_K - \mathbf{X}_I)\right] = \mathbf{I}. \end{aligned} \tag{4.19}$$

The properties in Equation (4.18) and (4.19) guarantee the linear completeness of the modified gradient. In addition, following the argument in Section 4.1.3, we can also conclude that the truncation error for cells in a regular family remains order $\mathscr{O}(h_I)$.

In closing this section, it is noted that the discrete gradient operator (4.10) coincides with the formula derived by Eymard et al. [44]. The boundary cell formula (4.16), however, is different. Eymard et al. [44] considered only Dirichlet boundary condition.

4.2 Discrete Galerkin formulation

The framework of the implementation of the discrete gradient Galerkin formulation in linear elasticity and finite strain are demonstrated in Section 3.3 and 3.4. For interior cell, the discrete formulations are exactly same in previous chapter. Here, we show the different process to handle boundary points because boundary cells are not entire Voronoi cells.

4.2.1 Linear elasticity formulation

Invoking Equation (4.16), the discrete strain over a boundary node is given by

$$(\nabla^h \mathbf{u})_I = \sum_{J \in N_I} \mathbf{u}_J \otimes \hat{\mathbf{R}}^*_{IJ} + \mathbf{u}_I \otimes \hat{\mathbf{R}}^*_{II} u_I. \tag{4.20}$$

Consequently,

$$[\boldsymbol{\varepsilon}^h_I] = \sum_{J \in N_I} \hat{\mathbf{B}}_{IJ} \mathbf{u}_J + \hat{\mathbf{B}}_{II} \mathbf{u}_I, \tag{4.21}$$

where $\hat{\mathbf{B}}_{IK}$ is the strain-displacement matrix formulated using the vectors $\hat{\mathbf{R}}^*_{IK}$.

The variation of strain, however, is evaluated differently. On a natural boundary, the normal gradient $(\nabla \mathbf{u})\mathbf{n}$ is prescribed. The variation of the corresponding term in Equation (4.14) should be kept zero; hence,

$$(\nabla^h \delta\mathbf{u})_I = \sum_{J \in N_I} \delta\mathbf{u}_J \otimes \mathbf{R}^*_{IJ} + \delta\mathbf{u}_I \otimes \mathbf{R}^*_{II} \quad \Rightarrow \quad [\delta\boldsymbol{\varepsilon}^h_I] = \sum_{J \in N_I} \mathbf{B}_{IJ} \delta\mathbf{u}_J + \mathbf{B}_{II} \delta\mathbf{u}_I. \tag{4.22}$$

The use of Equation $(4.22)_2$ proves to be crucial in passing the linear displacement patch test, as discussed in Section 4.3. This scheme, however, leads to an unsymmet-

rical stiffness matrix

$$\mathbf{K}_{KL} = A_I \mathbf{B}_{IK}^T \mathbf{D} \hat{\mathbf{B}}_{IL}, \qquad K, L \in \{I\} \cup N_I. \tag{4.23}$$

Note that this modification applies only to cells having natural boundaries.

4.2.2 Finite strain formulation

For the finite strain formulation, the discrete deformation gradient at interior node I is given by

$$\mathbf{F}_I^h := (\mathrm{Grad}^h \, \mathbf{\Phi})_I = \sum_{J \in N_I} (\mathbf{X}_J - \mathbf{X}_I) \otimes \mathbf{R}_{IJ}^*, \tag{4.24}$$

where the interpolant $\mathbf{R}_{IJ}^*$ is defined relative to the reference geometry, namely,

$$\mathbf{R}_{IJ}^* = \frac{S_{IJ}}{A_{0J} R_{IJ}} (\mathbf{X}_{IJ}^m - \mathbf{X}_I), \quad \mathbf{R}_{II}^* = -\sum_{J \in N_I} \mathbf{R}_{IJ}^*. \tag{4.25}$$

Denoting by $\mathbf{R}_{IJ} = \mathbf{X}_J - \mathbf{X}_I$, the equality (4.3) can be written as $\sum_{J \in N_I} \mathbf{R}_{IJ} \otimes \mathbf{R}_{IJ}^* = \mathbf{I}$, suggesting that the vectors $\mathbf{R}_{IJ}^*$ play a duality role to $\mathbf{R}_{IJ}$. Due to the identity (4.3), the discrete gradient in Equation (4.24) exactly preserves any homogeneous gradient, including as a special case the rigid body rotation.

For a boundary cell with a natural boundary, the gradient and its variation

are computed according to

$$\mathbf{F}_I^h = \sum_{J\in N_I} (\mathbf{X}_J - \mathbf{X}_I) \otimes \hat{\mathbf{R}}_{IJ}^*, \quad \delta\mathbf{F}_I^h = \sum_{J\in N_I} (\delta\mathbf{\Phi}_J - \delta\mathbf{\Phi}_I) \otimes \mathbf{R}_{IJ}^*, \tag{4.26}$$

where $\hat{\mathbf{R}}_{IJ}^* = \left[\sum_{K\in N_I} \mathbf{R}_{IK}^* \otimes (\mathbf{X}_K - \mathbf{X}_I)\right]^{-1} \mathbf{R}_{IJ}^*$. Consequently,

$$\begin{aligned} (\nabla^h \Delta\mathbf{X})_I &= \sum_{J\in N_I} (\Delta\mathbf{X}_J - \Delta\mathbf{X}_I) \otimes \hat{\mathbf{r}}_{IJ}^*, \qquad \hat{\mathbf{r}}_{IJ}^* = (\mathbf{F}_I^h)^{-T} \hat{\mathbf{R}}_{IJ}, \\ (\nabla^h \delta\mathbf{\Phi})_I &= \sum_{J\in N_I} (\delta\mathbf{\Phi}_J - \delta\mathbf{\Phi}_I) \otimes \mathbf{r}_{IJ}^*, \qquad \mathbf{r}_{IJ}^* = (\mathbf{F}_I^h)^{-T} \mathbf{R}_{IJ}. \end{aligned} \tag{4.27}$$

The tangent stiffness is modified accordingly, as

$$\mathbf{K}_{KL} = A_{0I} \mathbf{B}_{IK}^T \left[\mathbb{C} + \mathbf{I} \boxtimes \boldsymbol{\tau}\right] \hat{\mathbf{B}}_{IL}, \quad K, L \in \{I\} \cup N_I, \tag{4.28}$$

where $\hat{\mathbf{B}}_{IJ}$ is the strain-displacement matrix formulated from $\hat{\mathbf{r}}_{IJ}^*$.

4.3 Patch test

In section 3.6, we discussed the necessary conditions to satisfy patch test. The discrete gradient must be linearly complete and the condition (3.41). In this section, we will show that the present formulation identically satisfies the condition (3.41) for linear displacement patch test.

Recalling Equation (4.25), we find

$$A_{0I} \mathbf{R}_{II}^* + \sum_{J\in N_I} A_{0J} \mathbf{R}_{JI}^* = -\sum_{J\in N_I} \frac{S_{IJ}}{R_{IJ}} (\mathbf{X}_J - \mathbf{X}_I). \tag{4.29}$$

The right-hand-side is the discrete form of the integral $-\int \mathbf{N} dS$ over the interior edges of a cell. For an interior cell, the sum vanishes, according to Equation $(4.3)_1$. For a boundary cell, the sum equals $\sum_{K \in B_I} S_{IK} \mathbf{N}_{IK}$, according to Equation $(4.13)_1$. Therefore, the condition (3.41) is satisfied by construction. Note that while the vector interpolant $\mathbf{R}^*_{IJ}$ defined in Equation (4.7) remains linearly complete if the vector $\mathbf{X}_I$ is replaced by any vector $\mathbf{a}$, as implied by the identity in Equation (4.5), the insertion of $\mathbf{X}_I$ is crucial for satisfying the condition (3.41). Also, the following *consistent variation* $\delta \mathbf{F}^I = \delta \mathbf{\Phi}_I \otimes \hat{\mathbf{R}}^*_{II} + \delta \mathbf{\Phi}_J \otimes \hat{\mathbf{R}}^*_{IJ}$, which renders the stiffness matrix symmetric for a boundary cell, does not satisfy Equation (3.41) in general, and thus will fail the patch test.

4.4 Comparison with nodal integration natural element method

To put the current method further into perspective, we provide a short comparison between the present method and the recently published nodal integration natural element method (nodal-NEM) [51, 103]. The nodal-NEM starts from a continuous approximation of the displacement field

$$u(\mathbf{x}) = \sum_{J \in N_x} \phi_J(\mathbf{x}) \mathbf{u}_J, \tag{4.30}$$

where ϕ_I denotes the natural neighbor shape functions, which could be either the Sibsonian interpolants [87, 93] or the non-Sibsonian interpolants [7, 8, 92], and N_x stands for the set of nodes for which the associated shape function supports cover the point $\mathbf{x}$. The continuous strain is then averaged over a Voronoi cell (say cell I) to

give a approximate nodal strain at the corresponding node:

$$\tilde{\epsilon}^h(\mathbf{x}_I) = \sum_{J \in G_I} \tilde{\mathbf{B}}_J(\mathbf{x}_I)\mathbf{u}_J, \tag{4.31}$$

where the set G_I includes all the nodes for which the associated shape function supports overlap with the cell I. The ensuing node-wise approximate strain is substituted into the weak form to derive a discrete system, see [25,51,103] for further details. The Sibsonian and non-Sibsonian interpolants have Kronecker delta property; hence, the nodal parameters are the physical displacements. In addition, under mild restrictions the interpolation (4.30) reduces to a linear distribution over convex or non-convex boundaries [31, 92, 93]. Thus, a linear essential boundary condition can be exactly imposed. It is evident that, despite a different starting point, the current method has a number of properties in common with the nodal-NEM at the end.

Nevertheless, there are some subtle and yet important distinctions at the algorithmic level. A key difference between the discrete strain (4.21) and the algorithmic strain in Equation (4.31) is that the latter has, in general, a larger nodal support (the set of nodes that contributes to the nodal strain). The nodal support G_I includes not only the natural neighbors of the node I but also possibly some second-layer neighbors. An example of such a case is shown in Figure 4.3. Consider the approximate strain at the center node I. The support of the Sibsonian (and non-Sibsonian) shape functions is the union of all Delaunay circumcircles that pass through the corresponding node [92, 93]; here, the shape functions ϕ_I and ϕ_1 through ϕ_8 all have non-zero

distribution over the Voronoi cell I. The approximate strain (4.31) at the node I, therefore, depends on nodal values $\mathbf{u}_1$ through $\mathbf{u}_8$ (the average gradient of ϕ_I is zero due to symmetry). In contrast, the discrete strain (4.21) depends only on nodal values $\mathbf{u}_1$ through $\mathbf{u}_4$, resulting in a smaller bandwidth in the final system. In addition, the computation of the discrete strain is more straightforward. There is no need to compute the new tessellation due to the insertion of nodes at integration points.

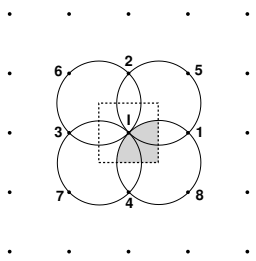


Figure 4.3: A rectangular grid to illustrate the difference in nodal support between the nodal-NEM and the present method. The Sibsonian (and non-Sibsonian) shape function at the corner node 8 has a non-zero distribution over the shaded region.

CHAPTER 5
BARYCENTRIC DISCRETE GRADIENT METHOD

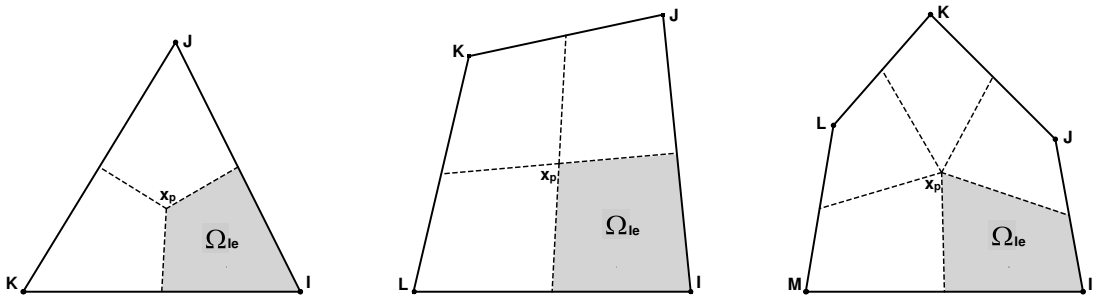
In this chapter, we introduce a new discrete formulation for domains originally discretized by a polygon mesh. The method retains the essential features of Chapter 4, but differs in the ways of domain partition and gradient computation. We assume that a mesh already exists; however, the mesh can contain an arbitrary combination of triangle, quadrilateral, and general convex polygons. Instead of using the natural neighbors as Chapter 4, we make use of the existing connectivity in the polygon mesh. We provide a unified geometric approach for constructing a family of linearly exact discrete gradient operators that applies to an arbitrary combination of polygon elements. The final Galerkin formulation also exactly passes the linear displacement patch test; however, unlike Chapter 4, the stiffness matrix is symmetric.

The chapter begins with the introduction of a polygon subdivision and the notion of subdivision cell. The discrete gradient operator and its property are discussed in Section 5.4, followed by the explicit formulations of the gradient interpolants in triangle, quadrilateral, and pentagon. The discrete Galerkin formulation in 2D linear elasticity is contained in Section 5.3. The aspect of patch test is discussed in Section 5.5.

5.1 Polygon subdivision

The proposed method begins with a partition of a polygon mesh into node-based cells. This is achieved by a subdivision process that generalizes the barycentric subdivision for polygons [29]. For an n-sided polygon, one first selects a pivoting

point $\mathbf{X}_P$ inside, and then partitions the polygon into n quadrilaterals by connecting a vertex, the middle points of the two edges incident on the vertex, and $\mathbf{X}_P$. The process is illustrated in Figure 5.1. If the pivot point $\mathbf{X}_P$ is the barycenter (i.e., the geometric centroid), this process reduces to the standard barycentric subdivision.



(a) (b) (c)

Figure 5.1: Examples of polygon subdivision. (a) triangle; (b) quadrilateral; (c) pentagon.

A domain Ω that originally contains N nodes is partitioned into N cells. Each cell is the union of all subdivision quadrilaterals that contain a common node, as illustrated in Figure 5.2(a). We denote by Ω_I the cell associated with the node I. Obviously, $\Omega_I \cap \Omega_J = \emptyset$ if $I \neq J$, and $\cup_{I=1}^{N} \Omega_I = \Omega$. If we denote the original polygon element with Ω_e and the set of the original polygon elements containing node I with $elem(I)$, we can define subcell Ω_{eI} by $\Omega_{eI} = \Omega_e \cap \Omega_I$ for $e \in elem(I)$. For subsequent use, we use subscript eI to denote subcells of node I for simplicity and introduce the

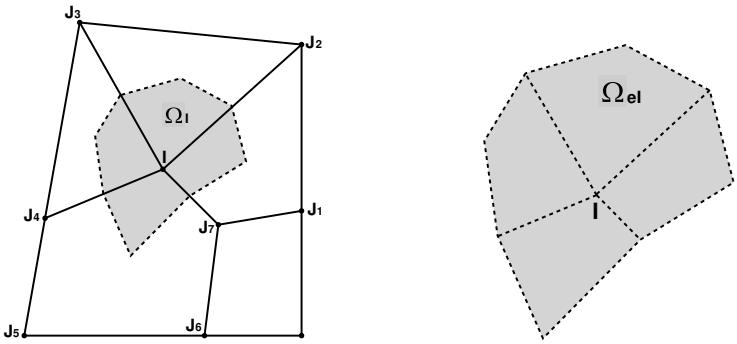
following notations:

A_e :	The area of polygon element Ω_e ;
A_{eI} :	The area of subcell Ω_{eI} ;
$subcell(I)$:	The set of the subcells that contain the node I;
$nodes(\Omega_{eI})$:	Nodes relating to subcell Ω_{eI}.

As shown in Figure 5.2(b), each nodal cell is the union of subcells Ω_{eI} and we have

$$A_I = \sum_{e \in subcell(I)} A_{eI}. \tag{5.1}$$

And the subcell areas A_{eI} are themselves affine covariant.



(a) (b)

Figure 5.2: The cell Ω_I and subcells Ω_{eI} associated with the node I. (a)Nodal cell Ω_I and its nodal set $node(\Omega_I)$ consisting of the nodes I and J_1 through J_7; (b)Nodal cell Ω_I is further divided into four subcells Ω_{eI}.

There are many possible way for selecting the pivot $\mathbf{X}_P$. For reasons that will

become evident later, we impose an invariance requirement on $\mathbf{X}_P$, such that any affine transform applying to the vertices applies also to $\mathbf{X}_P$. That is,

$$\mathbf{x}_J \mapsto \mathbf{F}\mathbf{X}_J + \mathbf{a} \quad \Longrightarrow \quad \mathbf{x}_P \mapsto \mathbf{F}\mathbf{X}_P + \mathbf{a} \quad \forall \text{ linear map } \mathbf{F} : \mathbb{R}^2 \mapsto \mathbb{R}^2 \text{ and } \mathbf{a} \in \mathbb{R}^2. \tag{5.2}$$

An easy example is the convex combination $\mathbf{X}_P = \sum_{J=1}^{n} c_J \mathbf{X}_J,\ c_J > 0,\ \sum_{J=1}^{n} c_J = 1$, where n is the number of nodes in an element. For implementation consideration, we further require that the formula of $\mathbf{X}_P$ be cyclically symmetric in $\mathbf{X}_J$. Examples of admissible choices include:

1. The geometric centroid (i.e., the barycenter)

$$\mathbf{X}_P = \frac{\int_{\Omega_e} \mathbf{X} da}{A_e}. \tag{5.3}$$

For a convex n-polygon with vertices $\mathbf{X}_1$, $\mathbf{X}_2 \cdots \mathbf{X}_n$, assuming the vertices are oriented counter-clockwise, the coordinates of the barycenter follows the formula

$$\mathbf{X}_P = \frac{1}{6A_e} \sum_{I=1}^{n} (\mathbf{X}_I + \mathbf{X}_{I+1})(X_I Y_{I+1} - X_{I+1} Y_I), \quad \text{with } \mathbf{X}_{n+1} = \mathbf{X}_1. \tag{5.4}$$

Here A_e is the element area, $A_e = \frac{1}{2} \sum_{I=1}^{n} (X_I Y_{I+1} - X_{I+1} Y_I)$. Note that both $(X_I Y_{I+1} - X_{I+1} Y_I)$ and A_e are area forms; the quotient $(X_I Y_{I+1} - X_{I+1} Y_I)/A_e$ is invariant under an affine transformation. Hence, the formula (5.4) is affinely invariant. It is also cyclicly symmetric.

2. The algebraic centroid

$$\mathbf{X}_P = \frac{1}{n}\sum_{J=1}^{n}\mathbf{X}_J. \tag{5.5}$$

Clearly, this point is affinely invariant and cyclically symmetric. For convex polygons, this point always lies inside the element. In a triangle, it coincides with the barycenter.

Admissible choices are not limited to the barycenter and the algebraic centroid. We will present an alternative scheme for quadrilaterals when discussing the explicit formulations of gradient interpolant.

When $\mathbf{X}_P$ satisfies the affine invariance, the area A_I has two invariant properties that are critical to our development. Specifically, if we write A_I as $A_I(\mathbf{X}_I, \mathbf{X}_{J_1}, \cdots, \mathbf{X}_{J_n})$ to indicate the dependence of A_I on the position of vertices in $node(\Omega_I)$, then A_I satisfies

1 (Translational invariance) :

$$A_I(\mathbf{X}_I + \mathbf{a}, \mathbf{X}_{J_1} + \mathbf{a}, \cdots, \mathbf{X}_{J_n} + \mathbf{a}) = A_I(\mathbf{X}_I, \mathbf{X}_{J_1}, \cdots, \mathbf{X}_{J_n}) \quad \forall \text{ vector } \mathbf{a} \in \mathbb{R}^2,$$

2 (Affine invariance) :

$$A_I(\mathbf{F}\mathbf{X}_I, \mathbf{F}\mathbf{X}_{J_1}, \cdots, \mathbf{F}\mathbf{X}_{J_n}) = det(\mathbf{F})A_I(\mathbf{X}_I, \mathbf{X}_{J_1}, \cdots, \mathbf{X}_{J_n})$$
$$\forall \text{ linear map } \mathbf{F} : \mathbb{R}^2 \mapsto \mathbb{R}^2, \ \det \mathbf{F} \neq 0. \tag{5.6}$$

The first condition states that the cell area is invariant under a rigid translation on the mesh. The second indicates that the area changes affinely under a uniform

stretching or shearing to the mesh. These conditions hold because the cell is the sum of subdivision quadrilaterals, which satisfy individually the above invariance properties when $\mathbf{X}_P$ observes (5.2).

5.2 Computation of gradient interpolants

In this chapter, the gradient interpolants are defined, uniformly for any combination of polygons, as

$$\mathbf{R}^*_{IJ} = \frac{1}{A_I}\frac{\partial A_I}{\partial \mathbf{X}_J}, \quad J \in nodes(\Omega_I). \tag{5.7}$$

Note that A_I depends smoothly on the positions of the vertices in $node(\Omega_I)$, and hence, the derivative is well-defined. Since $A_I = \sum_{e\in subcell(I)} A_{eI}$, Equation (5.7) can be written as

$$\mathbf{R}^*_{IJ} = \frac{1}{A_I}\sum_{e\in subcell(I)} \frac{\partial A_{eI}}{\partial \mathbf{X}_J}. \tag{5.8}$$

This is the canonical formula for the interpolant. The computation boils down to the derivative of the subcell area A_{eI} relative to the positions of vertices. The explicit form for $\frac{\partial A_{eI}}{\partial \mathbf{X}_J}$ depends on the type of polygon and the choice of $\mathbf{X}_P$. Some common cases are discussed below.

5.2.1 Triangle

Consider the triangle element in Figure 5.1(a), where $\mathbf{X}_P$ is the barycenter. Denoted by A_{eI} the area of the subcell Ω_{eI}. Obviously, A_{eI} equals to one-third of the parent triangle area. When the nodes I, J, K are oriented counter-clockwise, we

have

$$A_{eI} = \frac{1}{6}((X_J - X_I)(Y_K - Y_I) - (Y_J - Y_I)(X_K - X_I)). \tag{5.9}$$

It follows that

$$\begin{aligned}
\frac{\partial A_{eI}}{\partial \mathbf{X}_I} &= \frac{1}{6}(Y_J - Y_K)\mathbf{e}_x - \frac{1}{6}(X_J - X_K)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_J} &= \frac{1}{6}(Y_K - Y_I)\mathbf{e}_x - \frac{1}{6}(X_K - X_I)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_K} &= \frac{1}{6}(Y_I - Y_J)\mathbf{e}_x - \frac{1}{6}(X_I - X_J)\mathbf{e}_y.
\end{aligned} \tag{5.10}$$

Note that the finite element gradient in a linear triangular element is $(\nabla u)_e = \frac{1}{A_e}(\frac{\partial A_e}{\partial \mathbf{X}_I}u_I + \frac{\partial A_e}{\partial \mathbf{X}_J}u_J + \frac{\partial A_e}{\partial \mathbf{X}_K}u_K)$. In light of the relation $A_{eI} = \frac{1}{3}A_e$, the finite element gradient can also be written as

$$(\nabla u)_e = \frac{1}{A_{eI}}\left(\frac{\partial A_{eI}}{\partial \mathbf{X}_I}u_I + \frac{\partial A_{eI}}{\partial \mathbf{X}_J}u_J + \frac{\partial A_{eI}}{\partial \mathbf{X}_K}u_K\right). \tag{5.11}$$

With this result, the following remark is in order:

Remark 2. *Over a triangular mesh, the discrete gradient is related to the FE gradient through*

$$(\nabla^h u)_I = \frac{\sum_{e \in subcell(I)} A_{eI}(\nabla u)_e}{\sum_{e \in subcell(I)} A_{eI}}. \tag{5.12}$$

This is because

$$
\begin{aligned}
(\nabla^h u)_I &= \frac{1}{A_I}\left(\sum_{J\in nodes(\Omega_I)} \frac{\partial A_I}{\partial \mathbf{X}_J} u_J\right) = \frac{1}{A_I}\left(\sum_{J\in nodes(\Omega_I)}\sum_{e\in subcell(I)} \frac{\partial A_{eI}}{\partial \mathbf{X}_J} u_J\right) \\
&= \frac{1}{A_I}\left(\sum_{e\in subcell(I)}\sum_{J\in nodes(\Omega_e)} \frac{\partial A_{eI}}{\partial \mathbf{X}_J} u_J\right).
\end{aligned}
\tag{5.13}
$$

From (5.11)*, we see* $\sum_{J\in nodes(\Omega_e)} \frac{\partial A_{eI}}{\partial \mathbf{X}_J} u_J = A_{eI}(\nabla u)_e$*. The conclusion follows after recognizing* $A_I = \sum_{J\in subcell(I)} A_{eI}$*. The relation* (5.12) *indicates that the discrete gradient coincides with the nodal average strain introduced in [15, 36].*

5.2.2 Quadrilateral. Algebraic subdivision

The subcell area A_{eI} in the quadrilateral element as shown in Figure 5.1(b) follows the formula

$$
\begin{aligned}
A_{eI} =& \frac{1}{4}((X_J - X_I)(Y_P - Y_I) - (Y_J - Y_I)(X_P - X_I) \\
&+ (X_P - X_I)(Y_K - Y_I) - (Y_P - Y_I)(X_K - X_I)).
\end{aligned}
\tag{5.14}
$$

If $\mathbf{X}_P$ is the algebraic centroid, namely $\mathbf{X}_P = \frac{1}{4}(\mathbf{X}_I + \mathbf{X}_J + \mathbf{X}_K + \mathbf{X}_L)$, a straight forward calculation shows

$$
\begin{aligned}
A_{eI} =& \frac{1}{16}(3X_I(Y_J - Y_L) - X_J(3Y_I - Y_K - 2Y_L) \\
&- X_K(Y_J - Y_L) + X_L(3Y_I - 2Y_J - Y_K)).
\end{aligned}
\tag{5.15}
$$

It follows that

$$\begin{aligned}
\frac{\partial A_{eI}}{\partial \mathbf{X}_I} &= \frac{3}{16}(Y_J - Y_L)\mathbf{e}_x + \frac{3}{16}(X_L - X_J)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_J} &= \frac{1}{16}(-3Y_I + Y_K + 2Y_L)\mathbf{e}_x + \frac{1}{16}(3X_I - X_K - 2X_L)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_K} &= \frac{1}{16}(Y_L - Y_J)\mathbf{e}_x - \frac{1}{16}(X_L - X_J)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_L} &= \frac{1}{16}(3Y_I - 2Y_J + Y_K)\mathbf{e}_x - \frac{1}{16}(3X_I - 2X_J - X_L)\mathbf{e}_y.
\end{aligned} \tag{5.16}$$

5.2.3 Quadrilateral. Barycentric subdivision

Although the coordinates of the barycenter are rational functions, the subcell area turns out to be a simple polynomial. In fact, substituting (5.4) into (5.14) we find

$$A_{eI} = \frac{1}{12}[2X_I(Y_J - Y_L) - X_J(2Y_I - Y_K - Y_L) - X_K(Y_J - Y_L) + X_L(2Y_I - Y_J - Y_K)]. \tag{5.17}$$

Therefore,

$$\begin{aligned}
\frac{\partial A_{eI}}{\partial \mathbf{X}_I} &= \frac{1}{6}(Y_J - Y_L)\mathbf{e}_x + \frac{1}{6}(X_L - X_J)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_J} &= \frac{1}{12}(-2Y_I + Y_K + Y_L)\mathbf{e}_x + \frac{1}{12}(2X_I - X_K - X_L)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_K} &= \frac{1}{12}(Y_L - Y_J)\mathbf{e}_x - \frac{1}{12}(X_L - X_J)\mathbf{e}_y, \\
\frac{\partial A_{eI}}{\partial \mathbf{X}_L} &= \frac{1}{12}(2Y_I - Y_J - Y_K)\mathbf{e}_x - \frac{1}{12}(2X_I - X_J - X_K)\mathbf{e}_y.
\end{aligned} \tag{5.18}$$

5.2.4 Pentagon. Algebraic subdivision

For the pentagon in Figure 5.1(c), when $\mathbf{X}_P$ is the algebraic centroid, the area A_{eI} is

$$\begin{aligned} A_{eI} = \frac{1}{20}(&4X_I(Y_J - Y_M) - X_J(4Y_I - Y_K - Y_L - 2Y_M) \\ &- X_K(Y_J - Y_M) - X_L(Y_J - Y_M) + X_M(4Y_I - 2Y_J - Y_K - Y_L)). \end{aligned} \tag{5.19}$$

Therefore,

$$\begin{aligned} \frac{\partial A_{eI}}{\partial \mathbf{X}_I} &= \frac{1}{5}(Y_J - Y_M)\mathbf{e}_x + \frac{1}{5}(X_M - X_J)\mathbf{e}_y, \\ \frac{\partial A_{eI}}{\partial \mathbf{X}_J} &= \frac{1}{20}(-4Y_I + Y_K + Y_L + 2Y_M)\mathbf{e}_x + \frac{1}{20}(4X_I - X_K - Y_L - 2X_M)\mathbf{e}_y, \\ \frac{\partial A_{eI}}{\partial \mathbf{X}_K} &= \frac{1}{20}(Y_M - Y_J)\mathbf{e}_x - \frac{1}{20}(X_M - X_J)\mathbf{e}_y, \\ \frac{\partial A_{eI}}{\partial \mathbf{X}_L} &= \frac{1}{20}(Y_M - Y_J)\mathbf{e}_x - \frac{1}{20}(X_M - X_J)\mathbf{e}_y, \\ \frac{\partial A_{eI}}{\partial \mathbf{X}_M} &= \frac{1}{20}(4Y_I - 2Y_J - Y_K - Y_L)\mathbf{e}_x - \frac{1}{20}(4X_I - 2X_J - X_K - X_L)\mathbf{e}_y. \end{aligned} \tag{5.20}$$

Admissible algorithms are not limited to the barycentric subdivision and the algebraic subdivision. For example, for a quadrilateral element, a possible choice of $\mathbf{X}_p$ is the intersection of the two diagonals. This point can be shown to be affinely invariant and cyclically symmetric. The resulting algorithm has an interesting connection to a nodal average strain method, as we bring up in the remark below.

Remark 3. *If $\mathbf{X}_P$ is taken to be the intersection of the diagonals (Figure 5.3), the*

subcell area A_{eI} *is*

$$A_{eI} = \frac{1}{4}((Y_J - Y_I)(X_L - X_I) - (Y_L - Y_I)(X_J - X_I)). \tag{5.21}$$

It so happens that the subarea equals to the nodal value of the Jacobian J *of the standard 4-node geometric mapping, namely,*

$$A_{eI} = J_e(\mathbf{X}_I). \tag{5.22}$$

Therefore, for a mesh that consists of entirely quadrilaterals,

$$A_I = \sum_{e \in subcell(I)} J_e(\mathbf{X}_I). \tag{5.23}$$

It is straight forward to verify that the derivatives $\frac{\partial A_{eI}}{\partial \mathbf{X}_Q}$ $(Q = I, J, K, L)$ *are related to the physical derivatives of the shape functions* N_Q *by*

$$\frac{1}{A_{eI}} \frac{\partial A_{eI}}{\partial \mathbf{X}_Q} = \left. \frac{\partial N_Q}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}_I}, \quad Q = I, J, K, L. \tag{5.24}$$

In this case,

$$\begin{aligned} \sum_{J \in nodes(\Omega_I)} \frac{\partial A_I}{\partial \mathbf{X}_J} u_J &= \sum_{J \in nodes(\Omega_I)} \sum_{e \in subcell(I)} \frac{\partial A_{eI}}{\partial \mathbf{X}_J} u_J \\ &= \sum_{e \in subcell(I)} \sum_{J \in nodes(\Omega_e)} A_{eI} \left. \frac{\partial N_J}{\partial \mathbf{X}} \right|_{\mathbf{X}=\mathbf{X}_I} u_J = \sum_{e \in subcell(I)} A_{eI} (\nabla u(\mathbf{X}_I))_e. \end{aligned} \tag{5.25}$$

In light of this result and (5.22) *and* (5.23)*, the discrete gradient at node I can be written as*

$$(\nabla^h u)_I = \frac{\sum_{e\in subcell(I)} J_e(\mathbf{X}_I)(\nabla u(\mathbf{X}_I))_e}{\sum_{e\in subcell(I)} J_e(\mathbf{X}_I)}. \tag{5.26}$$

This expression recovers the the nodal average strain of NICE-Q4 element in [61] derived in a variational setting. It is worth noting that the derivative of A_{eI} with respect to the diagonal node $\mathbf{X}_K$ in Figure 5.3 is identically zero. Thus, the discrete gradient $(\nabla^h u)_I$ at the node I does not depend on the nodes diagonal to it. This leads to a smaller connectivity compared to the algebraic centroid and the barycentric algorithms, but the linear system may be less stable.

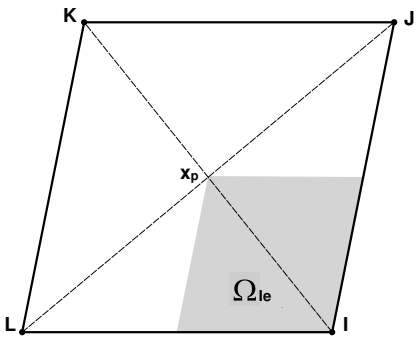


Figure 5.3: Alternative partition: $\mathbf{X}_P$ is the intersection of the diagonals.

5.2.5 Tetrahedron

Consider the tetrahedron element and select $\mathbf{X}_P$ as the barycenter. Denoted by V_{eI} the volume of the subcell Ω_{eI}. Obviously, V_{eI} equals to one-four of the parent

tetrahedron volume. When the vertex are nodes I, J, K, L, we have

$$\begin{aligned} V_{eI} &= \frac{1}{24} det(\mathbf{X}_J - \mathbf{X}_I, \mathbf{X}_K - \mathbf{X}_I, \mathbf{X}_L - \mathbf{X}_I) \\ &= \frac{1}{24} |(\mathbf{X}_J - \mathbf{X}_I) \cdot ((\mathbf{X}_K - \mathbf{X}_I) \times (\mathbf{X}_L - \mathbf{X}_I))|. \end{aligned} \tag{5.27}$$

Therefore,

$$\begin{aligned} \frac{\partial V_{eI}}{\partial \mathbf{X}_I} &= \frac{1}{24}(Y_K Z_J - Y_L Z_J - Y_J Z_K + Y_L Z_K + Y_J Z_L - Y_K Z_L)\mathbf{e}_x \\ &\quad + \frac{1}{24}(-X_K Z_J + X_L Z_J + X_J Z_K - X_L Z_K - X_J Z_L + X_K Z_L)\mathbf{e}_y \\ &\quad + \frac{1}{24}(X_K Y_J - X_L Y_J - X_J Y_K + X_L Y_K + X_J Y_L - X_K Y_L)\mathbf{e}_z \\ \frac{\partial V_{eI}}{\partial \mathbf{X}_J} &= \frac{1}{24}(-Y_K Z_I + Y_L Z_I + Y_I Z_K - Y_L Z_K - Y_I Z_L + Y_K Z_L)\mathbf{e}_x \\ &\quad + \frac{1}{24}(X_K Z_I - X_L Z_I - X_I Z_K + X_L Z_K + X_I Z_L - X_K Z_L)\mathbf{e}_y \\ &\quad + \frac{1}{24}(-X_K Y_I + X_L Y_I + X_I Y_K - X_L Y_K - X_I Y_L + X_K Y_L)\mathbf{e}_x \\ \frac{\partial V_{eI}}{\partial \mathbf{X}_K} &= \frac{1}{24}(Y_J Z_I - Y_L Z_I - Y_i Z_J + Y_L Z_J + Y_I Z_L - Y_J Z_L)\mathbf{e}_x \\ &\quad + \frac{1}{24}(-X_J Z_I + X_L Z_I + X_I Z_J - X_L Z_J - X_I Z_L + X_J Z_L)\mathbf{e}_y \\ &\quad + \frac{1}{24}(X_J Y_I - X_L Y_I - X_I Y_J + X_L Y_J + X_I Y_L - X_J Y_L)\mathbf{e}_z \\ \frac{\partial V_{eI}}{\partial \mathbf{X}_L} &= \frac{1}{24}(-Y_J Z_I + Y_K Z_I + Y_I Z_J - Y_K Z_J - Y_I Z_K + Y_J Z_K)\mathbf{e}_x \\ &\quad + \frac{1}{24}(X_J Z_I - X_K Z_I - X_I Z_J + X_K Z_J + X_I Z_K - X_J Z_K)\mathbf{e}_y \\ &\quad + \frac{1}{24}(-X_J Y_I + X_K Y_I + X_I Y_J - X_K Y_J - X_I Y_K + X_J Y_K)\mathbf{e}_z. \end{aligned} \tag{5.28}$$

5.3 Discrete Galerkin formulation

Over each cell, the strain is approximated by the node-wise discrete gradient, which, when applies to the vector field $\mathbf{u}$, gives

$$(\nabla^h \mathbf{u})_I = \sum_{J \in nodes(\Omega_I)} \mathbf{u}_J \otimes \mathbf{R}^*_{IJ}. \tag{5.29}$$

The discrete Galerkin formulation in the context of linear elasticity and its extension to finite elasticity had been described in Section 3.3 and 3.4. Therefore, the implementation details are omitted here.

5.4 Discrete gradient and linear consistency

We seek to approximate the gradient of a physical variable u at node I by a linear combination of the nodal values in the nodal set $node(\Omega_I)$, viz.

$$(\nabla^h u)_I = \sum_{J \in nodes(\Omega_I)} \mathbf{R}^*_{IJ} u_J. \tag{5.30}$$

We show that the consistency conditions (3.4) are satisfied by construction. Recall that the area A_I obeys the invariance properties (5.6). Consider an one-parameter family of vector $\mathbf{a}(\varepsilon)$ with $\mathbf{a}(0) = \mathbf{0}$. Taking the derivative of both sides of $(5.6)_1$ with respect to ε, evaluating the derivatives at $\varepsilon = 0$, and noticing that right-hand-side is independent of ε, we obtain

$$\left[\sum_{J \in nodes(\Omega_I)} \frac{\partial A_I}{\partial \mathbf{X}_J} \right] \cdot \dot{\mathbf{a}}(0) = 0. \tag{5.31}$$

where $\dot{\mathbf{a}} = \frac{\mathrm{d}\mathbf{a}}{\mathrm{d}\varepsilon}$. Since the equation holds for any $\dot{\mathbf{a}}(0)$, and the vector inside the square bracket is independent of $\dot{\mathbf{a}}(0)$, we conclude that

$$\sum_{J \in nodes(\Omega_I)} \frac{\partial A_I}{\partial \mathbf{X}_J} = \mathbf{0}, \tag{5.32}$$

which implies $(3.4)_1$. Similarly, consider an one-parameter family of non-singular linear map $\mathbf{F}(\varepsilon)$ with $\mathbf{F}(0) = \mathbf{I}$. Taking the derivative with respect to ε on both sides of $(5.6)_1$, noticing $\dot{\overline{\det \mathbf{F}}} = (\det \mathbf{F})\mathbf{F}^{-T} : \dot{\mathbf{F}}$, and evaluating the ensuing result at $\varepsilon = 0$, we get

$$\sum_{J \in nodes(\Omega_I)} \frac{\partial A_I}{\partial \mathbf{X}_J} \cdot \left[\dot{\mathbf{F}}(0)\mathbf{X}_J\right] = \left(det\mathbf{F}(0)\mathbf{F}^{-T}(0) : \dot{\mathbf{F}}(0)\right) A_I, \tag{5.33}$$

where (:) stands for the usual tensor contraction. Using the identity $\mathbf{u}\cdot\mathbf{A}\mathbf{v} = \mathbf{u}\otimes\mathbf{v} : \mathbf{A}$ for any vectors ($\mathbf{u}$, $\mathbf{v}$) and second order tensor $\mathbf{A}$, and noticing $\mathbf{F}(0) = \mathbf{I}$, we can write (5.33) as

$$\left[\sum_{J \in nodes(\Omega_I)} \frac{\partial A_I}{\partial \mathbf{X}_J} \otimes \mathbf{X}_J - A_I \mathbf{I}\right] : \dot{\mathbf{F}}(0) = 0. \tag{5.34}$$

By the same argument, the bracketed term is a zero tensor, and that proves $(3.4)_2$.

5.5 Patch test

In Section 3.6, we listed the necessary conditions to satisfy patch test. And in follows, we show the condition (3.41) is satisfied *by construction* in the present method. Invoking the relation (5.7), we find

$$\sum_{J \in nodes(\Omega_I)} A_J \mathbf{R}^*_{JI} = \frac{\partial\left(\sum_{J \in nodes(\Omega_I)} A_J\right)}{\partial \mathbf{X}_I}, \tag{5.35}$$

where $\mathbf{R}^*_{JI}$ is the node-I interplant associated with the cell J. The numerator $\sum_{J\in nodes(\Omega_J)} A_J$ constitutes the sum of the area A_I and its one-ring neighboring cells. Since the area outside the one-ring elements connecting to the node I is independent of $\mathbf{X}_I$, we have $\frac{\partial\left(\sum_{J\in nodes(\Omega_I)} A_J\right)}{\partial \mathbf{X}_I} = \frac{\partial\left(\sum_{e\in elem(I)} A_e\right)}{\partial \mathbf{X}_I}$, and hence the equation (5.35) becomes

$$\sum_{J\in nodes(\Omega_I)} A_J \mathbf{R}^*_{JI} = \frac{\partial\left(\sum_{e\in elem(I)} A_e\right)}{\partial \mathbf{X}_I} = \sum_{e\in elem(I)} \frac{\partial A_e}{\partial \mathbf{X}_I}. \tag{5.36}$$

The derivative $\frac{\partial A_e}{\partial \mathbf{X}_I}$ can be readily computed with the aid of the triangle area formula (5.9). With reference to the notations in Figure 5.4(a), we can write

$$\frac{\partial A_e}{\partial \mathbf{X}_I} = \mathbf{S}^{\perp}, \tag{5.37}$$

where $\mathbf{S}^{\perp}$ is normal vector on the line M_1M_2 whose length equals to the that of M_1M_2, or, $\mathbf{S}^{\perp} = \int_{M_1M_2} \mathbf{n} ds$ where $\mathbf{n}$ is the unit normal vector on M_1M_2 pointing to the node I. Note that this formula applies to any polygon element. Invoking the divergence theorem $\oint \mathbf{n} ds = \mathbf{0}$ and applying it to the triangle M_1IM_2, we can further conclude

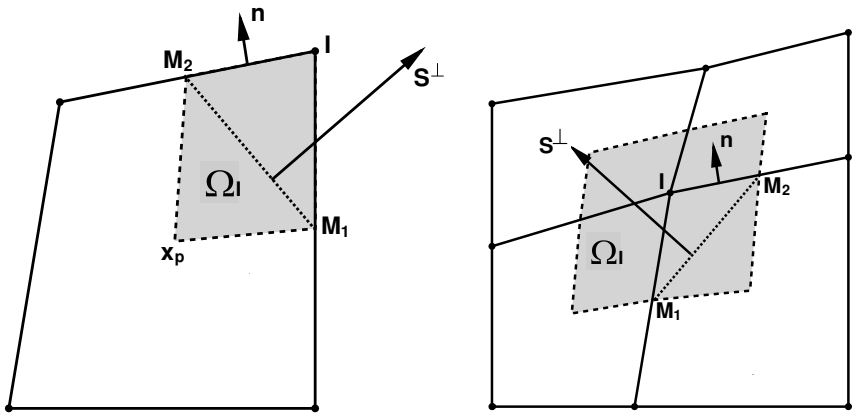
$$\frac{\partial A_e}{\partial \mathbf{X}_I} = \int_{M_1I} \mathbf{n} ds + \int_{IM_2} \mathbf{n} ds. \tag{5.38}$$

Therefore,

$$\sum_{e\in elem(I)} \frac{\partial A_e}{\partial \mathbf{X}_I} = \sum_{e\in elem(I)} \left(\int_{M_1I} \mathbf{n} ds + \int_{IM_2} \mathbf{n} ds \right)_e. \tag{5.39}$$

When I is an interior node (Figure 5.4(b)), the line integrals cancel in pair and eventually $\sum_{e\in elem(I)} \frac{\partial A_e}{\partial \mathbf{X}_I} = \mathbf{0}$. When I falls on the boundary (Figure 5.4(a)), $\sum_{e\in elem(I)} \frac{\partial A_e}{\partial \mathbf{X}_I} =$

$\int_{\partial\Omega_t \cap \partial\Omega_I} \mathbf{n} ds$. Hence, the integration condition (3.41) is identically satisfied.



(a) (b)

Figure 5.4: Illustration of the vector $\frac{\partial A_e}{\partial \mathbf{X}_I}$. (a) In an element or a boundary cell; (b) In an interior cell.

CHAPTER 6
NUMERICAL TESTS

In this chapter, we present the numerical results of some benchmark tests to validate the accuracy, convergence and capability of dealing with incompressibility constraint without severe locking. In examples, we test the behavior of discrete gradient method based on different domain partitions, which are Voronoi cell (VC.), Delaunay triangle (Tri.), quadrilateral (Quad.) and pentagon. Moreover, three formulations for quadrilaterals have implemented: the geometric subdivision, the algebraic subdivisions and the diagonal subdivision (NICE-Q4). Since the result of first two formulations remain close to each other in the considered examples as the quadrilateral are more or less parallelograms, only the algebraic subdivision as the representative for the two. It should be noted that in distorted meshes these two formulations are expected to yield different results.

6.1 Patch test

Consider the extension of a thin plate under uniform end load. The plate is modeled as a linear isotropic elastic material with Young's modulus $E = 500$ and Poisson's ratio $\nu = 0.3$. The left edge is constrained against the x-displacement at all nodes, and the y-displacement at the lower-left corner. A uniform traction of $\sigma_x = 10$ is applied at the right edge. Under the plane strain condition the displacements in the beam are

$$u = 0.018200x, \qquad v = -0.007800y$$

and $\sigma_x = 10$.

Here, we tested several meshes, including Voronoi cell, triangle mesh, a quadrilateral mesh, a combination of triangles and quadrilaterals, and a combination of quadrilaterals and pentagon. The displacement error is monitored by the error function

$$Error = \left(\frac{\sum_{I=1}^{N} (\mathbf{u}_I - \mathbf{u}_I^h) \cdot (\mathbf{u}_I - \mathbf{u}_I^h)}{\sum_{I=1}^{N} (\mathbf{u}_I \cdot \mathbf{u}_I)} \right)^{\frac{1}{2}}, \tag{6.1}$$

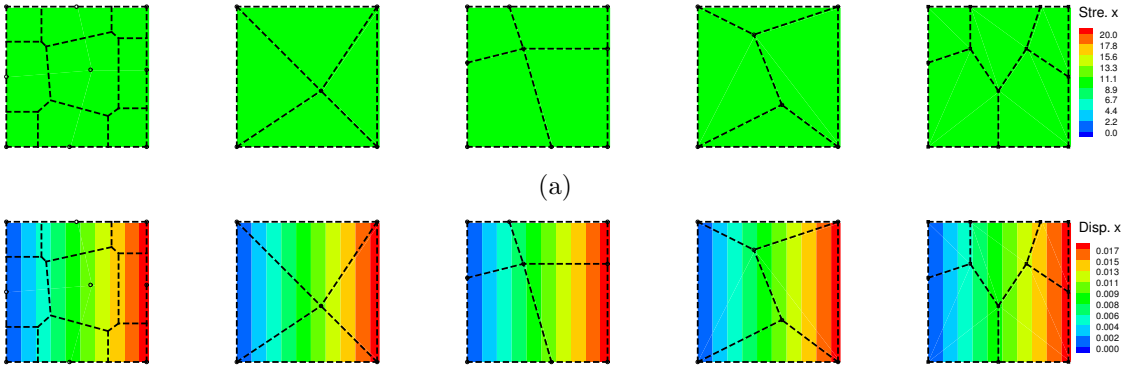
where N is the total number of nodes in the mesh. Table 6.1 lists the displacement errors and stress values in different meshes. The displacements and stresses are also graphically presented in Figure 6.1. Clearly, in all meshes the patch tests are satisfied to within numerical precision.

Tessellation	Err	Stress-x
Voronoi cell	2.103E-16	10.0
Triangle	4.255E-16	10.0
Quadrilateral	3.479E-16	10.0
Tri. and Quad.	2.642E-16	10.0
Quad and Pent.	2.263E-16	10.0

Table 6.1: Uniaxial tension test

6.2 Coupling with FEM

In the present formulation, the nodal coefficients u_I are the physical values of the unknown. This feature makes it straightforward to introduce essential boundary conditions in the formulation or to couple the method with the finite element method. Essential boundary conditions can be imposed in exactly the same manner as the finite element method. Here, we demonstrate the procedure of coupling with the



(a)

(b)

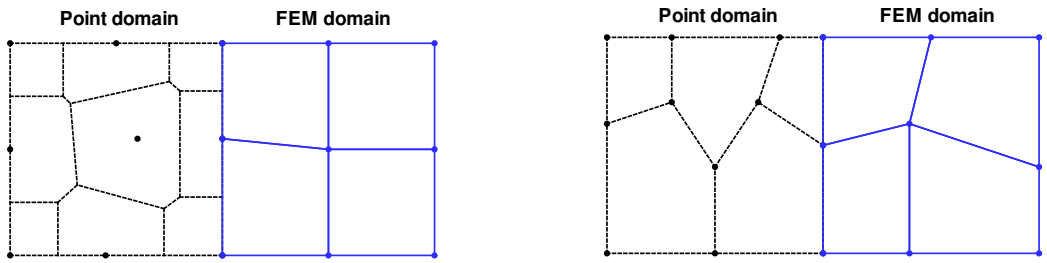
Figure 6.1: Patch test results. (a) Contours of σ_x; (b) Contours of axial displacement $\mathbf{u}_x$

finite element model using the example shown in Figure 6.2.

In this example, the domain (i.e., the plate) is divided into two subdomains; the left subdomain is represented by points while the right is further divided into elements. The two subdomains share a common interface on which the nodes from both sides coincide. Each subdomain is treated individually to yield their respective equations. The global equations are assembled from the subdomain contributions in exactly the same manner as the finite element assembly. Note that the nodal stiffness of the interface nodes consists of contributions from both the point domain and the element domain.

This coupling scheme is tested against a uniaxial tension problem. The plate is subject to a uniform traction of $\sigma_x = 10$ at the right end, and properly fixed at the left end to allow for a simple tension. The resulting axial displacement is shown in Figure 6.3(a) and (b). The predicted axial stress is homogeneous and exactly equals

the analytical value as shown in Figure 6.3(c) and (d). Evidently, the applied force is transmitted correctly from the finite element domain to the point domain.



(a) (b)

Figure 6.2: Schematics of the model of Coupling with finite element method. (a) Voronoi cell partition and FEM; (b) Polygon partition and FEM.

6.3 Beam subject to transverse tip load

In order to assess the convergence of the formulation, we consider the benchmark problem of bending of a cantilever beam under a shear load at the tip illustrated in Figure 6.4. The analytical solution of this problem is provided by Timoshenko and Goodier [99]. The beam considered here has a length $L = 10$ and a width $D = 2$. The material is assumed to be linear isotropic. The shear force is $P = 0.2$.

A convergence study is performed using regularly distributed nodes of 3×11, 5×21, 7×31 and 9×41 with triangle and quadrilateral partition. The plane strain condition is assumed, and both a compressible material ($E = 500$, $\nu = 0.3$) and a nearly incompressible material ($E = 500$, $\nu = 0.4999$) are considered. At each nodal density, the method is compared with the linear triangle finite element, quadrilateral

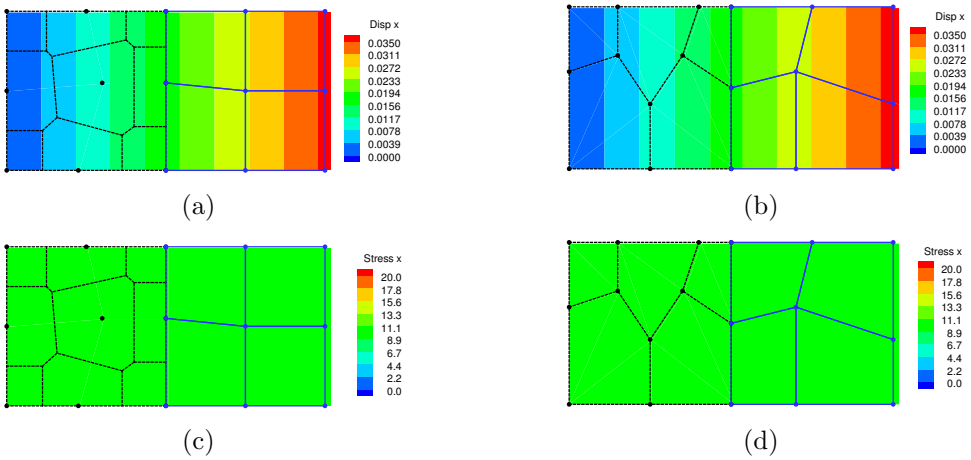


Figure 6.3: Results of Coupling with finite element method. (a) Contour of axial displacement; (b) Contour of axial displacement; (c) Contour of axial stress; (d) Contour of axial stress;

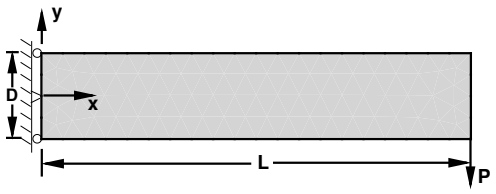


Figure 6.4: Schematics of cantilever beam bending.

element and the the 5-parameter enhanced strain element developed by Simo and Rifai [88]. The error is measured by the same function in Equation (6.1).

The convergence results are presented in Figure 6.5. For compressible material, the present method displays a comparable accuracy and convergence rate as the finite element. In the triangle mesh, the discrete method shows a slightly higher accuracy than the linear elements, but the quad mesh shows the other way. The enhanced element displayed a superior accuracy than all other methods but the rate of convergence appears lower. For nearly incompressible case, the discrete method retains the same level of accuracy and convergence rate as in the compressible case, without evidence of locking. It is not surprising that the enhanced element gives better accuracy in both cases because the element is designed to alleviate numerical locking.

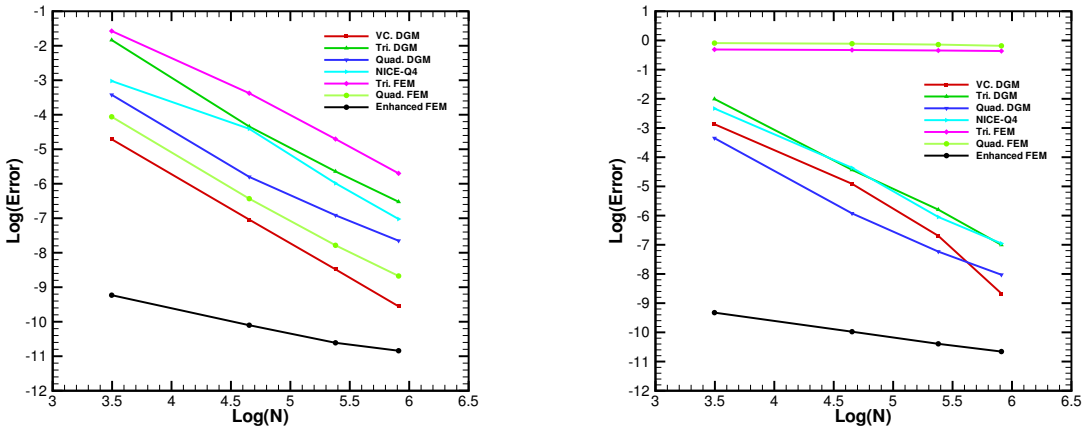


Figure 6.5: Displacement error versus number of nodes. (a) Compressible material; (b) Incompressible material.

6.4 Shear locking in thin beam

It is known that the displacement-based bilinear quadrilateral element suffers from shear locking in the thin-element limit. Remedies were developed mainly along two lines, the enhanced strain method [88] and selective integration with hourglass control [11]. Since the current method is essentially a strain average scheme, it is expect to alleviate shear locking in thin beams. To assess its behavior, we consider a series of progressively thin cantilevers with the slenderness ratio (the ratio of beam length versus height) varying from 2.5 to 640. The beams are subject a transverse load at the tip, and modeled by linear elasticity material $E = 500$, $\nu = 0.3$. One layer of elements is placed over the height.

The effect of locking is measured by the ratio

$$r := \left| \frac{v^h}{v} \right|. \tag{6.2}$$

where v and v^h are the analytical and numerical tip displacements. As locking intensifies, $v^h \to 0$ and hence r approaches zero. On the contrary, r approaches unity for locking-free response. Figure 6.6, shows that the present method exhibits locking-free behavior. The ratio is 1.001137 when slenderness is 10. For the largest slenderness ratio 640, the method still yields a very accurate tip displacement of $r = 0.991113$.

6.5 Mesh distortion

It is well known that the accuracy of finite element solution depends on mesh quality. If an element has a large aspect ratio, the element matrices may be rank

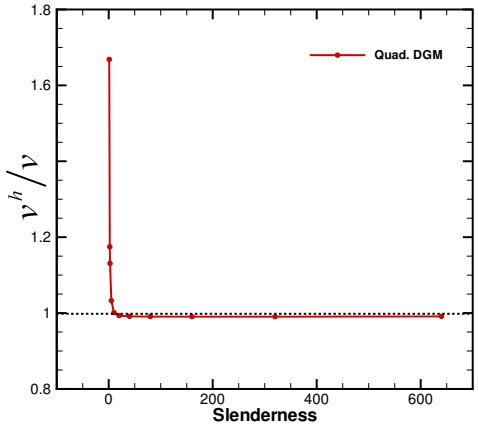


Figure 6.6: The ratio of numerical versus analytical tip deflections.

deficient and the resulting linear system may be poorly conditioned. The discrete method, on the contrary, is expected to be resilient to mesh distortion. This is because the discrete gradient consists of weighted average of the finite element gradients where the weights are directly related to element subareas (see Remarks 2 and 3). Typically, in a reasonably uniform mesh, a skew element would have a relatively small area, and thus, its contribution to the nodal gradient may be less significant.

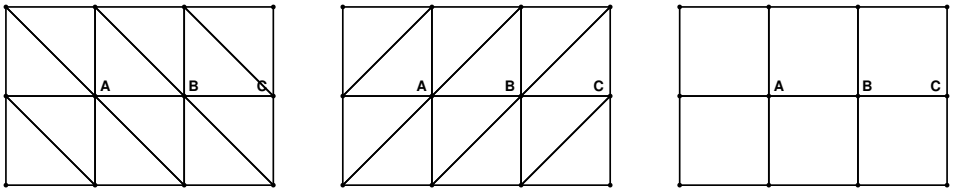
This speculation is verified numerically using the example below. Consider the (short) cantilever of dimension of 3×2, subjected to a 0.2 shear load at right end and fixed at the left end (Figure 6.7). The material is linear elastic, $E = 500$ and $\mu = 0.3$. The beam is discretized using two sets of meshes, one regular and the other distorted. Within each set, a triangle and a quad mesh are employed, as shown in Figures 6.7 and 6.8. Note that the triangle elements in Figure 6.8(b) are severely distorted. The problem is solved by FEM and the current method. The vertical displacements at nodes A, B and C are compared. These nodes are placed on the

middle and remain intact during mesh distortion. The analytical solution is included as a reference. The displacements are not expected to agree well with the analytical solution because of the coarse mesh used. Nevertheless, their variation across meshes provides a reasonable indicator for the mesh distortion effect. The problem is also analyzed using the discrete method with Delaunay triangles. It is known that the Delaunay mesh is the optimal triangle tessellation for an arbitrary set of nodes in the sense that the Delaunay triangles maximize the minimum angle among all possible triangulation.

The displacements at the said nodes are listed in Table 6.2 and Table 6.3. It is clear that, in overall, mesh distortion appears to have a small to moderate effect on the current method. The largest relative difference between the regular and the distorted triangle meshes is 11%, occurred at node B. In the quadrilateral meshes, the difference is below 7%. On the other hand, for FEM, the maximum displacement differences is 16% for quadrilateral element and 20% for triangle element, both occurred at node C. Evidently, compared to FEM the discrete formulation shows a lesser degree of sensitivity to mesh distortion. The deviations between the distorted triangle and the Delaunay mesh, which is considered to be a "good" triangle mesh, are also relatively small (within 10%).

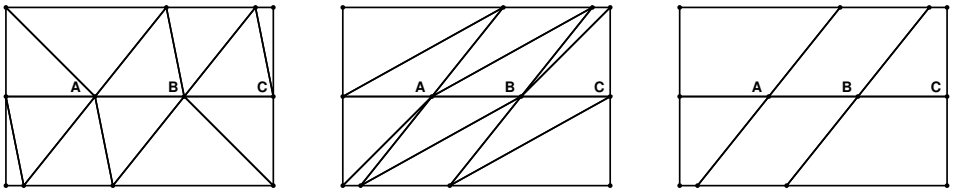
6.6 Stress concentration around circular hole

We analyze a problem that involves moderately large stress concentration. The system in Figure 6.9 corresponds to the central region of an infinite plate with a circular hole subject to a uniform tensile load in the y-direction. The stress solution



(a) (b) (c)

Figure 6.7: Regular meshes. (a) Delaunay triangle; (b) Triangle; (c) Quadrilateral.



(a) (b) (c)

Figure 6.8: Distorted meshes. (a) Delaunay triangle; (b) Triangle; (c) Quadrilateral.

Point	Analytic	Quad. FEM	Tri. FEM	Quad. DGM	Tri. DGM	Delaunay
A	1.508E-03	1.315E-03	1.005E-03	1.518E-03	1.841E-03	1.841E-03
B	4.108E-03	3.465E-03	2.599E-03	4.998E-03	5.699E-03	5.699E-03
C	7.254E-03	6.095E-03	4.463E-03	8.492E-03	1.013E-02	1.013E-02

Table 6.2: Nodal displacements from regular meshes

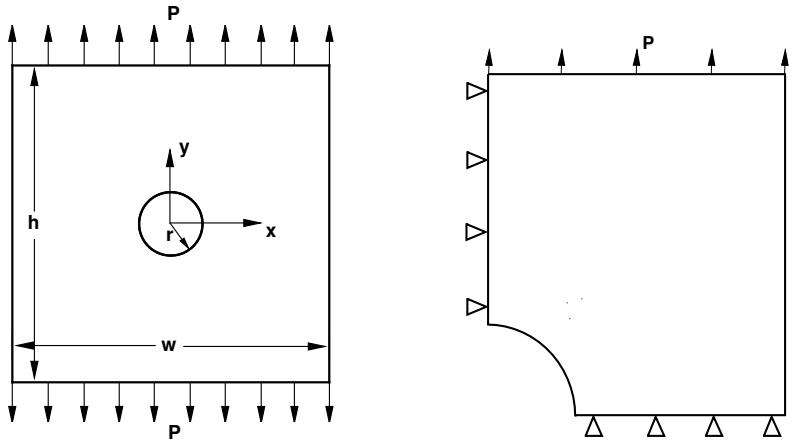
Point	Analytic	Quad. FEM	Tri. FEM	Quadrilateral	Triangle	Delaunay
A	1.508E-03	1.341E-03	1.059E-03	1.578E-03	1.906E-03	1.738E-03
B	4.108E-03	3.059E-03	2.293E-03	4.636E-03	5.022E-03	5.817E-03
C	7.254E-03	5.095E-03	3.576E-03	8.118E-03	9.172E-03	9.841E-03

Table 6.3: Nodal displacements from distorted meshes

for the infinite system is given, in cylindrical coordinates, as [99]

$$\begin{aligned}
\sigma_r &= \frac{p}{2}[(1-\frac{a^2}{r^2}) - (1+\frac{3a^4}{r^4}-\frac{4a^2}{r^2})\cos 2\theta] \\
\sigma_\theta &= \frac{p}{2}[(1+\frac{a^2}{r^2}) + (1+\frac{3a^4}{r^4})\cos 2\theta] \\
\tau_{r\theta} &= \frac{p}{2}(1-\frac{3a^4}{r^4}+\frac{2a^2}{r^2})\sin 2\theta,
\end{aligned} \tag{6.3}$$

where a is the radius of the hole. At $(r, \theta) = (a, 0)$, the tangential stress σ_θ (namely σ_y) has a concentration factor of 3.

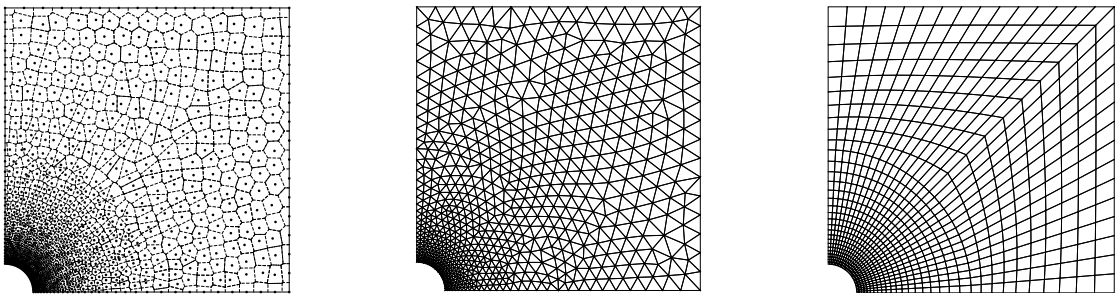


(a) (b)

Figure 6.9: Plate with a hole in the center subject to uniform tensile stress. (a) Central portion; (b) Simplified quarter model.

In the simulation, the central square region of 20×20 with the hole of radius $a = 1$ is modeled. Due to symmetry, only a quarter of the plate is considered, and symmetric boundary conditions are placed along the left and the bottom edges. Other parameters are set such that $p = 0.1$, $E = 500$, $\nu = 0.3$. The quarter domain

is discretized with Voronoi cell, triangular and quadrilateral mesh as shown in Figure 6.10, and is analyzed by the barycentric discrete method. The result contour of σ_y is plotted in Figure 6.11(a), where we see the stress concentration. The distributions of σ_x and σ_y along the bottom edge are depicted in Figure 6.11(b) and are compared to the analytical solution. It can be seen that the stress results compare well with the analytical solutions.

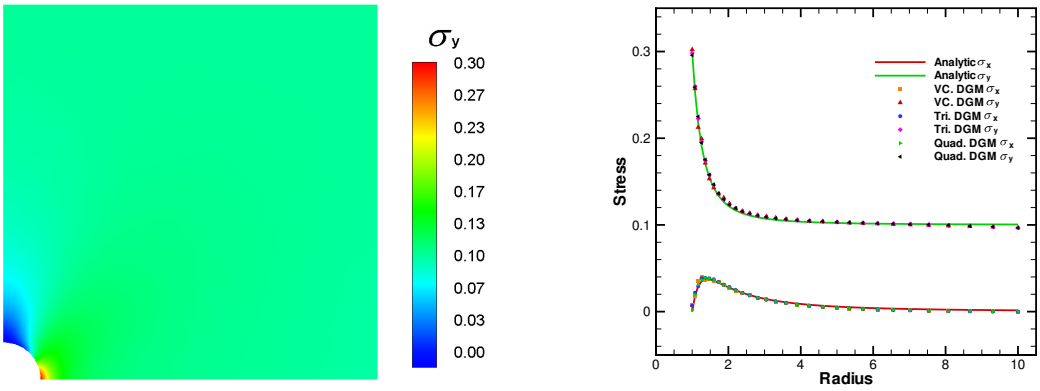


(a) (b) (c)

Figure 6.10: Partition of domain. (a) Voronoi diagram; (b) Triangular partition; (c) Quadrilateral partition.

6.7 Cook's membrane

The Cook's membrane is a tapered and swept panel of unit thickness. The left edge is clamped and the right edge is subjected to a distributed shear as illustrated in Figure 6.12. Frequently, this problem is used to assess the convergence properties of a numerical method near the incompressible limit under a mixture of shear and bending strains [1, 32]. In this test, finite strain deformation is assumed and the membrane is



(a) (b)

Figure 6.11: Stress calculated by DGM. (a)Contours of σ_y; (b)Distributions of σ_x and σ_y along the bottom edge.

modeled by a neo-Hookean material with the energy function

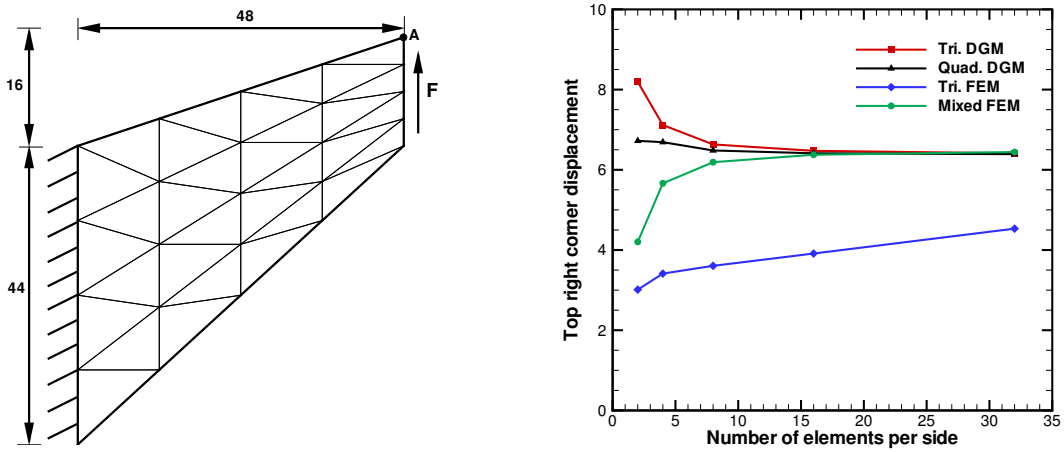
$$W = \frac{\mu}{2}(I_1 - 2\log J - 3) + \frac{\lambda}{2}(\log J)^2$$

where $\mu = 80.1938$ and bulk modulus $\lambda = 40.0942 \times 10^4$ is adopted. Also, a vertical load $F = 100$ is employed. The near incompressibility condition is introduced using a λ/μ ratio on the order of 10^4. The plane strain condition is assumed.

A mesh of four elements per side is depicted in Figure 6.12(a). The mesh is then progressively refined in order to assess the convergence. For comparison, the vertical displacement at the upper right corner of the panel (point A in Figure 6.12(a)) is reported. The results are shown in Figure 6.12(b), where the displacements are plotted against the number of elements per side. Solutions obtained from the standard three-node triangular element and finite strain mixed formulation [89] are

also included.

The discrete gradient method in overall produces accurate results. With quadrilateral meshes the solutions are reasonably good even at the coarse mesh end ($< 8 \times 8$). As the mesh refines the solutions from both triangle and quadrilateral meshes approach that of the mixed formulation, although from above (i.e., the present method over predicts the displacements). In contrary, the linear triangular element displays a locking behavior across all meshes.



(a) (b)

Figure 6.12: Cook's membrane problem. (a) Schematics of the system; (b) Convergence under mesh refinement.

6.8 Radial expansion of a cylindrical tube

This example assesses the performance of the discrete method in dealing with finite strain incompressible materials. The tube considered has an undeformed inner radius $R_i = 1$ and an outer radius of $R_0 = 2$. The tube is made of an incompressible

hyperelastic material described by the energy function

$$W = \frac{\mu_1}{2}(I_1 - 3) + \frac{\mu_2}{4}(I_1 - 3)^2,$$

where $I_1 = \mathrm{tr}\,(\mathbf{F}^T\mathbf{F})$. In the numerical computation, a penalty energy $\frac{\kappa}{2}(\log(J))^2$ with a large value κ is augmented to the energy function to introduce nearly incompressibility. The following numerical values are used:

$$\mu_1 = 100, \qquad \mu_2 = 5, \qquad \kappa = 1000000.$$

The analytical solution of this problem can be obtained through a procedure described by Green and Zerna [52]. Briefly, the incompressibility condition alone determines the radial motion to within an integration constant, which is in turn solved from an integral form of the radial equilibrium equation. Once the motion is determined, the stress in the wall can be found.

The system is modeled using the proposed discrete method. Due to symmetry, only a quarter of the domain is considered. The quarter domain is discretized into 9×25 non-uniform nodes, as shown in Figure 6.15(a). Three types for partitions are employed: Voronoi cell, triangular and quadrilateral mesh. A uniform finite element mesh of the same nodal density is introduced for comparison. The mixed element developed by Simo *et al* [88] is employed in the finite element simulation. Figure 6.8 presents the radial displacement of the inner wall as a function of the internal pressure. The discrete result clearly agrees well with both the analytical and the finite

element solutions. The displacements of the bottom nodes at the maximum pressure of $P = 150$ are plotted in Figure 6.14. Again, the results are in good agreement with the analytical and the finite element solutions. The contour of nodal radial displacements and hoop stress are plotted in Figure 6.15. However, nodal radial displacements distribution is not perfectly axis-symmetric. This is understandable because the nodes are not axisymmetrically placed.

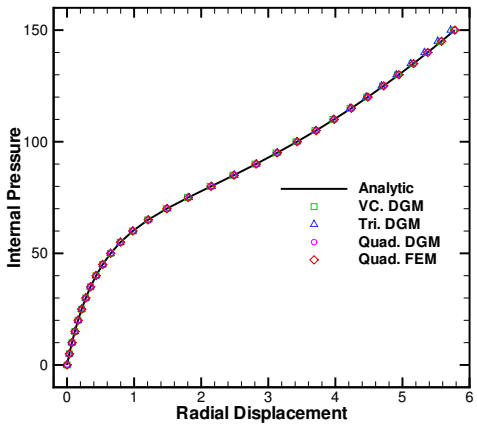


Figure 6.13: The inner wall radial displacement versus applied pressure.

6.9 Twist of three-dimensional beam

Although we developed and demonstrated discrete gradient method and testing examples in two-dimension space, all of the formulas can be easily extended three-dimension application. In this section, we use an example of twist of beam to show the capability of our method in dealing with three dimensional large deformation.

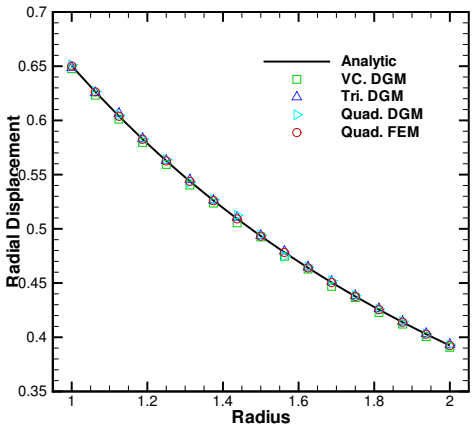
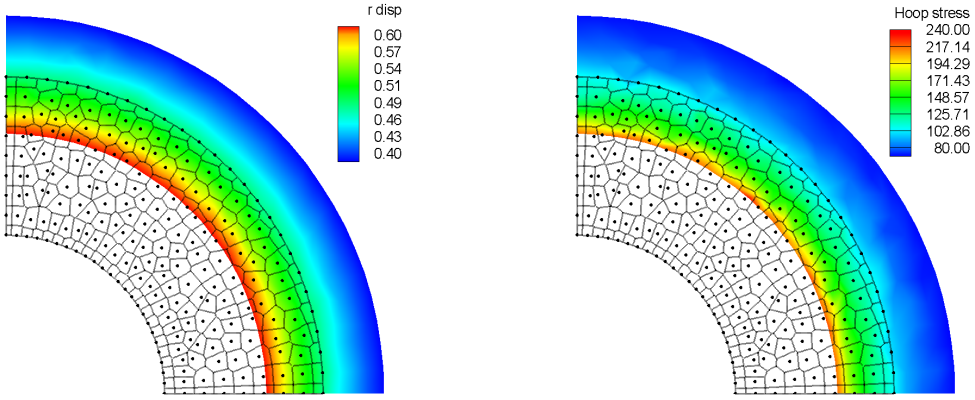


Figure 6.14: Radial displacement along the wall thickness.



(a) (b)

Figure 6.15: Radial expansion of a cylinder under internal pressure. (a) Radial displacement; (b) Hoop stress.

The beam is modeled as a hyperelastic material described by the energy function

$$W = \frac{100}{2}(I_1 - 3) + \frac{5}{4}(I_1 - 3)^2.$$

One of its ends is fixed. Moreover, displacements control is added on the other end to simulate the twist. The twisted angle of this beam is gradually increased to 720°. Figure 6.16 shows the whole process. Form it, we can see that the beam model lost stability only after the twisted angle is larger than 630°.

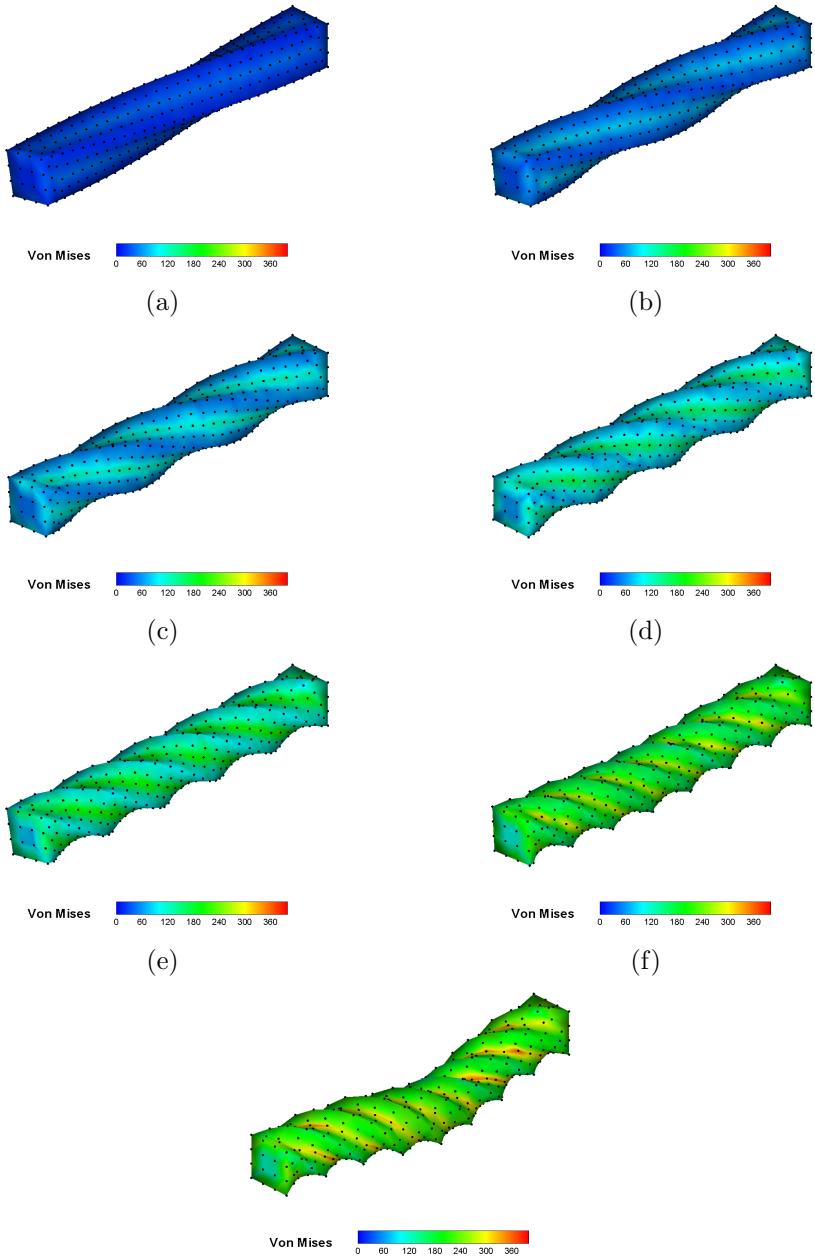


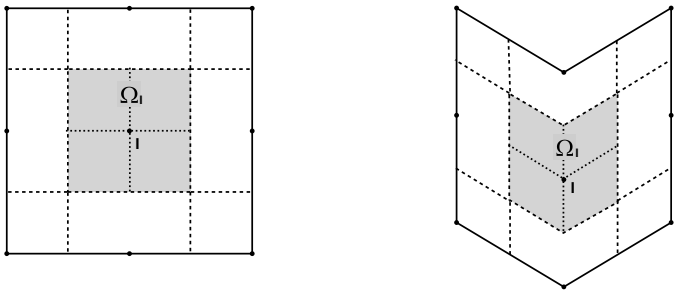
Figure 6.16: Contour of Von Mises stress for twisted beam. (a) Twisted angle is 90°; (b) Twisted angle is 180°; (d) Twisted angle is 270°; (e) Twisted angle is 360°; (f) Twisted angle is 450°; (g) Twisted angle is 540°; (h) Twisted angle is 630°; (i) Twisted angle is 720°.

CHAPTER 7
STABILIZATION OF DISCRETE GRADIENT METHOD

7.1 Introduction

Similar to the nodal integration methods and other particle methods, the discrete gradient method is prone to have spurious singular modes arising from either improper representation of gradient or under-integration of the weak form. A simple example is the saw-teeth mode. As shown in Figure 7.1(a), the domain is divided into nine barycentric cells or nodal cells. And the shade region Ω_I is the nodal cell domain assigned to the centric node I. This nodal cell domain can be further divided into four equal subcells by dotted lines. Under the saw-teeth deformation in Figure 7.1(b), the left and right half undergoes pure shear differing by a sign. Therefore, the average strain in the center nodal cell will be zero although the shear strain in each subcell is not. Only the boundary cells contribute to the strain energy and hereby, in the limit of mesh refinement the total energy approaches zero, leading to spurious zero energy mode. Unlike the hourglass mode that appears in reduced integration quadrilateral elements, which is independent of element shape, here the local zero energy mode could be geometry and connectivity dependent and thus, some spurious modes can be prevented by perturbing nodal position. Nevertheless, they cannot be exhaustively suppressed without any additional skill.

Some methods have been developed to improve the stability of nodal integration method. To overcome the instability resulted from under integration of weak from in element-free Galerkin method, Beissel and Belytschko [6] added a square of



(a) (b)

Figure 7.1: Example of deformation with zero average strain. (a)Original configuration. Shade region in the nodal cell domain assigned to centric node I, and dotted lines divide the region into four equal subcells. (b)Deformed configuration.

the residual in the governing equation, which includes the second derivatives of the displacements, to the potential energy functional as a stabilization. Later, Bonet and Kulasegaram [14] introduced a least-square stabilization in variational potential with a integration correction to avoid high order derivatives of shape function. These methods address the singularities arising from rank deficiency. Chen [25, 26] and Yoo [103] proposed a stabilized conforming nodal integration with a strain smoothing stabilization. Bonet [15] employed a modified nodal deformation gradient, which relies on modified shape functions, to remove unrealistic modes in linear tetrahedral element of average strain. The aspects of stability, consistency and explicit time integration in both element-based and meshfree-based nodal integration methods are discussed in a recent publication [84]. However, we will not use the two methods because we do not want to include other algorithms to calculate deformation gradient and make our method more complex. Puso *et al.* [85] proposed a stabilized strain energy for nodally integrated tetrahedral element [36] by adding a term of element strain. This

method addressed the issue of cancellation of opposite strain, and hence the idea is directly applicable to the current case. Following this work, we proposed a stabilized discrete gradient method by penalize the difference between nodal average strain and subcell strain. In our formulation, the stabilized scheme relies on discrete gradient operators but not fall back on the element interpolation. This preserves simplicity and discrete nature of our method. On the other hand, we will focus our work on the discrete gradient method founded on polygon mesh but not Voronoi diagram for the simplicity.

7.2 Subcell gradient

In this section, we introduce the discrete gradient in subcells, which are defined in Chapter 5. Let us define

$$\mathbf{R}^*_{eIJ} = \frac{1}{A_{eI}} \frac{\partial A_{eI}}{\partial \mathbf{X}_J}, \quad J \in nodes(\Omega_{eI}). \tag{7.1}$$

Obviously, the following *subcell gradient*

$$(\nabla^h u)_{eI} = \sum_{J \in nodes(\Omega_{eI})} \mathbf{R}^*_{eIJ} u_J, \tag{7.2}$$

is also linear consistent. In fact, the nodal gradient interpolant is related to the subcell ones by the following relation:

$$\mathbf{R}^*_{IJ} = \frac{1}{A_I} \sum_{e \in subcell(I)} A_{eI} \mathbf{R}^*_{eIJ}. \tag{7.3}$$

It follows that the nodal gradient (3.2) is a weighted average of the subcell gradients, viz,

$$(\nabla^h u)_I = \frac{1}{A_I} \sum_{e \in subcell(I)} A_{eI} (\nabla^h u)_{eI}. \tag{7.4}$$

This relation plays a key rule in the ensuing development.

7.3 Stabilized discrete Galerkin formulation

Puso *et al.* [85] proposed a stabilized strain energy for nodally integrated tetrahedral element [36]. The instability also arises from the cancellation of nodal average strains of opposite sign in saw-teeth type mode. Puso suggested penalizing the difference between the nodal average strain and the element strains by adding to the energy as quadratic term:

$$\int_\Omega \boldsymbol{\varepsilon} : \mathbb{D} \boldsymbol{\varepsilon} da \;\rightarrow\; \sum_I^N V_I \boldsymbol{\varepsilon}_I : \mathbb{D} \boldsymbol{\varepsilon}_I + \sum_I^N \sum_{e \in elem(I)} \alpha \frac{V_e}{4} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_e) : \tilde{\mathbb{D}} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_e). \tag{7.5}$$

Here, α is a stabilization parameter, $\tilde{\mathbb{D}}$ is a positive-definite material tensor which could be different from $\mathbb{D}$, V_e is the volume of tetrahedral finite element and $\boldsymbol{\varepsilon}_e$ is element strain. Note that the right-hand-side of (7.5) can be further simplified into

$$\sum_I^N V_I (\boldsymbol{\varepsilon}_I : (\mathbb{D} - \alpha \tilde{\mathbb{D}}) \boldsymbol{\varepsilon}_I) + \sum_I^N \sum_{e \in elem(I)} \alpha \frac{V_e}{4} \boldsymbol{\varepsilon}_e : \tilde{\mathbb{D}} \boldsymbol{\varepsilon}_e. \tag{7.6}$$

Puso's idea can be readily applied to the discrete formulation. Here, we do not have element strains but a natural substitution that is the subcell strain (7.2).

We construct a modified energy form

$$\int_\Omega \boldsymbol{\varepsilon} : \mathbb{D}\boldsymbol{\varepsilon} da \;\rightarrow\; \sum_I^N A_I \boldsymbol{\varepsilon}_I : \mathbb{D}\boldsymbol{\varepsilon}_I + \sum_I^N \sum_{e \in subcell(I)} \alpha A_{eI} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_{eI}) : \tilde{\mathbb{D}} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_{eI}). \quad (7.7)$$

where $\boldsymbol{\varepsilon}_{eI}$ is the subcell strain and $\boldsymbol{\varepsilon}_{eI} = (\nabla_s \mathbf{u})_{eI}$. The added term penalizes the difference between nodal average strain and subcell strain to avoid the instability caused by cancellation. Invoking the relation (7.4), a straightforward computation shows

$$\begin{aligned} &\sum_I^N A_I \boldsymbol{\varepsilon}_I : \mathbb{D}\boldsymbol{\varepsilon}_I + \sum_I^N \sum_{e \in subcell(I)} \alpha A_{eI} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_{eI}) : \tilde{\mathbb{D}} (\boldsymbol{\varepsilon}_I - \boldsymbol{\varepsilon}_{eI}) \\ =&\sum_I^N A_I \boldsymbol{\varepsilon}_I : (\mathbb{D} - \alpha \tilde{\mathbb{D}}) \boldsymbol{\varepsilon}_I + \sum_I^N \sum_{e \in subcell(I)} \alpha A_{eI} \boldsymbol{\varepsilon}_{eI} : \tilde{\mathbb{D}} \boldsymbol{\varepsilon}_{eI}. \end{aligned} \quad (7.8)$$

Following the previous procedure to compute the subcell strain in discrete kinematics, we have

$$\boldsymbol{\varepsilon}^h_{eI} = (\nabla^h_s \mathbf{u})_{eI} = \frac{1}{2} \sum_{J \in nodes(\Omega_{eI})} (\mathbf{u}_J \otimes \mathbf{R}^*_{eIJ} + \mathbf{R}^*_{eIJ} \otimes \mathbf{u}_J). \quad (7.9)$$

In vector form,

$$[\boldsymbol{\varepsilon}^h_{eI}] = \sum_{J \in nodes(\Omega_{eI})} \mathbf{B}_{eIJ} \mathbf{u}_J, \quad \text{where} \quad \mathbf{B}_{eIJ} = \begin{bmatrix} R^{*1}_{eIJ} & 0 \\ 0 & R^{*2}_{eIJ} \\ R^{*2}_{eIJ} & R^{*1}_{eIJ} \end{bmatrix}. \quad (7.10)$$

The stabilized cell-level stiffness matrix is

$$\mathbf{K}_I^{stable} = A_I \mathbf{B}_{IK}^T (\mathbf{D} - \alpha \tilde{\mathbf{D}}) \mathbf{B}_{IL} + \sum_{e \in subcell(I)} \alpha A_{eI} \mathbf{B}_{eIL}^T \tilde{\mathbf{D}} \mathbf{B}_{eIK} \quad K, L \in nodes(\Omega_I), \tag{7.11}$$

where $\mathbf{D}$ is the matrix form of elasticity tensor $\mathbb{D}$. Moreover, the second term on the right hand side in above equation is the contribution to stiffness matrix from subcells and can be notated by $\alpha \mathbf{K}_e$. The treatment of external force and inertia force remain the same.

Remark 4. *1. If we let $\tilde{\mathbb{D}} = \mathbb{D}$, from Equation* (3.11) *and* (7.11)*, we can see that $\mathbf{K}_I^{stable} = (1 - \alpha)\mathbf{K}_I + \alpha \mathbf{K}_e$. It is a linear combination of stiffness from nodal cell and subcells.*

2. As $\tilde{\mathbb{D}} = \mathbb{D}$ and $\alpha = 1$, the stiffness contributed by nodal cell is cancelled out, so that $\mathbf{K}_I^{stable} = \mathbf{K}_e$. This formulation is very similar to the stiffness matrix defined in the smoothed finite element method (SFEM) [65, 66]. And Liu et al had shown that a quadrilateral finite element divided into four smoothing subcells can suppress zero energy modes and gives stable results. The energy in (7.7) *reduces to $\sum_I^N \sum_{e \in subcell(I)} A_{eI} \boldsymbol{\varepsilon}_{eI} : \mathbb{D} \boldsymbol{\varepsilon}_{eI}$. This energy is always nonzero for any displacement field except for rigid body motions [27].*

3. If $\tilde{\mathbb{D}} = \mathbb{D}$, $0 < \alpha \leq 1$ and material is incompressible, the subcell stiffness has severe volume locking same as the behavior of low order finite element. This will lead the $\mathbf{K}_I^{stable}$ to have the same problem even with very small α.

4. *A possible extension of the stabilization to finite strain elasticity is to replace the cell energy $A_I W(\mathbf{F}_I)$ with the convex combination*

$$A_I W(\mathbf{F}_I) \rightarrow (1-\alpha) A_I W(\mathbf{F}_I) + \alpha \sum_e A_{Ie} W(\mathbf{F}_e) \tag{7.12}$$

where $\mathbf{F}_I$ and $\mathbf{F}_e$ are cell deformation gradient and subcell deformation gradient. If $\alpha = 1$, the method leads to a smoothed element stiffness matrix.

5. *Puso et al suggest a appropriate choice for α and $\tilde{\mathbb{D}}$. For isotropic elastic properties such that the effective lamé parameters λ and μ, the effective stabilization parameter and material modulus can be chosen as*

$$\alpha = 0.05, \quad \hat{\mu} = \mu \quad and \quad \hat{\lambda} = min(\lambda, 25\hat{\mu}).$$

7.4 A one-dimensional example

The following one-dimensional example may provide some insight into the stabilization algorithm. Consider the axial vibration of an elastic bar of unit cross section area. The governing equation is

$$c^2 \frac{\partial^2 u}{\partial x^2} = \ddot{u}, \tag{7.13}$$

where $c = \sqrt{\frac{E}{\rho}}$ is the wave speed, E and ρ are young's modulus and material density, respectively. The nodes are uniformed spaced over the length L as shown in Figure 7.2. The length that the nodal cell of x_I occupies is $L_I = h = \frac{x_{I+1} - x_{I-1}}{2}$. The gradient

interpolant formula, which applies to other dimensions, provides that A_I in Equation (5.7) is the corresponding Lebesgue measure (length in 1D, area in 2D, and volume 1n 3D). For the present case, at the node I, we have

$$R^*_{IJ} = \frac{1}{L_I}\frac{\partial L_I}{\partial x_J}. \tag{7.14}$$

It follows that

$$R^*_{I\,I-1} = -\frac{1}{2h}, \quad R^*_{I\,I} = 0, \quad R^*_{I\,I+1} = \frac{1}{2h}, \tag{7.15}$$

and

$$u'_I \approx R^*_{I\,I-1}u_{I-1} + R^*_{I\,I+1}u_{I+1} = \frac{u_{I+1} - u_{I-1}}{2h}. \tag{7.16}$$

Without stabilizer, the strain energy of the cell I is

$$E_I = \frac{Eh}{2}(u'_I)^2 = \frac{E}{8h}(u_{I+1} - u_{I-1})^2, \tag{7.17}$$

which notably doesn't depend on u_I. The equation of motion for the node I is

$$(\frac{c}{2h})^2(u_{I-2} - 2u_I + u_{I+2}) = \ddot{u}_I. \tag{7.18}$$

Assuming a solution which in the continuum limit has the form

$$u = \sin(kx - \omega t), \tag{7.19}$$

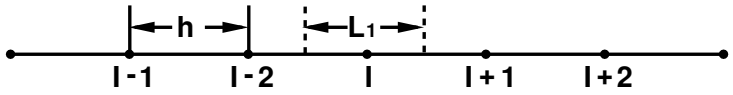


Figure 7.2: Nodes are uniformly spaced over one-dimensional bar.

where $k = \frac{n\pi}{L}$ and $n = 1, 2, 3, \cdots$. In the discrete setting, the nodal displacement is $u_{I+i} = \sin(kih + kx_I - \omega t)$, $i = 0, \pm 1, \pm 2, \cdots$, and $\ddot{u}_I = -\omega^2 \sin(kx_I - \omega t)$. Substituting them into (7.18), invoking the identified $\sin(kih + kx_I - \omega t) = \sin(kih)\cos(kx_I - \omega t) + \cos(kih)\sin(kx_I - \omega t)$, and cancelling a common factor $\sin(kx_I - \omega t)$ from both sides, we obtain

$$\omega^2 = (\frac{c}{h})^2(2 - 2\cos 2kh). \tag{7.20}$$

Solving for ω gives the dispersion equation

$$\frac{\omega}{kc} = \frac{1}{kh}\sqrt{\frac{1}{2}(1 - \cos 2kh)}. \tag{7.21}$$

Let us now consider the basic cut-off wavelength $kh = \pi$. Substituting $kh = \pi$ into the above formula, we find $\omega = 0$. Consequently, the nodal displacements are $[\cdots, -1, 1, -1, 1, \cdots]$ after shifting the wave phase with $\frac{\pi}{2}$. This solution translates to the saw-teeth form as show in Figure 7.3 in the discrete setting. Evidently, the saw-teeth wave is a spurious stationary wave. The source of instability is rank-deficient instability in the stiffness matrix; one can readily see that the saw-teeth mode is a zero-energy mode of the stiffness matrix. Although the source can be readily identified from the stiffness matrix, the spectrum analysis above allows us to investigate the

effect of stabilization analytically.

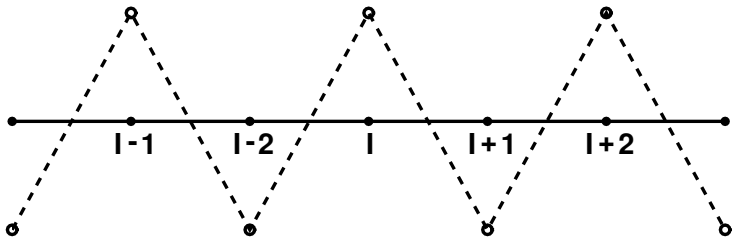


Figure 7.3: A spurious stationary wave in saw-teeth form.

7.4.1 Stabilization

Following the 2D treatment, the augmented energy function should have the form

$$E_I = \frac{Eh}{2}(u'_I)^2 + \frac{\alpha Eh}{2}\left[(u'_I - u'_{I,1})^2 + (u'_I - u'_{I,-1})^2\right] \tag{7.22}$$

where $u'_{I,1} = \frac{u_{I+1}-u_I}{h}$ and $u'_{I,-1} = \frac{u_I-u_{I-1}}{h}$ are subcell gradients which coincide with the finite element gradients in the elements incident on node I. The energy reduces to

$$E_I = \frac{(1-\alpha)E}{8h}(u_{I+1} - u_{I-1})^2 + \frac{\alpha Eh}{4}\left[\left(\frac{u_{I+1} - u_I}{h}\right)^2 + \left(\frac{u_I - u_{I-1}}{h}\right)^2\right]. \tag{7.23}$$

The equation of motion for the node I is

$$(1-\alpha)(\frac{c}{2h})^2(u_{I-2} - 2u_I + u_{I+2}) + \alpha\frac{c}{h}(u_{I-1} - 2u_I + u_{I+1}) = \ddot{u}_I. \tag{7.24}$$

Following the same procedure, the dispersion equation is found to be

$$\frac{\omega}{kc} = \frac{1}{kh}\sqrt{\frac{1-\alpha}{2}(1-\cos 2kh) + 2\alpha(1-\cos kh)}. \tag{7.25}$$

Some conclusions can be readily drawn from the dispersion equation:

1. The natural frequency at the cut-off wavelength is no longer zero.

2. There is no other zero-energy mode. If one considers a bar fixed at least at one end, all natural frequencies are positive.

Figure 7.4 presents the dispersion relation (7.25) for kh varying from $10^{-6}\pi$ to π under various values of $\alpha = 0, 0.005, 0.05, 0.5, 1$. We can see that $\frac{\omega}{kc}$ has the limitation of 1 as kh approaches to zero. In addition, $\frac{\omega}{kc}$ will get bigger as α increase. It means that the stabilized scheme results in a correct solution in the continuum limit. Moreover, the stabilization parameter can increase the system stiffness.

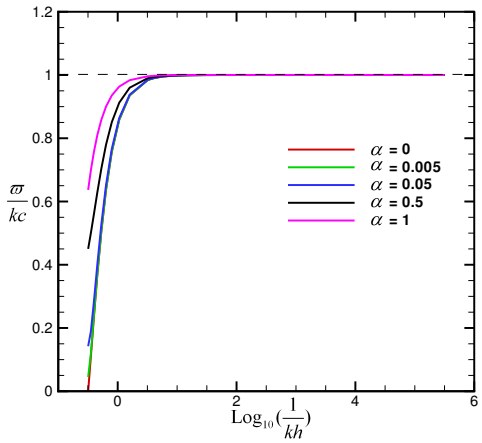
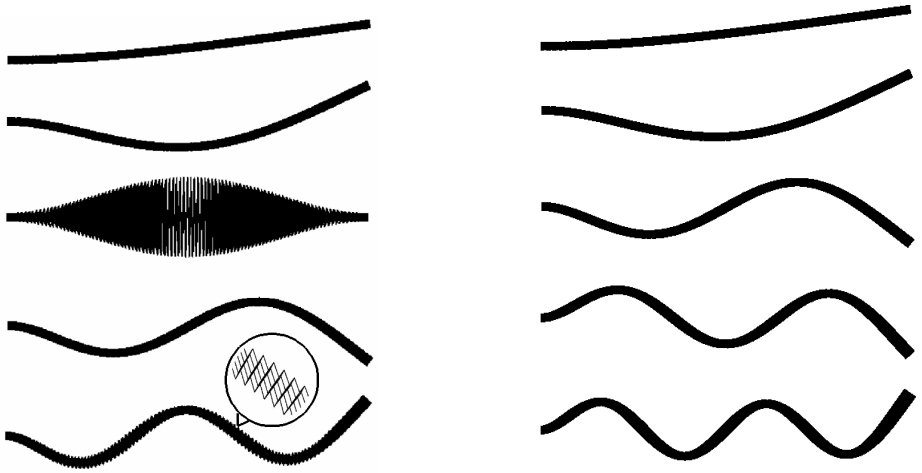


Figure 7.4: Dispersion relation under various α.

7.5 Stability test

7.5.1 Natural frequencies of cantilever beam

As we mentioned in Section 7.1, only the boundary cells contribute to the strain energy when interior nodal cells have zero average strain. Therefore, the total system energy may approach zero in the limit of mesh refinement. To support this discussion, we perform mode analysis on a very thin clamped cantilever beam with fine mesh. The beam's slenderness is 50. Both triangle and quadrilateral mesh with 6×200 regularly distributed nodes are considered to make the each nodal cell very small. Elastic material, which $E = 50000$, $\nu = 0.3$ and $\rho = 1$ is assumed. The natural frequencies computed under various stabilization parameters are listed in Tables 7.1 and 7.2. Form Table 7.1, we can see that spurious modes appear after the seventh mode in the unstable situation ($\alpha = 0$) with triangular mesh. With quadrilateral mesh and as $\alpha = 0$, spurious mode appears at the third mode, as shown in Table 7.2. Overall, as spurious modes appear, their corresponding frequencies are below the analytic values. However, the stabilized results are very close to analytic ones. In another word, the stabilized scheme preserves system energy very well. The first five modes obtained from the quadrilateral mesh are depicted in figure 7.5. It can be seen from Figure 7.5(a) that the spurious modes have the saw-teeth pattern in the middle span of the beam. The results of the triangular mesh follow the same pattern and are not reported here. Figure 7.5(b) shows the modes captured after stabilization ($\alpha = 0.05$).



(a) (b)

Figure 7.5: The first five mode shapes. (a) Unstabilized ($\alpha = 0$); (b) Stabilized ($\alpha = 0.05$).

Frequency	Analytic	Tri. DGM				
		$\alpha = 0$	$\alpha = 0.005$	$\alpha = 0.05$	$\alpha = 0.5$	$\alpha = 1$
1	0.072	0.072	0.072	0.072	0.078	0.084
2	0.453	0.448	0.449	0.453	0.489	0.526
3	1.268	1.251	1.252	1.263	1.365	1.468
4	2.484	2.439	2.442	2.464	2.661	2.861
5	4.106	4.008	4.012	4.047	4.371	4.698
6	6.134	5.942	5.948	6.000	6.481	6.963
7	8.568	7.818	8.234	8.307	8.972	9.635
8	11.407	8.226	10.853	10.945	11.826	12.694
9	14.652	10.842	13.785	13.909	15.023	16.112
10	18.302	13.771	17.012	17.166	18.541	19.883
11	22.358	15.616	20.512	20.670	22.358	23.966
12	26.819	16.994	24.266	24.490	26.455	28.343

Table 7.1: The first twelve frequencies with triangular mesh.

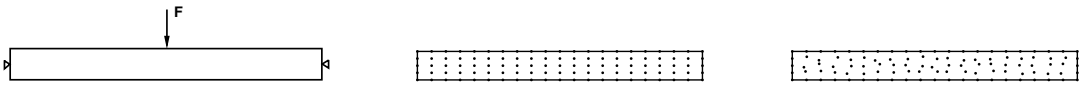
Frequency	Analytic	Quad. DGM				
		$\alpha = 0$	$\alpha = 0.005$	$\alpha = 0.05$	$\alpha = 0.5$	$\alpha = 1$
1	0.072	0.074	0.074	0.074	0.075	0.076
2	0.453	0.461	0.462	0.462	0.469	0.476
3	1.268	0.930	1.288	1.290	1.310	1.330
4	2.484	1.288	2.510	2.515	2.554	2.594
5	4.106	2.510	4.124	4.131	4.197	4.264
6	6.134	2.524	6.111	6.123	6.223	6.324
7	8.568	4.123	8.458	8.474	8.617	8.761
8	11.407	4.849	11.144	11.167	11.362	11.226
9	14.652	5.863	14.149	14.180	14.437	14.691
10	18.302	6.110	17.454	17.493	17.587	17.589
11	22.358	7.821	21.036	21.086	21.502	21.903
12	26.819	8.456	24.874	24.937	25.448	25.940

Table 7.2: The first twelve frequencies with quadrilateral mesh.

7.5.2 Supported beam subject to concentrated force

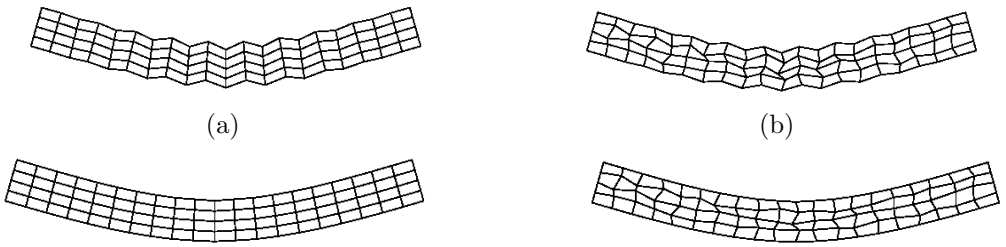
In this example, we can clearly see the need of stabilization for discrete gradient method. Without any additional stabilization, the spurious modes caused by zero average strain cannot be entirely excluded and may lead to wrong result although the nodal cell pattern is not symmetric. A two-dimensional beam example as shown in Figure 7.6(a) is employed to demonstrate spurious oscillation in DGM. The beam has a length 10 with a width 1, and plane strain condition is assumed. The material is linear isotropic ($E = 500$, $\nu = 0.3$). The force of 0.2 is applied at the middle span. As show in Figure 7.6(b) and (c), regularly and irregularly distributed 5×21 nodes with quadrilateral mesh are considered. Figure 7.7(a) and (b) are plots of the unstabilized numerical solution ($\alpha = 0$) corresponding to two types of node distribution. It is not surprised to see the saw-teeth pattern in 7.7(a) because the symmetric nodal cells, which are born by rectangular mesh, are prone to have zero average strain. However,

large errors are still apparent in Figure 7.7(b) although perturbation of nodal position changes the mesh to irregular but does not prevent spurious modes. Figure 7.7(c) and (d) show the plots of stabilized numerical solution ($\alpha = 0.05$) corresponding to regular and irregular mesh, respectively. The stabilized term has eliminated the large errors in Figure 7.7(a) and (b).



(a) (b) (c)

Figure 7.6: Beam model. (a) Supported beam subject to concentrated force; (b) Regularly distributed nodes; (c) Irregularly distributed nodes.



(a) (b)

(c) (d)

Figure 7.7: Deformed configuration of the beam. (a) Regularly distributed nodes without stabilization $\alpha = 0$; (b) Irregularly distributed nodes without stabilization $\alpha = 0$; (c) Regularly distributed nodes with stabilization $\alpha = 0.05$; (d) Irregularly distributed nodes with stabilization $\alpha = 0.05$.

7.5.3 Mode analysis of square area

To test the capability of DGM in dynamic analysis and the effect of stabilization parameter, we consider the other example of mode analysis. A 2×2 square area with regularly distributed 21×21 nodes is employed. The material is linear isotropic ($E = 50000$, $\nu = 0.3$ and $\rho = 1$). We still employ two types of domain partitions and compare their results with FEM. In Table 7.3 and 7.4, we list the first twelve eigenvalues calculated with different penalty. (The non-zero frequencies appear in pairs due to geometric symmetry. Here, only one mode is reported for each pair.) Among these eigenvalues listed in the two tables, the first three zero eigenvalues corresponding to rigid body motions. We also can see that spurious modes existing as $\alpha = 0$ and $\alpha = 0.005$. Clearly, a very small penalty is insufficient to suppress the spurious zero energy modes in this example. However, as $\alpha = 0.05, 0.5, 1.0$, the numerical solution are very close to the result of FEM.

Figure 7.8 shows the first twelve mode shapes calculated by unstabilized DGM. Among them, the fourth, fifth, sixth, eighth, eleventh and twelfth mode are all spurious modes. Figure 7.9 shows the corresponding modal shapes after stabilization $\alpha = 0.05$. Evidently, all the spurious modes are eliminated.

7.5.4 Effect of the stability parameter

It is of interest to detect the effect of the penalty term on the accuracy and convergence. We consider the problem of a cantilever beam bending illustrated in Figure 6.4.

A convergence study is performed using regularly distributed nodes of 3×11,

Mode	Tri. FEM	Tri. DGM				
		$\alpha = 0$	$\alpha = 0.005$	$\alpha = 0.05$	$\alpha = 0.5$	$\alpha = 1$
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	43.481	32.129	43.111	43.134	43.310	43.482
5	46.476	32.857	47.185	47.071	47.426	47.792
6	49.346	35.327	47.557	48.644	48.908	49.150
7	56.831	35.345	48.613	57.140	57.838	58.597
8	70.552	35.872	48.871	68.531	69.507	70.312
9	83.005	35.906	55.312	78.555	80.722	82.576
10	95.437	37.377	55.656	90.351	92.902	95.351
11	96.454	39.674	56.817	93.852	95.850	97.587
12	100.854	41.050	57.066	95.132	97.113	99.073

Table 7.3: The first twelve eigenvalues with triangular mesh.

Mode	Quad. FEM	Quad. DGM				
		$\alpha = 0$	$\alpha = 0.005$	$\alpha = 0.05$	$\alpha = 0.5$	$\alpha = 1$
1	0	0	0	0	0	0
2	0	0	0	0	0	0
3	0	0	0	0	0	0
4	43.400	32.917	43.115	43.135	43.287	43.437
5	46.233	35.451	45.283	47.192	47.392	47.589
6	49.094	39.546	47.183	48.684	48.797	48.911
7	55.787	43.110	48.538	57.309	57.512	57.728
8	69.783	43.587	51.839	68.719	69.244	69.766
9	81.012	47.165	53.686	78.829	80.029	81.174
10	93.480	48.670	57.288	90.966	92.331	93.619
11	94.850	50.710	60.941	94.144	95.271	96.348
12	98.584	54.601	64.331	95.419	96.270	97.131

Table 7.4: The first twelve eigenvalues with quadrilateral mesh.

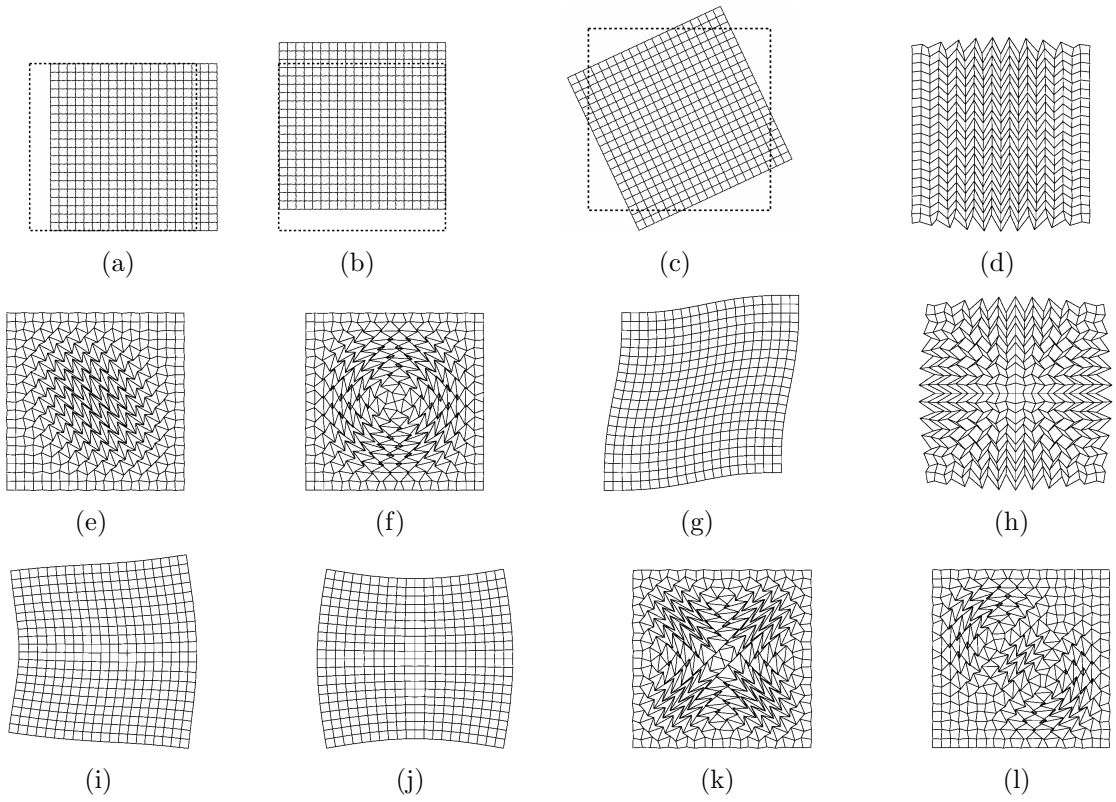


Figure 7.8: The first twelve modes calculated by DGM without stabilization ($\alpha = 0$).

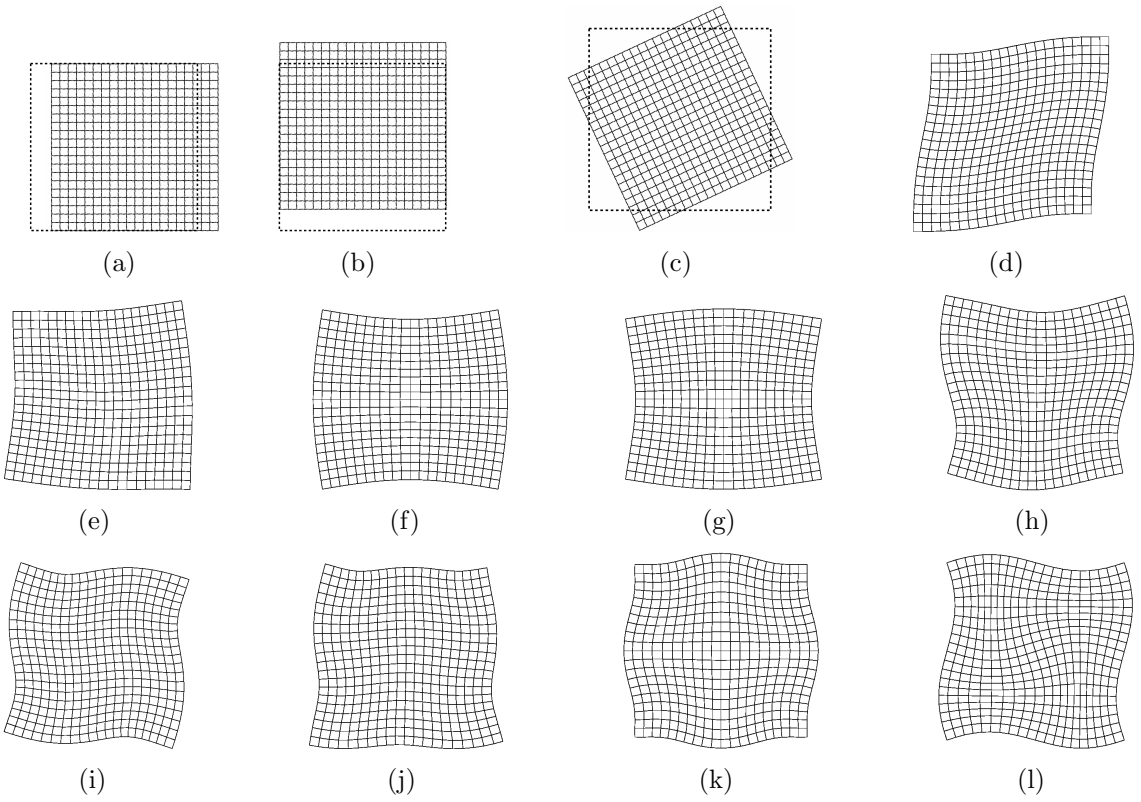
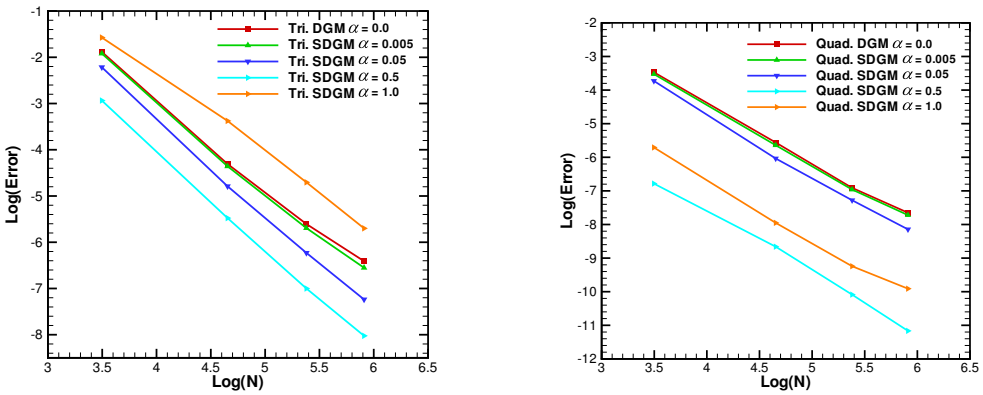


Figure 7.9: The first twelve modes calculated by stabilized DGM ($\alpha = 0.05$).

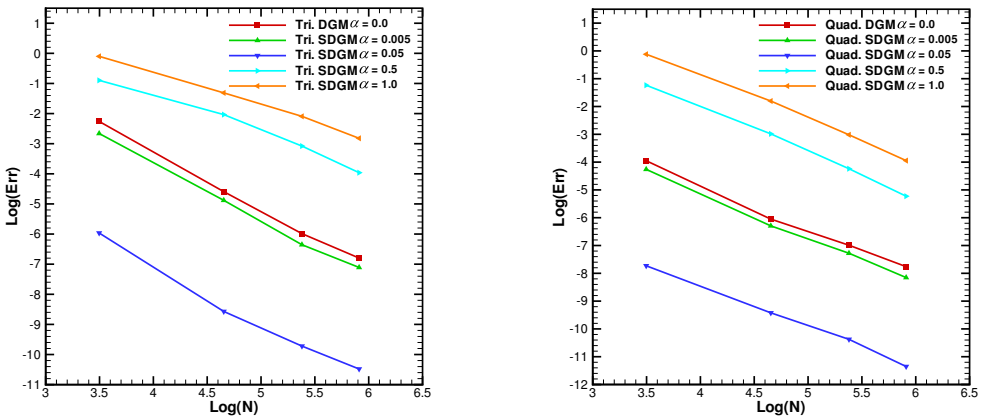
5×21, 7×31 and 9×41. Triangle and quadrilateral mesh are considered. Both a compressible material ($E = 500$, $\nu = 0.3$) and a nearly incompressible material ($E = 500$, $\nu = 0.4999$) are considered. Five seniors of $\alpha = 0.0, 0.005, 0.05, 0.5, 1$ are tested to study the influence of the penalty. The error is measured by the same function in Equation (6.1).

As is known, average strain methods usually have a softer stiffness. On the contrary, FEM and SFEM have harder stiffness. Therefore, in our stabilized scheme as $0 < \alpha < 1$, a proper selection of penalty may lead to more accurate result because of the combined stiffness. While, as $\alpha = 1$, the stabilization formulation degenerates to SFEM and its behavior is same as the low order FEM. This assumption can be proved by the convergence results shown in Figure 7.10 and 7.11. For compressible material, the results become more accurate as α increases from 0 till to 0.5. When $\alpha = 1$, the accuracy is worse because the results just have the same accuracy as triangular and bilinear finite element. For nearly incompressible material, the best convergence is reached as $\alpha = 0.05$. Moreover, it not surprised to see the appearance of volume locking in stabilization formulation as $\alpha = 1$ because the current formulation is same as low order FEM and suffers with severe volume locking for incompressible material. Overall, the stabilized formulation displays a better accuracy than low order FEM. It is also more accurate than unstabilized scheme with a proper penalty. This test confirms the that $\alpha = 0.05$ is proper selection which is consistent with the suggestion of Puso.



(a) (b)

Figure 7.10: Error norm versus number of nodes with compressible material. (a) Triangle mesh; (b) Quadrilateral mesh.



(a) (b)

Figure 7.11: Error norm versus number of nodes with incompressible material. (a) Triangle mesh; (b) Quadrilateral mesh.

CHAPTER 8
IMAGED-BASED DISCRETE ANALYSIS

8.1 Apply discrete gradient method in biomechanical analysis

The aim of developing discrete gradient method is to perform stress analysis directly on domains represented by point-clouds as happens in medical images without converting the model into a finite element mesh. The method computes the gradient of unknown field variables at a node using discrete differentials involving a set of neighboring nodes, and then establish the discrete governing equations by substituting the discrete gradient into a Galerkin weak form. Therefore, its computational structure is very simple to implement since it bypasses the numerical complications associated with the construction of implicit shape functions in meshfree methods. It has been demonstrated that the method retains the accuracy and convergence rate of the displacement FEM but exhibits a locking-free behavior in the incompressible limit. Hereby, the method offers a greater flexible modeling tool to conduct mechanical analysis on point-cloud representations of patient-specific organs without resorting to finite element method.

On the other hand, it is necessary to form Delaunay triangulation among the complicated point-cloud model because of the need to define nodal cells and boundary. Please notate that forming Delaunay triangulation is not equal to creating triangular mesh for FEM. A correctly identified boundary is required in defining nodal cells and applying boundary conditions for analysis. To carry out mechanical analysis directly upon image-based model with the discrete solvers, the procedure consists of

the following steps:

1. Convert pixels or voxels of images to material points by following some selection criteria and form a point-cloud model.

2. Implement Delaunay triangulation among the point-cloud model.

3. Identify boundary of the point-cloud model and remove spurious triangles or tetrahedrons.

4. Partition the domain into a set of non-overlapping, node-centered cells using existing triangles or tetrahedrons.

5. At each node, approximate the gradient of the primary unknown by a linear combination of nodal values in a set of neighboring nodes.

6. Substitute the gradients into a Galerkin weak from to derive the algebraic equations governing the nodal variables.

8.2 Extract point-cloud model from medical image

Radiographic images register different physical constituents (soft tissue, bone, fluid etc) by different image types. In our work, we employ a basic type of gray-scale image, also known as gray level image. In this type of images, the pixel values represent intensity within some range. Therefore, it is straightforward to identify a material constituent, or a specific organ comprised of certain material by extracting pixels within certain range of gray-scale value. In addition, pixel coordinates can be easily transferred to physical spatial coordinates with the resolution information

(between pixels and slices) of images. Representing each pixel by a point yields point-cloud model of organs at the pixel resolution.

Figure 8.1 shows a cross section scan of an aorta with 0.2 mm resolution. Since the data matrix in this image is saved in the format of "unit8", We select pixels with gray-scale values in the range of $200 - 255$ to correspond material points in the aorta wall. By retaining the pixels within the given range we obtain a discrete representation of the aorta at the pixel-resolution as shown in Figure 8.3(a).

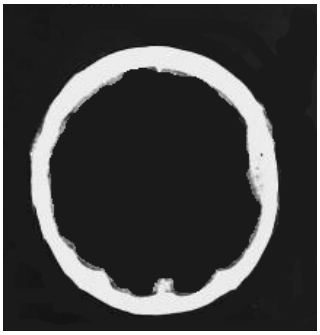


Figure 8.1: Scanning image of aorta.

However, pixel-resolution model usually contains too many points and the model size may be prohibitively large for mechanical analysis. To save the computational cost, the model can be made coarse by lumping several pixels into a node. To fulfill this idea, the notion of *window* is introduced. Moreover, this concept will play important role in our succeeding development. In our work, a two-dimensional window is defined as a square that size is large enough to contain $n \times n$ pixels. In three-dimensional applications, the window is a cube that contains $n \times n \times m$ pixels.

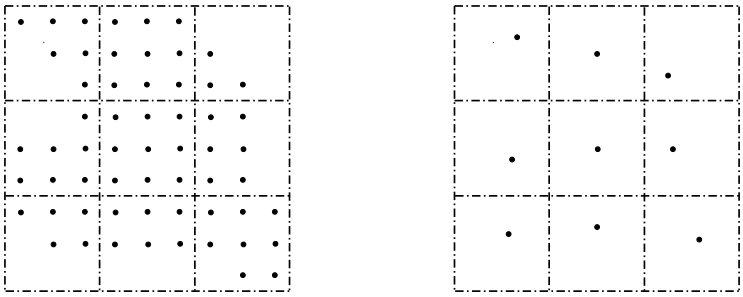
Here the m denotes the number of image slices. Consequently, image(s) can be divided into an array of non-overlapping windows, which cover the entire image(s). By concentrating pixels inside a window into one material point, the coarser models can be derived. Similar to the determination of the mass center of several particles, the position of the material point in one window is calculated according to

$$\mathbf{X} = \frac{\sum_I^N \mathbf{X}_I \varphi_I}{\sum_I^N \varphi_I}, \tag{8.1}$$

where $\mathbf{X}_I$ and φ_I are, respectively, the spatial coordinates of the each pixel and the its gray-scale value, N represents all pixels in the window. Figure 8.2(a) shows several pixels and defined windows in the size of containing 3×3 pixels. And Figure 8.2(b) shows the material points after lumping pixels in windows. Consequently, the points in the model have been reduced with 9-fold. Following the same way, a point-cloud aorta model containing 895 material points is derived and is shown in Figure 8.3(b). The model is extracted from above CT image using 3×3 windows, which are 0.6mm $\times$ 0.6mm in spatial size. Notably, the procedure can be made with minimal user interference, or even fully automated.

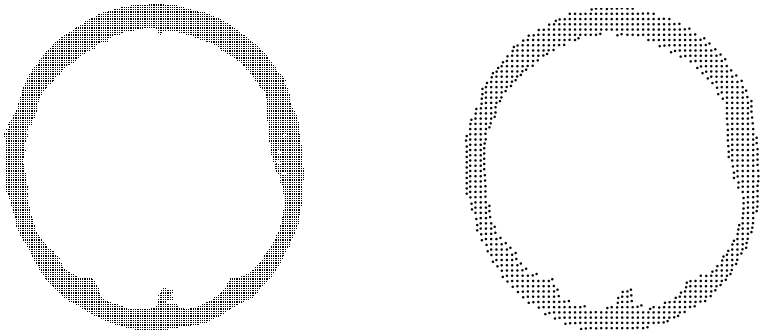
8.3 Tessellation of point-cloud model

The implementation of discrete gradient method relies on the domain tessellation to identify the neighbouring relation between points. In our work, an open source program Qhull [3] is employed to perform Delaunay triangulation on derived point clouds. However, this general algorithm of tessellation cannot identify the con-



(a) (b)

Figure 8.2: windows and pixels grouping. (a) Windows containing 3×3 pixels; (b) No more than one material point can exist in each window after pixels grouping.



(a) (b)

Figure 8.3: Aorta point cloud model extracted from image. (a) Point-cloud model extracted by pixel resolution. (b) Point-cloud model extracted by window size 3×3.

figuration of point clouds and may produce some spurious triangles or tetrahedrons, which lead to a wrong identification of nodal cells and boundary of point clouds. Figure 8.5(a) shows Delaunay triangles formed by Qhull upon a set of points in a plane. Obviously, several points are wrongly connected together through spurious Delaunay triangles. The incorrectly defined relationships among points not only throw the point-cloud model into confusion but also lead to wrong analysis result. Therefore, the point clouds need some reconstruction after the triangulation to delete those incorrect connections among points. For general point clouds, available method is α-shape, which was introduced by Edelsbrunner [40, 41]. α-shapes can provide the mathematical framework and a general algorithm to define of the hierarchy of point clouds through Delaunay triangulation, and some applications of α-shape upon point clouds can be found in [24, 30, 31]. However, we propose a more efficient algorithm that works well for point clouds extracted from images with aid of their special data structure.

8.3.1 Two-dimensional triangular tessellation

For the window array defined over an image, each window is always surrounded by other neighbouring windows, which can be called *adjacent windows*. Hereby, a window can have eight adjacent windows at most. By construction, each window only contains no more than one material point. Therefore, we can identify all points in adjacent windows are the *adjacent points* of the material points occupying the central window. For example, in Figure 8.4, point A has eight adjacent points but point B has seven. With this predefined adjacent relationships among all points, we

define a triangle is not spurious triangle when at least one of its three vertexes is simultaneously adjacent to the other two. Therefore, we inspect every triangle after the implementation of Delaunay triangulation and delete all spurious triangles. All existing triangles will give a correct tessellation of the point clouds. The result can be seen in Figure 8.5(b)

In addition, care must be exercised to "hanging points" (points do not properly connect to the rest of the clouds) before the tessellation. If a point just has one adjacent point, we think this point is isolated and should be removed from model.

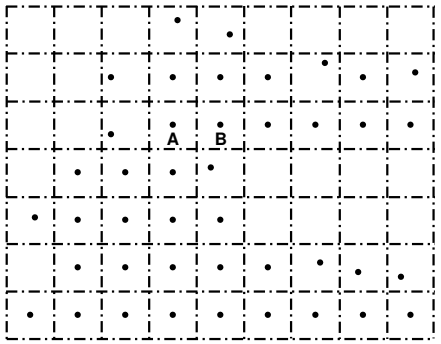
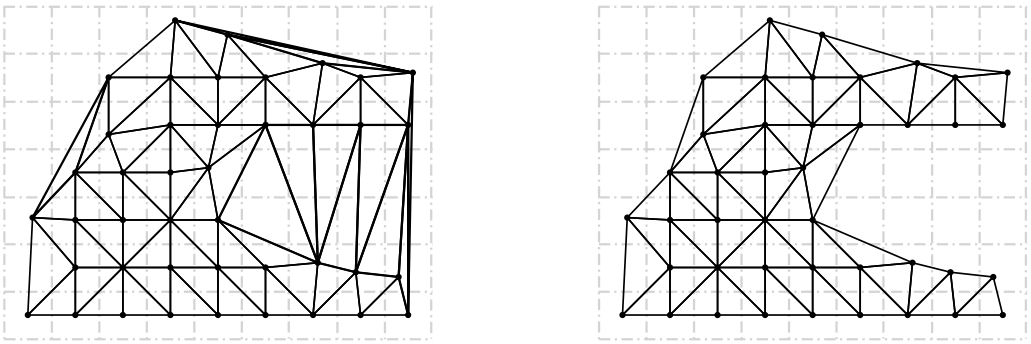


Figure 8.4: Adjacent points defined by windows.

8.3.2 Three-dimensional tetrahedral tessellation

Following the above definitions, a material point may have at most 26 adjacent points in three-dimensional case. If one point has less three adjacent points, this point is considered as "hanging points" and should be deleted before tessellation. Similar to two-dimensional definition, a tetrahedron is not a spurious tetrahedron if one of



(a) (b)

Figure 8.5: Delaunay triangulation of point clouds. (a) Delaunay triangulation before removing spurious triangles; (b) Delaunay triangulation after removing spurious triangles.

its vertexes is simultaneously adjacent to the other three. Moreover, the definition of spurious tetrahedron is not unique. For example, a spurious tetrahedron can be defined if its four vertexes are not simultaneously adjacent to each other. It can be imaged that different definitions may lead to various profiles of point-cloud model. As well as, such definitions need to consider the requirements of accuracy and boundary smoothing.

8.4 Boundary tracking of point-cloud model

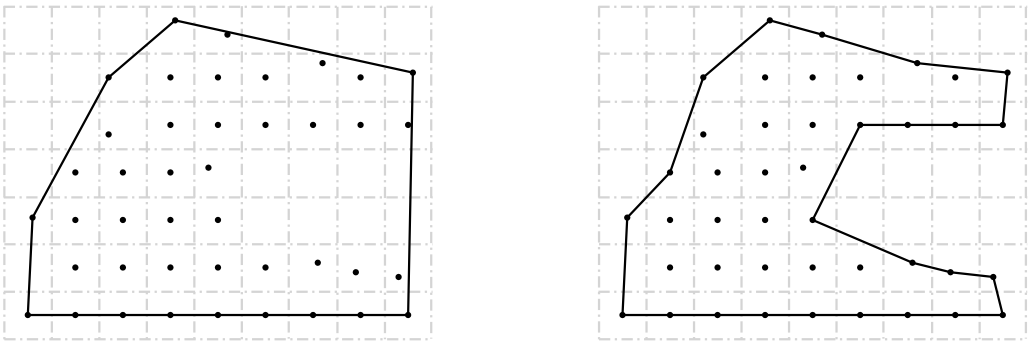
For the reason of applying boundary conditions before any analysis, it is necessary to identify the boundary facets of a point-cloud model. This process relies on the existing triangles or tetrahedrons. A general method is to inspect each existing facet and examine whether this facet is shared by other triangles (tetrahedrons) or not. If the facet only exists in one triangle (tetrahedrons), this facet will be considered as boundary facet and its vertices will be the boundary points. In the end, all

the boundary facets will form the boundary of this model. Since this algorithm needs to search all facets, it is very time consuming especially for large scale model. However, we can reduce the searching range and save time using the predefined adjacent relationships among points.

By observing Figure 8.4, we can find that points with eight adjacent points always stay inside the domain of clouds and are interior points. On the contrary, those points, which have less eight adjacent points, can be considered as "boundary points". Please notice that we use "boundary points" here to indicate these points are candidates of boundary points but not the indeed ones. In remaining triangles, edges only connecting two "boundary points" will be picked up for further consideration. Figure 8.6(a) shows the identified boundary before removing spurious triangles. It can be seen that this boundary cannot accurately represent the profile of this sets of points. In Figure 8.6(b), it shows the defined boundary of the point cloud after removing all spurious triangle. And, the boundary is more coincident with the profile of the point clouds.

This procedure can be easily expanded to three-dimensional application. In three-dimensional case, we can define a point as "boundary points" only if this point has less than 26 adjacent points. The other searching process to define boundary is exactly same as process in two-dimensional boundary tracking.

Overall, the proposed method to extract point-cloud model from images is very simple without requiring user-specified algorithmic parameters such as the α. This method considerably reduces the computer time during domain tessellation and



(a) (b)

Figure 8.6: Boundary defined by window technique. (a) Boundary of point cloud before removing spurious triangles; (b) Boundary of point cloud after removing spurious triangles.

boundary tracking.

8.5 Discrete stress analysis on point-cloud model

8.5.1 The inflation of aorta

The point-cloud model shown in Figure 8.3(b) is assumed to represent the stress free configuration of the aorta. The deformed configuration and wall stress under 100 mmHg are sought. Although aortic tissue typically exhibits anisotropic nonlinear behavior, for demonstration purpose, we model the wall tissue by an isotropic hyperelastic energy function

$$W = \frac{\mu_1}{2}(I_1 - 3) + \frac{\mu_2}{4}(I_1 - 3)^2,$$

where $I_1 = \mathrm{tr}\,(\mathbf{F}^T\mathbf{F})$. A penalty energy $\frac{\kappa}{2}(\log(J))^2$ with a large value κ is augmented to the energy function to introduce nearly incompressibility. The following numerical

values are used:

$$\mu_1 = 100, \qquad \mu_2 = 5, \qquad \kappa = 1000000.$$

To simplify the problem we also ignore the residual stress, which is known to exist in aortas. Displacement constraints just enough to eliminate rigid body motion are imposed. The pressure is applied on the inner wall as a follower-force.

Stress analysis is first performed on the point model extracted by 3×3 windows. For comparison, an alternative FE model consisting of mainly four-node mixed element [88] is also implemented. While the quadrilateral mesh requires a manual creation of the geometric model (the cross section area) first. The area is then meshed in commercial software.

Figure 8.7 shows the distributions of the von Mises stress from the two models. As shown from the stress contours, the stress results from all models are close to each other. The point model results however appear to be closer to that of the quadrilateral mesh, indicating that the discrete method achieves nearly the same accuracy as the four-node mixed element in this problem.

8.5.2 The impact of three dimensional skull

In this example, we demonstrate the propagation of stress wave in a skull under impact. A life-size plastic skull model is scanned by CT with 0.4 mm resolution (between pixels and slices). The entire image stack contains 400 CT images. In each image, the gray-scale values are stored in a 512×512 data matrix. To extract three-dimension model with accurate bone structure, we employ $4 \times 4 \times 4$ windows in

CHAPTER 9
CONCLUSION

The dissertation presents a novel method belonging to discrete solvers, which work directly on nodal values and discrete differentials without introducing continuous approximation of the primary unknown. The gradient of the field variable is computed using discrete differentials for arbitrary nodes and then is incorporated into a weak form of Galerkin formation. The most noteworthy attribute in the present method is the absence of continuous interpolation (or approximation) of the unknown variable. Compared to other continuum based average strain methods, the gradient in the present approach is computed directly from discrete nodal values using a closed-form formula. The method therefore bypasses the numerical difficulties associated with the construction of implicit shape functions in either meshfree methods or element-base methods. Meanwhile, it avoids the loss of interpolantory property, which is common to meshfree shape functions.

Voronoi diagram and general convex polygons are employed to tessellate the domain into a set of non-overlapping nodal cells for the identification of nodal supports in discrete differentials. The tessellation process is fully automatic and does not require algorithmic parameters such as the size of influence region. Over each nodal cell, the constructions of gradient interpolants are proposed to compute gradient of field variable. In addition, all developed gradient interpolants show that they have a common tensorial identity property, although the constructions of gradient interpolants are different with various definitions of supporting domain,

Mathematically and numerically, it has been proved that this discrete gradient method automatically satisfies the derivative consistency conditions and the patch test. In addition, with the Kronecker delta properties, it is straightforward to introduce essential boundary condition and to couple the method with the finite element method. Moreover, numerical tests show that the method not only has comparable accuracy and convergence rate as the displacement finite element method, but also displays robustness for large deformation simulations and hyper-elasticity. This method resists numerical locking in the incompressibility limit too.

The stability of discrete gradient method is fully investigated. With analytic solution and numerical examples, the existence of spurious modes in current method is exhibited. To suppress the instability, a stabilization scheme is presented under the Galerkin framework with penalized strain energy. The modified formulation is tested against several benchmark problems. Moreover, the penalty influence to stability, accuracy and convergence is studied. Overall, the modified formulation shows to be stable and gives a superior performance for both compressible and nearly incompressible material.

Since of the newly developed method is feasible to be implemented in the stress analysis of point-cloud models, applications in biomechanical field are demonstrated. To fulfill the promise of delivering automated analysis, an efficient method is developed to automatically extract point-cloud models from medical images, which provide depiction for complicated anatomies. With exhibited two and three-dimensional examples, the method demonstrates its advantages as a flexible numerical tool for

engineering analysis, particularly in biomechanical applications.

REFERENCES

[1] F. M. Andrade, E. A. de Souza Neto, and J. L. de la Cuesta Padilla. An assessment of the average nodal volume formulation for the analysis of nearly incompressible solids under finite strains. *Communications in Numerical Methods in Engineering*, 20:569–583, 2004.

[2] S. N. Atluri and T. Zhu. A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics. *Computational Mechanics*, 22:117–127, 1998.

[3] C. B. Barber, D. P. Dobkin, and H. T. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Trans. on Mathematical Software*, 22:469–483, 1996.

[4] T. Barth and M. Ohlberger. Finite volume methods: foundation and analysis. In Erwin Stein, Rene de Borst, and Thomas J. R. Hughes, editors, *Encyclopedia of Computational Mechanics*. John Wiley and Sons, 2004.

[5] R. S. Baty and W. P. Villon. Least-square solutions of a general numerical method for arbitrary irregular grids. *International Journal for Numerical Methods in Engineering*, 140:1701–1717, 1997.

[6] S. Beissel and T. Belytschko. Nodal integration of the elementfree galerkin method. *Computer Methods in Applied Mechanics and Engineering*, 139:49–74, 1996.

[7] V. V. Belikov, V. D. Ivanov, V. K. Kontorovich, S. A. Korytnik, and 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*, 37:9–15, 1997.

[8] V. V. Belikov and A. Yu. Semenov. Non-Sibsonian interpolation on arbitrary system of points in Euclidean space and adaptive isoline generations. *Applied Numerical Mathematics*, 32:371–387, 2000.

[9] T. Belytschko, Y. Guo, W. K. Liu, and S. P. Xiao. A unified stability analysis of meshless particle methods. *International Journal for Numerical Methods in Engineering*, 48:1359–1400, 2000.

[10] T. Belytschko, Y. Y. Lu, and L. Gu. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[11] T. Belytschko, D. Organ, and C. Gerlach. Element-free galerkin method for dynamic fracture in concrete. *Computer Methods in Applied Mechanics and Engineering*, 187:385–399, 2000.

[12] W. Benz. Smooth particle hydrodynamics: a review. In *Numerical Modeling of Non-linear Stellar Pulsation: Problems and Prospects*. Kluwer Academic, Boston, 1990.

[13] J. Bonet and A. J. Burton. A simple average nodal pressure tetrahedral element for incompressible and nearly incompressible dyanmic explicit applications. *Communications in Numerical Methods in Engineering*, 14:437–449, 1998.

[14] J. Bonet and S. Kulasegaram. Correction and stabilization of smooth particle hydrodynamics methods with applications in metal forming simulations. *International Journal for Numerical Methods in Engineering*, 47:1189–1214, 2000.

[15] J. Bonet, H. Marriott, and O. Hassan. An averaged nodal deformation gradient linear tetrahedral element for large strain explicit dynamic applications. *Communications in Numerical Methods in Engineering*, 17:551–561, 2001.

[16] S. K. Boyd and R. Müller. Smooth surface meshing for automated finite element model generation from 3D image data. *Journal of Biomechanics*, 39:1287–1295, 2006.

[17] P. Breitkopf, G. Touzot, and P. Villon. Consistency approach and diffuse derivation in element free methods beased on moving least squares approximation. *Computer Assisted Mechanics and Engineering Sciences*, 5:479–501, 1998.

[18] P. Breitkopf, G. Touzot, and P. Villon. Double grid diffuse collocation method. *Computational Mechanics*, 25:199–206, 2000.

[19] D. Bushnell. Analysis of buckling and vibration of ring-stiffened, segmented shells of revolution. *International Journal of Solids and Structures*, 6:157–181, 1970.

[20] Z. Cai. On the finite volume element method. *Numerische Mathematik*, 58:713–735, 1991.

[21] Z. Cai, J. Mandel, and S. M. Cormick. The finite volume element method for diffusion equations on general triangulations. *Journal on Numerical Analysis*, 28:392–402, 1991.

[22] D. L. A. Camacho, R. H. Hopper, G. M. Lin, and B. S. Myers. An improved method for finite element meeh generation of geometrically complex structures with application to the skullbase. *Journal of Biomechanics*, 30:1067–1070, 1997.

[23] D. R. Cater and W. C. Hayes. The compressive behavior of bone as a two-phase porous structure. *Journal of Bone Joint Surgery*, 59:954–962, 1977.

[24] F. Cazals and J. Giesen. Conformal alpha shapes. *Eurograhhics Symposium*.

[25] J. S. Chen, C. T. Wu, S. Yoon, and Y. You. A stabilized conforming nodal integration for galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 50:435–466, 2001.

[26] J. S. Chen, S. Yoon, and C. T. Wu. Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. *International Journal for Numerical Methods in Engineering*, 53:2587–2615, 2002.

[27] P. G. Ciarlet. *The finite element method for elliptic problems*. North-Holland, Amsterdam, 2002.

[28] F. Cosmi. Numerical solution of plane elasticity problems with the cell method. *Modeling in engineering and science*, 2:365–372, 2001.

[29] F. H. Croom. *Basic concepts of algegraic topology*. Springer, Berlin, 1978.

[30] E. Cueto, B. Calvo, and M. Doblare. Modelling three dimensional piece-wise homogenetous domains using the α-shape-based natural element method. *International Journal for Numerical Methods in Engineering*, 54:871–897, 2002.

[31] E. Cueto, J. Cegonino, B. Calvo, and M. Doblare. On the imposition of essential boundary conditions in natural neighbour galerkin methods. *Communications in Numerical Methods in Engineering*, 19:361–376, 2003.

[32] E. A. de Souza Neto, D. Pericacutec, M. Dutko, and DRJ Owen. Design of simple low order finite elements for large strain analysis of nearly incompressible solids. *International Journal for Numerical Methods in Engineering*, 33:3277–3296, 96.

[33] M. Delanaye, P. Geuzaine, and J. A. Esser. The quadratic reconstruction finite volume scheme: An attractive sequel to linear reconstruction, used on unstructured adaptive meshes. *Lecture Notes in Physics*, 490:617–622, 1997.

[34] M. Delanaye, P. Geuzaine, J. A. Esser, and P. Rogiest. A second-order finite-volume scheme solving euler and navier-stokes equations on unstructured adaptive grids with an implicit acceleration procedure. *In AGARD-CP-578, Progress and Challenges in CFD Methods and Algorithms*, 1995.

[35] M. Doblare, E. Cueto, B. Calvo, M. A. Martinez, J. M. Garcia, and J. Cegonino. On the employ of meshless methods in biomechanics. *Computer Methods in Applied Mechanics and Engineering*, 194:801–821, 2005.

[36] C. R. Dohrmann, M. W. Heinstein, J. Jung, S. W. Key, and W. R. Witkowski. Node-based uniform strain elements for three-node triangular and four-node tetrahedral meshes. *International Journal for Numerical Methods in Engineering*, 47:1549–1568, 2000.

[37] C. A. Duarte and J. T. Oden. H-p clouds – an h-p meshless method. *Numerical Methods for Partial Differential Equations*, 12:673–705, 1996.

[38] C. T. Dyka. Addressing tension instability in SPH method. *Technical report NRL/MR/6384*, 1994.

[39] E. Onate E, S. R. Idelsohn, O. C. Zienkiewicz, R. L. Taylor, and C. Sacco. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 139:315–346, 1996.

[40] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *Information Theory*, 29:551–559, 1983.

[41] H. Edelsbrunner and E. P. Mucke. Three-dimensional alpha shapes. *ACM Transactions on Graphics*, 13:43–72, 1994.

[42] R. Eymard, T. Gallouet, and R. Herbin. Finite volume method. In *Handbook of Numerical Analysis*, volume 7, pages 713–1020. North Holland, 2000.

[43] R. Eymard, T. Gallouet, and R. Herbin. Finite volume approximation of elliptic problems and convergence of an approximate gradient. *Applied Numerical Mathematics*, 37:31–53, 2001.

[44] R. Eymard, T. Gallouet, and R. Herbin. A cell-centered finite volume approximation for second order partial derivative operators with full matrix on unstructured meshed in any space dimension. *IMA Journal of Numerical Analysis*, 26:326–353, 2006.

[45] E. Ferretti. Crack propatation modeling by remeshing using the cell method. *Modeling in engineering and science*, 4:51–72, 2003.

[46] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20:19–27, 2003.

[47] M. S. Floater. One-to-one piecewise linear mappings over triangulations. *Mathematics of Computations*, 72:685–696, 2003.

[48] D. P. Frey, B. Sarter, and M. Gauthene. Fully automated mesh generation for 3-d domains based upon voxel data sets. *International Journal for Numerical Methods in Engineering*, 37:2735–2753, 1994.

[49] D. P. Fyhire, M. S. Hamid, R. F. Kuo, and S. M. Lang. Direct three-dimensional finite element analysis of human vertebral cancellous bone. *Transactions of* 38^th *Annual Meeting of Orthopaedic Research Society*, 1992.

[50] R. Gingold and J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181:375–389, 1977.

[51] D. González, E. Cueto, M. A. Martínez, and M. Doblaré. Numerical integration in natural neighbour galerkin methods. *International Journal for Numerical Methods in Engineering*, 60:2077–2104, 2004.

[52] AE Green and W. Zerna. *Theory of Elasticity*. Clarendon Press, Oxford, 1954.

[53] S. Hao, H. S. Park, and W. K. Liu. Moving particle finite element method. *International Journal for Numerical Methods in Engineering*, 53:1937–1958, 2002.

[54] S. J. Hollister, J. M. Brennan, and N. Kikuchi. A homogenization sampling procedure for calculating trabecular bone effective stiffness and tissue elvel stress. *Jouranl of Biomechanics*, 27:433–444, 1994.

[55] S. R. Idelsohn and E. Onate. Finite volumes and finite elements: two 'good friends'. *International Journal for Numerical Methods in Engineering*, 37:3323–3341, 1994.

[56] P. S. Jensen. Finite difference techniques for variable grids. *Computers and Structures*, 2:17–29, 1972.

[57] D. E. Johnson. A difference-based variational methods for shells. *International Journal of Solids and Structures*, 6:699–724, 1970.

[58] J. H. Keyak, J. M. Meagher, H. B. Skinner, and C. D. Mote. Automated three-dimensional finite element modelling of bone: a new method. *Journal of Biomedical engineering*, 12:389–397, 1990.

[59] Y. Krongauz and T. Belytschko. Consistent pseudo-derivatives in meshless methods. *Computer Methods in Applied Mechanics and Engineering*, 146:371–386, 1997.

[60] Y. Krongauz and T. Belytschko. A Petrov-Galerkin diffuse element method (PG DEM) and its comparison to EFG. *Computational Mechanics*, 19:327–333, 1997.

[61] P. Krysl and B. Zhu. Locking-free continuum displacement finite elements with nodal integration. *International Journal for Numerical Methods in Engineering*, page 10.1002/nme.2354., 2008.

[62] T. Liszka. An interpolation method for an irregular net of nodes. *International Journal for Numerical Methods in Engineering*, 20:1599–1612, 1984.

[63] T. Liszka, C. A. M. Duarte, and W. W. Tworzydlo. hp-meshless cloud method. *Computer Methods in Applied Mechanics and Engineering*, 139:263–288, 1996.

[64] T. Liszka and J. Orkisz. The finite difference method at arbitrary irregular grids and its application in applied mechanics. *Computers and Structures*, 11:83–95, 1980.

[65] G. R. Liu, K. Y. Dai1, and T. T. Nguyen. A smoothed finite element method for mechanics problems. *Computational Mechanics*, 39:859–877, 2007.

[66] G. R. Liu, T. T. Nguyen, K. Y. Dai1, and K. Y. Lam. Theoretical aspects of the smoothed finite element method. *International Journal for Numerical Methods in Engineering*, 71:902–930, 2007.

[67] W.K. Liu, S. Jun, and Y. F. Zhang. Reproducing kernel particle methods. *International Journal for Numerical methods in Fluids*, 20:1081–1106, 1995.

[68] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.

[69] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astronomical Journal*, 82:1013–1024, 1977.

[70] R. H. MacNeal. An asymmetrical finite difference network. *Quarterly of Applied Mathematics*, 11:295–310, 1953.

[71] L. W. Marks and T. N. Gardner. The use of strain energy as convergence criterion in the finite element modelling of bone and the effect of model geometry on stress convergence. *ASME Journal of Biomedical Engineering*, 15:474–476, 1993.

[72] J. M. Melenk and I. Babuska. The partition of unity finite element method: basic theory and applications. *Computer Methods in Applied Mechanics and Engineering*, 139:289–314, 1996.

[73] M. Meyer, H. Lee, A. H. Barr, and M. Desbrun. Generalized barycentric coordinates for irregular n-gons. *Journal of Graphics Tools*, 7:13–22, 2002.

[74] J. Monaghan. Why particle methods works. *SIAM Journal on Scientific Computing*, 3:422–433, 1982.

[75] J. Monaghan. Particle methods for hydrodynamics. *Computational physics report*, 3:71–124, 1985.

[76] J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48:89–96, 1988.

[77] J. Monaghan. Smoothed particle hydrodynamics. *Report on progress in physics*, 68:1703–1759, 2005.

[78] R. Müller and P. Rüegsegger. Three-dimensional finite element modeling of non-invasively assessed trabecular bone structures. *Jouranl of Medical Engineering and Physics*, 17:126–133, 1995.

[79] E. Onate, M. Cervera, and O. C. Zienkiewicz. A finite volume format for structural mechanics. *International Journal for Numerical Methods in Engineering*, 37:181–201, 1994.

[80] E. Onate and S Idelsohn. A mesh-free finite point method for advective-diffusive transport and fluid flow problems. *Computational Mechanics*, 21:283–292, 1998.

[81] E. Onate, S. R. Idelsohn, O. C. Zienkiewicz, and R. L. Taylor. A finite point method in computational mechanics applications to convective transport and fluid flow. *International Journal for Numerical Methods in Engineering*, 39:3839–3866, 1996.

[82] V. Pavlin and N. Perrone. Finite difference energy techniques for arbitrary meshes applied to linear plate problems. *International Journal for Numerical Methods in Engineering*, 14:647–664, 1979.

[83] N. Perrone and R. Kao. A general finite difference method for arbitrary meshes. *Computers and Structures*, 5:45–57, 1975.

[84] M. A. Puso, J. S. Chen, E. Zywicz, and W. Elmer. Meshfree and finite element nodal integration methods. *International Journal for Numerical Methods in Engineering*, 74:416–446, 2008.

[85] M. A. Puso and J. Solberg. A stabilized nodally integrated tetrahedral. *International Journal for Numerical Methods in Engineering*, 67:841–867, 2006.

[86] A. A. Samarskii. On monotone difference schemes for elliptic and parabolic equations in the case of a nonselfadjoint elliptic operator. *Computational Mathematics and Mathematical Physics*, 5:548–551, 1965.

[87] R. Sibson. A vertor identity for the dirichlet tessellation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 87:151–155, 1980.

[88] J. C. Simo and M. 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.

[89] J. C. Simo, R. L. Taylor, and K. S. Pister. Variational and projection methods for the volume constraint in finte deformation elasto-plasticity. *Computer Methods in Applied Mechanics and Engineering*, 51:177–208, 1985.

[90] N. Sukumar. Voronoi cell finite difference method for the diffusion operator on arbitrary unstructured grids. *International Journal for Numerical Methods in Engineering*, 57:1–34, 2003.

[91] N. Sukumar and J. E. Bolander. Numerical computation of discrete differential operators on non-uniform grids. *Computer Modeling in Engineering and Sciences*, 4:691–705, 2003.

[92] N. Sukumar, B. Moran, A. Yu Semenov, and V. V. Belikov. Natural neighbour Galerkin method. *International Journal for Numerical Methods in Engineering*, 50:1–27, 2001.

[93] N. Sukumar, B. Noran, and T. Belytschko. The natural element method in solid mechanics. *International Journal for Numerical Methods in Engineering*, 43:839–887, 1998.

[94] N. Sukumar and A. Tabarraei. Conforming polygonal fintie elemets. *International Journal for Numerical Methods in Engineering*, 61:2045–2066, 2004.

[95] J. W. Swegle, S. W. Attaway, M. W. Heinstein, F. J. Mello, and D. L. Hicks. An analysis of smoothed particle hydrodynamics. *Sandia Report SAND93-2513*, 1994.

[96] J. W. Swegle, D. L. Hicks, and S. W. Attaway. Smoothed particle hydrodynamics stability analysis. *Journal of Computational Physics*, 116:123–134, 1995.

[97] R. L. Taylor, J. C. Simo, O. C. Zienkiewicz, and A. C. H. Chan. The patch test: A condition for assessing FEM convergence. *International Journal for Numerical Methods in Engineering*, 22:39–62, 1986.

[98] A. N. Tichonov and A. A. Samarskii. Homogeneous difference schemes on nonuniform nets. *Computational Mathematics and Mathematical Physics*, 2:812–832, 1962.

[99] S. P. Timoshenko and J. N. Goodier. *Theory of Elasticity*. McGraw Hill, New York, 3rd edition, 1987.

[100] E. Tonti. A direct discrete formulation of field laws: The cell method. *Computer Modeling in Engineering and Sciences*, 2:237–258, 2001.

[101] B. van Rietbergen, H. Weinans, R. Huiskes, and A. Odgaard. A new method to determine trabecular bone elastic properties and loading using micromechanical finite-element models. *Jouranl of Biomechanics*, 28:69–81, 1995.

[102] V. Vidal, J. Bonet, and A. Huerta. Stablized updated lagrangian corrected sph for explicit dynamics problems. *International Journal for Numerical Methods in Engineering*, 69:2687–2710, 2007.

[103] J. W. Yoo, B. Moran, and J. S. Chen. Stabilized conforming nodal integration in the natural-element method. *International Journal for Numerical Methods in Engineering*, 60:861–890, 2004.