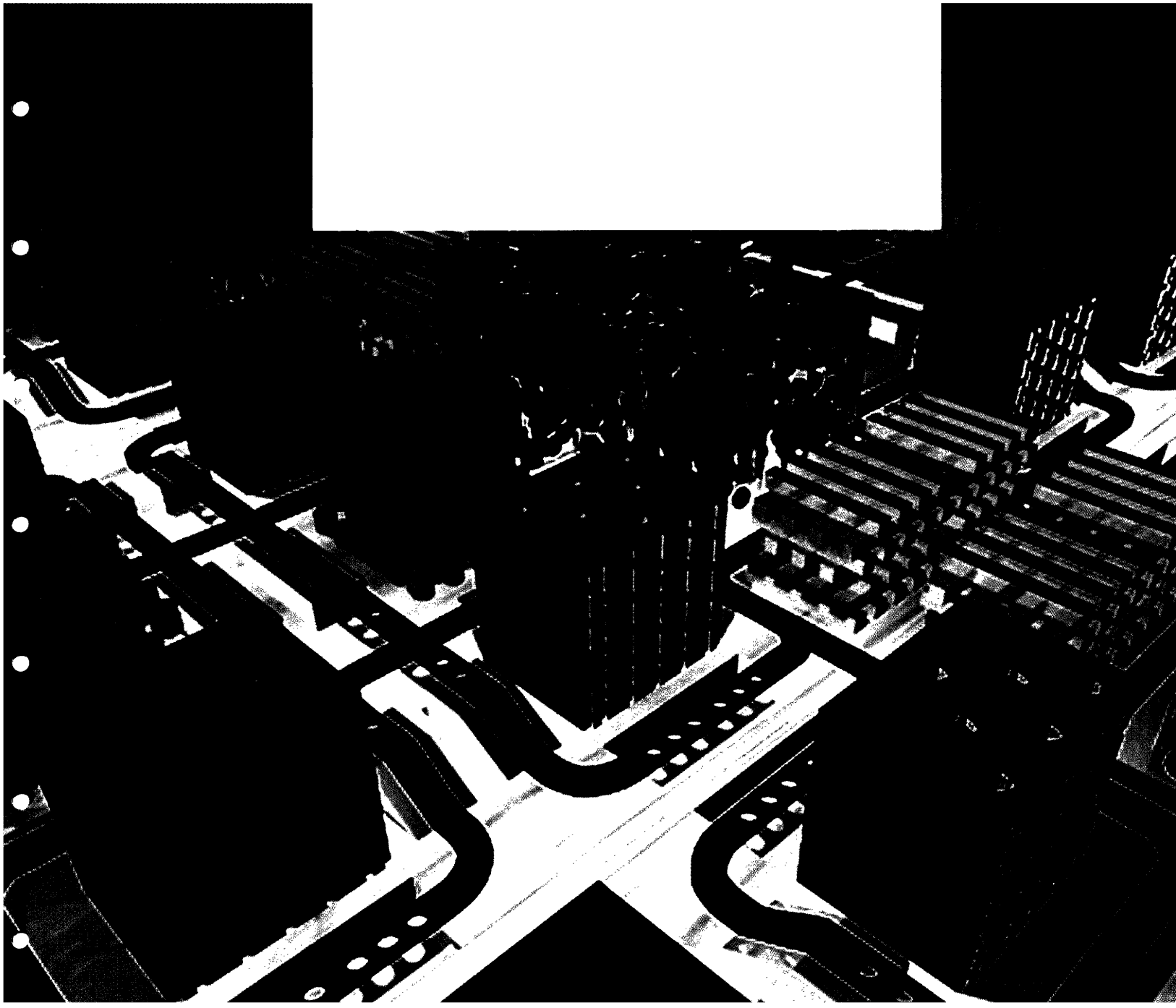
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

The RESEARCH LABORATORY *of* ELECTRONICS



On the Development of an Efficient Truly Meshless Discretization Procedure in Computational Mechanics

By: Suvranu De, K. J. Bathe
and Mandayam A. Srinivasan

RLE Technical Report No. 650
February 2001

On the Development of an Efficient Truly Meshless Discretization Procedure in Computational Mechanics

By: Suvranu De, K. J. Bathe
and Mandayam A. Srinivasan

RLE Technical Report No. 650

February 2001

On the Development of an Efficient Truly Meshless Discretization Procedure in Computational Mechanics

by

Suvranu De

B.E., Mechanical Engineering, Jadavpur University, Calcutta, India (1993)
M.E., Mechanical Engineering, Indian Institute of Science, Bangalore, India (1995)

Submitted to the Department of Mechanical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Science in Mechanical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2001

Signature of Author .....
Department of Mechanical Engineering
28 September 2000

Certified by .....
Mandayam A. Srinivasan
Principal Research Scientist
Thesis Supervisor

Certified by .....
Klaus-Jürgen Bathe
Professor
Thesis Supervisor

Accepted by .....
Ain A. Sonin
Chairperson, Department Committee on Graduate Students

On the Development of an Efficient Truly Meshless Discretization Procedure in Computational Mechanics

by

Suvranu De

Submitted to the Department of Mechanical Engineering
on 28 September 2000, in partial fulfillment of the
requirements for the degree of
Doctor of Science in Mechanical Engineering

Abstract

The objective of this thesis is to present an efficient and reliable meshless computational technique – the method of finite spheres – for the solution of boundary value problems on complex domains. This method is truly meshless in the sense that the approximation spaces are generated and the numerical integration is performed without a mesh.

While the theory behind meshless techniques is rather straightforward, the generation of a computationally efficient scheme is quite difficult. Computational efficiency may be achieved by proper choice of the interpolation functions, effective ways of incorporating the essential boundary conditions and efficient and specialized numerical integration rules.

The pure displacement formulation is observed to exhibit volumetric "locking" during incompressible (or nearly incompressible) analysis. A displacement/pressure mixed formulation is developed to overcome this problem. The stability and optimality of the mixed formulation are tested using numerical inf-sup tests for a variety of discretization schemes.

Solutions to several example problems are presented showing the application of the method of finite spheres to problems in solid and fluid mechanics. A very specialized application of the technique to physically based real time medical simulations in multimodal virtual environments is also presented.

In the current form of implementation, the method of finite spheres is about five times slower than the finite element techniques for problems in two-dimensional elastostatics.

Thesis Supervisor: Mandayam A. Srinivasan
Title: Principal Research Scientist

Thesis Supervisor: Klaus-Jürgen Bathe
Title: Professor

Acknowledgments

I would like to take this opportunity to thank my advisors Professor Klaus-Jürgen Bathe and Dr. M. A. Srinivasan for their constant support and encouragement.

Special thanks are due to Professor Bathe for guiding me through this research work and for his enthusiasm to push the state-of-the-art of numerical techniques. As an undergraduate student I was introduced to the fascinating world of finite element techniques through his well known text book on the topic. It was a great privilege to be able to work with him so closely for the past few years.

Personal thanks are due to Dr. Srinivasan for getting me to M.I.T and thus starting my career and supporting me through out this work. His care, compassion and enthusiasm have always been the wind beneath my wings.

I have had the privilege of working closely with two different research groups; the Touch Lab and the Finite Element Research Group. Thanks are due to all past and present members of these two labs for providing me a stimulating and exciting atmosphere to work in. Special thanks to Jung Kim for working closely with me in developing the surgical simulation routines and generating some of the figures in Chapter 8 of this thesis.

Thank you also to all my friends who have always been there for me.

I consider myself extremely fortunate to be blessed with a supportive and loving family. I am truly grateful to my parents for being with me every moment of the years that I have been away from hearth and home.

Finally, I would like to add that it has been a great experience working at M.I.T and I will always cherish the haloed memory of my *alma mater.*

List of symbols:

A		Interior of a set A.
∂A		Boundary of a set A.
$\overline{A}$	$=$	$A \cup \partial A$ Closure of a set A.
$supp(v)$	$=$	$\{\mathbf{x} \in X;\, v(\mathbf{x}) \neq 0\}$ Support of a function v.
Ω		Open bounded domain in R^d, $d = 1, 2, 3$.
Γ	$=$	$\partial\Omega$ Boundary of Ω (assumed to be Lipschitz continuous).
$\mathbf{n}$		Outward unit normal defined on Γ.
$B(\mathbf{x}_I, r_I)$	$=$	$\{\mathbf{x} \in X;\, \|\mathbf{x} - \mathbf{x}_I\|_0 < r_I\}$ Open sphere of radius r_I centered at $\mathbf{x}_I$ in d-dimensional Euclidean space ($d = 1, 2, 3$).
$S(\mathbf{x}_I, r_I)$	$=$	$\{\mathbf{x} \in X;\, \|\mathbf{x} - \mathbf{x}_I\|_0 = r_I\}$ Surface of the d-dimensional sphere of radius r_I centered at $\mathbf{x}_I$.
h		a global measure of the support radii (determined by the values of r_I).
$h_{Im}(\mathbf{x})$		The m^{th} shape function at node I.
$C^m(A)$		Space of functions with continuous derivatives upto order m on a subset A of R^d.
$C_0^m(A)$	$=$	$\{v \mid v \in C^m(A);\, supp(v)$ is a compact subset of $A\}$.
$Q_m(A)$		Space of polynomials of degree $\leq m$ defined on A.

Contents

List of Figures

List of Tables

Chapter 1

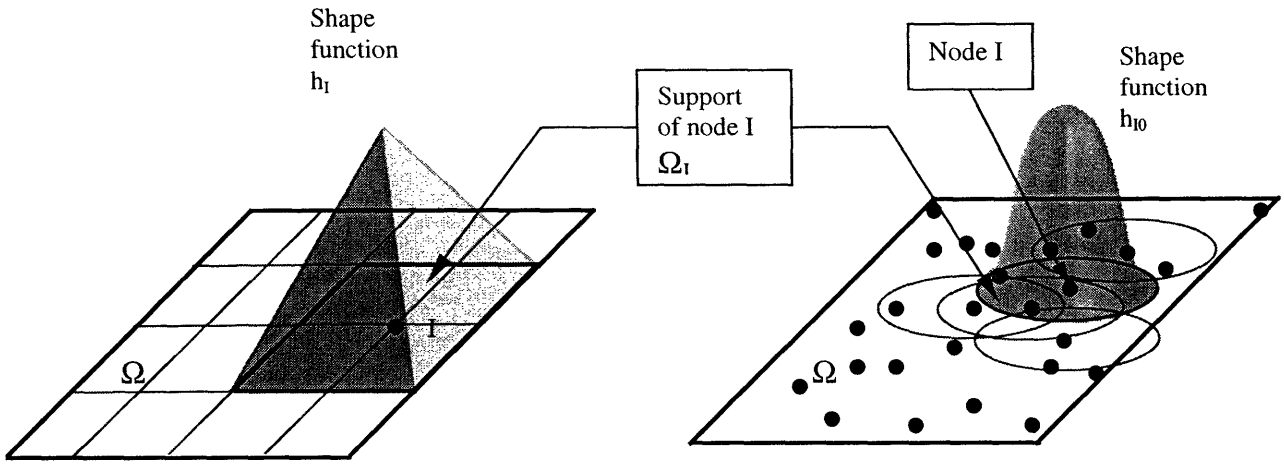
Introduction

1.1 Overview

Over the past few decades the finite element/finite volume techniques have developed into effective and powerful numerical tools for the solution of a wide variety of boundary value problems on domains having complex geometries [1]. However, even though the underlying theory is well understood, these techniques still suffer from certain drawbacks when used for the analysis of problems of practical engineering interest.

The mathematical principle behind the finite element methods is the classical weighted residual technique where the solution of a boundary value problem is sought in a finite dimensional space of trial functions by requiring the solution error to be orthogonal to a set of test functions (which may or may not coincide with the trial functions). The novelty of the finite element idea, as also the primary reason of its success, is however, the choice of the trial functions as piecewise polynomials.

To generate piecewise polynomial approximations on a geometrically complex domain, the domain is subdivided into a mesh of smaller subdomains called "elements" which have simple geometries. The elements are interconnected through special points called "nodes" (see figure 1-1(a)). Within each element the trial functions have a simple form – usually lower order polynomials. This simplifies the computations substantially. The imposition of boundary conditions along geometrically complex boundaries also becomes straightforward.



(a) Finite element method (b) Method of finite spheres

Figure 1-1: Discretization of a domain Ω in R^2 by the finite element method (a) and the method of finite spheres (b). In (a) the domain is discretized by quadrilateral elements with a node at each vertex point. The finite element shape function h_I is shown at node I. In (b) the domain is discretized using a set of nodes only. Corresponding to each node I, there is a sphere (i.e. a disk in R^2), centered at the node, which is the support of a set of shape functions corresponding to that node. One such shape function h_{I0} is shown in the figure.

While the mesh is essential for the generation of the trial functions and numerical integration of the Galerkin weak forms in the finite element techniques the automatic generation of a good quality mesh for complex geometries of practical interest presents significant difficulties (especially in three dimensions). The elements in a "good" mesh need to satisfy certain aspect ratio and included angle criteria. In the isoparametric formulation, for example, the element shape functions are expressed in terms of the natural coordinates $\mathbf{r} = (r, s, t)$ and are mapped to the local coordinates of the element $\mathbf{x} = (x, y, z)$. Hence, the spatial gradients with respect to the local coordinates, $\partial/\partial\mathbf{x}$, are related to the spatial gradients with respect to the natural coordinates, $\partial/\partial\mathbf{r}$, by

$$\frac{\partial}{\partial \mathbf{x}} = \mathbf{J}^{-1} \frac{\partial}{\partial \mathbf{r}}$$

where $\mathbf{J}$ is the Jacobian matrix

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial r} & \frac{\partial y}{\partial r} & \frac{\partial z}{\partial r} \\ \frac{\partial x}{\partial s} & \frac{\partial y}{\partial s} & \frac{\partial z}{\partial s} \\ \frac{\partial x}{\partial t} & \frac{\partial y}{\partial t} & \frac{\partial z}{\partial t} \end{bmatrix}.$$

$\mathbf{J}^{-1}$ exists if there is a one-to-one mapping between the natural and local coordinates of the element.

To ensure the invertibility of the Jacobian matrix, the element should not be too distorted or fold back on itself [1]. For example, triangular elements with very small included angles ("slivers") and nonconvex polygons need to be avoided (see figure 1-2). This makes mesh generation an extremely time consuming preprocessing stage for most problems of industrial importance. Once a good quality mesh has been generated, however, the solution time for the same problem on a reasonably fast workstation may require only a small fraction of the preprocessing time.

While mesh generation is itself a time consuming process, the modeling and analysis of certain types of problems like dynamic crack growth and machining may require frequent remeshing of the analysis domain as the existing mesh gets increasingly distorted or there is change in topology of the mesh with progress in loading. For the analysis of such problems it would be certainly beneficial if no mesh were required. This is the idea behind the development of the so-called "meshless" techniques; numerical schemes that do not require a mesh.

A number of meshless techniques have been proposed so far. Some of the earliest ones are the generalized finite difference techniques [2, 3, 4, 5] and smoothed particle hydrodynamics (SPH) method [6, 7, 8, 9]. More recent ones include the diffuse element method (DEM) [10], the element free Galerkin (EFG) method [11, 12] , the reproducing kernel particle method (RKPM) [13], the moving least-squares reproducing kernel method (MLSRK) [14, 15] the partition of unity finite element method (PUFEM) [16, 17, 18, 19] , the hp-clouds method [20, 21], the reproducing kernel hierarchical partition of unity method [22, 23], the finite point method [24], the local boundary integral equation (LBIE) method [25] and the mesh-

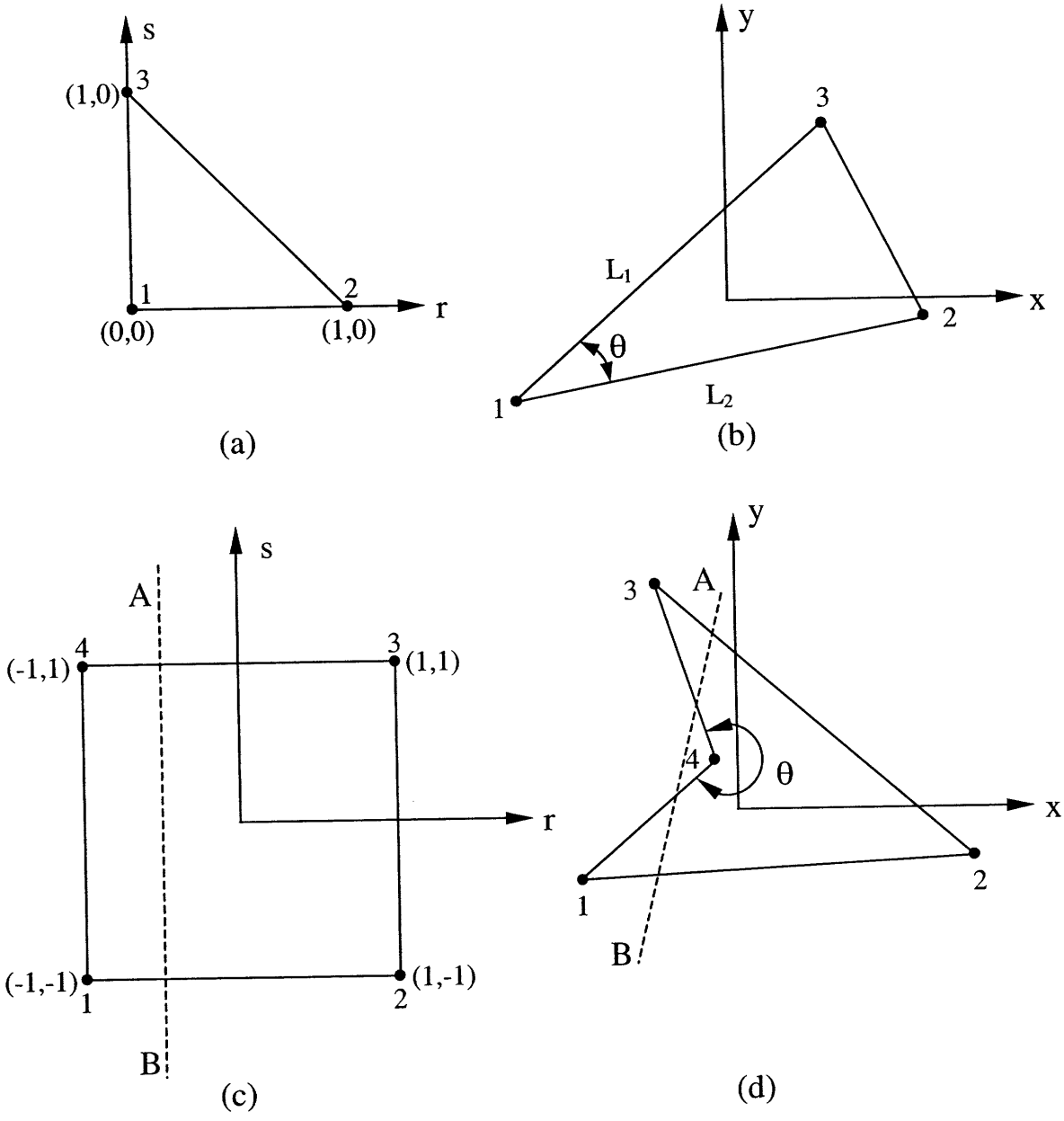


Figure 1-2: Isoparametric finite elements. A triangular element is shown in natural coordinates, see (a), and local coordinates, see (b). det(**J**)= $L_1 L_2 sin\theta$, where L_1 and L_2 are the lengths of the sides $\bar{13}$ and $\bar{12}$ of triangle 123. Small θ results in "bad" elements called "slivers". A 4-noded quadrilateral element is shown in natural (see (c)) and local (see(b)) coordinates. The mapping is not one-to-one since a segment of line AB that was inside the square in (c) is outside the quadrilateral in (d). This occurs when the included angle $\theta > \pi$.

less local Petrov-Galerkin (MLPG) method [26, 27, 28, 29].

The established computational techniques based on the weighted residual scheme have three key ingredients:
Interpolation: An expansion of the unknown field variable/s in terms of trial basis/shape functions and unknown parameters,
Integration: The determination of the governing algebraic equations by setting the residual error orthogonal to a set of test functions which may or may not coincide with the trial functions, and
Solution of the algebraic equations: The solution of the governing equations for the unknown parameters.
If the first two steps can be performed without a mesh, then what results is a "truly" meshless method. Many of the early "meshless" techniques such as the DEM, EFGM, hp-clouds method etc. are not truly meshless since even if the interpolation is independent of a background mesh, the integration is not.

References [30] and [31] present a survey of meshless techniques. The major meshless methods described in these two review works, namely the RKPM, SPH method, DEM and EFG method are based on two classes of interpolation functions: moving least squares functions (used in the DEM and the EFG method), and the partition of unity (PU) or hp-clouds functions (used in the PUFEM and hp-clouds methods). All these methods are really "pseudo meshless" since they use a background mesh for the numerical integration (and sometimes even for imposing the Dirichlet boundary conditions).

The finite point method [24] is a truly meshless scheme. The method uses a weighted least squares (WLS) interpolation and point collocation, thus bypassing integration. However, the weighted least squares technique generates multivalued approximation functions and methods based on point collocation are notorious for the sensitivity of the solution on the choice of "proper" collocation points.

As a truly meshless technique, the meshless local Petrov-Galerkin (MLPG) method [29] seems to be the most promising. The technique is based on a weak form computed over a

local sub-domain, which can be any simple geometry like a sphere, cube or an ellipsoid for ease of integration. The trial and test function spaces can be different or may be the same. Any class of functions with compact support satisfying certain approximation properties (like the MLS functions or PU functions) can be used as trial and test functions [26]. This method has been successfully applied to a wide range of problems [26, 27, 28, 29] and is of very general nature. A method using a similar approach but boundary integral techniques is the local boundary integral equation (LBIE) method [25].

However, although considerable efforts have been made in the development of meshless methods, the currently available techniques are still computationally much less efficient than the well established finite element procedures. The primary reason is that complicated (non-polynomial) shape functions are employed and the required numerical integration is very difficult to perform efficiently. Hence some researchers (refer to [32], [33]) have reverted back to developing finite element techniques incorporating certain aspects of the meshless methods. However, in this work we focus attention on truly meshless techniques.

Computational efficiency and reliability are clearly the most important issues for the eventual success of a meshless technique. Computational efficiency is achieved by the proper choice of geometric sub-domains, test and trial function spaces, numerical integration techniques and procedures for imposing the essential boundary conditions. With these issues in mind we have developed the method of finite spheres [34] as a specific implementation of the meshless discretization methodology. In this dissertation we discuss all these issues in the context of the method of finite spheres.

The method of finite spheres is a truly meshless technique. The discretization is performed using functions that are compactly supported on general d-dimensional spheres and the Galerkin weak form of the governing partial differential equations is integrated using specialized numerical integration rules. The primary reason for the selection of spherical support being that the relative orientation and the region of overlap of two spheres are completely determined by the coordinates of their centres and their radii.

In the traditional finite element/finite volume methods, the interpolation functions are

piece-wise continuous polynomials (or mapped polynomials) which are truly interpolatory; i.e. they satisfy the Kronecker delta property. This ensures that Dirichlet boundary conditions are enforced rather simply. Moreover, numerical integration is performed most efficiently using Gauss-Legendre product rules on integration domains that are d-dimensional cubes or tetrahedra. The Gauss-Legendre quadrature rule ensures arbitrary polynomial accuracy and therefore the stiffness terms (for undistorted elements) are exactly integrated with minimal cost [1].

In the method of finite spheres, however, the interpolation functions do not satisfy the Kronecker delta property at the nodes. Hence, efficient imposition of Dirichlet boundary conditions is an important issue. Moreover, the interpolation functions are rational (non-polynomial) functions and the integration domains are spheres, spherical shells or general sectors. Hence effective numerical integration rules have to be developed and exact integration can never be achieved.

1.2 Thesis outline

A brief outline of this dissertation is as follows. In chapter 2 we discuss in detail our justification for the use of PU basis functions based on Shepard partitions of unity and summarize key results relating to consistency and *a-priori* error analysis. In chapter 3 we derive the weak form for a symmetric second-order differential operator. We also discuss the efficient imposition of Neumann and Dirichlet boundary conditions in the absence of the Kronecker delta property. Even though we primarily concentrate on the d-dimensional sphere as our integration domain, we realize that to deal with doubly-connected domains efficiently, the ideas of support and integration domain have to be decoupled. To address the solution of such problems we present our developments using d-dimensional spherical shells.

In chapter 4 we deal with issues related to numerical integration on the d-dimensional spheres and spherical shells. In chapter 5 we apply the method of finite spheres to the solution of problems in linear elastostatics. We observe that while for the analysis of compressible media the rate of convergence is quite high, the pure displacement-based formulation

"locks" when the Poisson's ratio, $\nu \rightarrow 0.5$. In chapter 6 we propose a displacement/pressure mixed formulation as a remedy to the problem of volumetric locking. We analyze the stability and optimality of the mixed discretization schemes using numerical inf-sup tests. In chapter 7 we discuss the computational issues in the method of finite spheres and compare with the traditional finite element techniques and other meshless techniques based on the moving least squares interpolation functions.

In chapter 8 we present a very special application of the method of finite spheres to a problem in surgical simulation in multimodal virtual environments. We discuss how a reasonably accurate physically based real time haptic and graphical rendering technique for deformable objects may be obtained when the point collocation method is used as the weighted residual scheme. Finally, in chapter 9 we present a summary of the major conclusions, contributions of the present work and directions for future study.

Chapter 2

The approximation scheme

The first step in the Galerkin procedure is to construct finite dimensional subspaces of a Sobolev space, in which the weak solution is assumed to exist. We list the desirable properties of these trial function spaces in section 2.1. In principle multiple choices exist for the construction of the approximation spaces. The choice of a particular family of spaces, however, plays a key role in deciding the overall computational efficiency of the resulting numerical scheme. For example, the functions and their derivatives should be relatively inexpensive to compute. More importantly, numerical integration should not be computationally expensive. As we have already mentioned, the simplicity in the choice of the trial function spaces as piece-wise polynomials is one of the primary reasons for the success of the finite element technique.

In the method of finite spheres we have chosen the partitions of unity paradigm [17] of constructing the trial function spaces using the Shepard partitions of unity functions [35]. Our choice of the particular partitions of unity functions and the local approximation spaces is based on the consideration of computational efficiency without sacrificing solution accuracy. We discuss the partition of unity paradigm in section 2.2 and the approximation properties of the trial functions in section 2.3. In section 2.4 we provide guidelines for choosing the functional form of the partition of unity functions.

2.1 Desirable properties of the approximation spaces

Our aim is to generate approximation spaces which satisfy the following minimal requirements:

Consistency or polynomial reproducing property: The consistency condition is related to the degree of the governing partial differential equation. For example, when solving an elasticity problem using a displacement-based formulation, the approximating functions should not only be able to reproduce constant functions (the so called "rigid body" modes) but also linear functions (the "constant strain" states), i.e., we look for at least first-order consistency. Thus, we should be able to reproduce polynomials to a certain order to satisfy the consistency requirement.

Local approximability: This is a more general requirement than just consistency and is related to the reproducing properties of the trial functions. If we know the nature of the solution in certain subdomains of Ω, we should be able to incorporate specific functions in the global approximation space in order to enrich this space to closely represent the solution. There are certain situations where singularities arise naturally in the solution of the governing differential equations and polynomials perform poorly in resolving such singularities. The idea is to use analytically available solutions to improve numerical predictions.

Continuity: The approximation functions should satisfy certain minimal continuity conditions.

Localization by compact support: The main advantages of using compactly supported functions, i.e. functions that are nonzero only on small subsets of Ω, are that (1) they allow the localization of the approximation (and hence steep gradients can be handled by using more functions locally), (2) they result in banded system matrices since only a few of the supports overlap at any given point of the domain and (3) they allow a natural means of controlling the rate of convergence of numerical schemes through h, p or hp-type refinements.

There is no unique way of constructing the approximation spaces. The current interest in

the so-called meshless methods has been primarily spurred by the ability to construct non-polynomial approximation spaces with compact support without the need for a background mesh. As we mentioned already, examples of such approximation functions are compactly supported wavelet functions, the MLS (moving least squares) functions [10, 11], the partitions of unity functions of Babuška and Melenk [17] and the hp-cloud functions of Duarte and Oden [20, 21]. In the wavelet-based methods, compactly supported functions with desirable properties are developed using FIR (finite impulse response) filter-banks. The difficulties of using wavelets as basis functions are that they are designed to have desirable orthogonality properties in $L^2(\Omega)$ but not in higher-order Hilbert spaces, the computation of inner products through "connection coefficients" is very cumbersome and the application of wavelets to arbitrary domains is still being researched.

The moving least squares technique of generating compactly supported functions having desirable reproducing properties is quite appealing, but has some drawbacks, most important of which is the need to invert a $n \times n$ matrix (when n basis functions are used to generate the MLS shape functions). Besides increasing the computational cost for any $n > 1$, this requirement means that at each evaluation point at least n weight functions should be nonzero for the matrix to be invertible.

Of the existing techniques for the generation of compactly supported basis functions, the methods developed using the partition of unity (PU) paradigm appear to be the most general. They possess all the desirable properties we have listed above and as we shall see, it is possible to generate low-cost partitions of unity.

2.2 The shape functions

In this section we discuss the generation of global approximation spaces using the PU paradigm. Let $\Omega \in R^d$ ($d = 1, 2$ or 3) be an open bounded domain and let Γ be its boundary (see figure 2-1). Let a family of open spheres $\{B(\mathbf{x}_I, r_I); I = 1, 2, \cdots, N\}$ form a covering for Ω, i.e., $\Omega \subset \bigcup_{I=1}^{N} B(\mathbf{x}_I, r_I)$, where $\mathbf{x}_I$ and r_I refer to the center and radius of the I^{th} sphere respectively. We associate a "node" with the geometric center of each sphere.

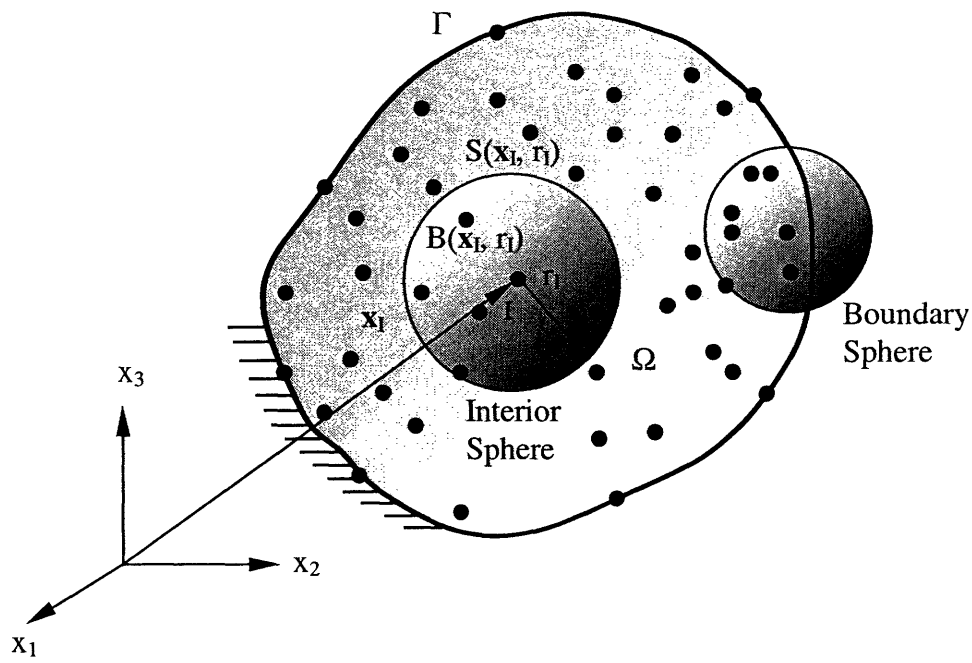


Figure 2-1: General three-dimensional body, Ω, discretized using a set of nodes. Associated with each node I is a sphere $B(\mathbf{x}_I, r_I)$. Spheres that lie completely inside the domain are called "interior spheres" while those which intersect the boundary of the domain, S, are called "boundary spheres".

By $S(\mathbf{x}_I, r_I)$ we denote the surface of the I^{th} sphere. The spheres may be entirely within the domain (interior spheres) or may have nonzero intercepts with the boundary (boundary spheres), see figure 2-1.

2.2.1 The Shepard functions

The first step in the PU paradigm of generating global approximation spaces is of course the generation of the partition of unity functions.

Definition 2.1 There exists a system of functions $\{\varphi_I\}_{I=1}^N$ such that

1. $\sum_{I=1}^N \varphi_I(\mathbf{x}) = \mathbf{1} \quad \forall \mathbf{x} \in \Omega$.
2. $supp(\varphi_I(\mathbf{x})) \subset B(\mathbf{x}_I, r_I)$.
3. $\varphi_I(\mathbf{x}) \in C_0^s(R^n), s \geq 0$.

This system of functions $\{\varphi_I\}_{I=1}^N$ is defined as a **partition of unity** subordinate to the open cover $\{B(\mathbf{x}_I, r_I)\}$ [36].

As an example, let us consider R^1 where Ω is a line segment. Let us consider a regular arrangement of nodes with internodal spacing 'h'. Then the usual piecewise linear "hat" functions defined by

$$\varphi(x) = \begin{cases} 1 + \frac{x}{h} & \text{for } x \in (-h, 0] \\ 1 - \frac{x}{h} & \text{for } x \in (0, h) \\ 0 & \text{elsewhere} \end{cases} \tag{2.1}$$

form a partition of unity

$$\varphi_I(x) = \varphi(x - x_I)$$

subordinate to the open cover $\{(x_I - h, x_I + h)\}$.

There is, however, a general technique of generating partitions of unity on a complex domain with a general covering. We define a radial weighting function (or window function) $W_I(\mathbf{x})$ compactly supported on the sphere centered at node I such that

1. $W_I(\mathbf{x}) \in C_0^s(B(\mathbf{x}_I, r_I))$, $s \geq 0$
2. $supp(W_I) \subset B(\mathbf{x}_I, r_I)$
3. $W_I(\mathbf{x}) \geq 0 \ \forall \mathbf{x} \in \Omega$

The functions

$$\varphi_I^0(\mathbf{x}) = \frac{W_I}{\sum_{J=1}^N W_J}. \tag{2.2}$$

form a partition of unity subordinate to the open cover $\{B(\mathbf{x}_I, r_I)\}$. These functions are known as the Shepard functions [35]. This technique is very intuitive as it generates the partition of unity by a simple "normalization" procedure. The window functions provide compact support as well as the smoothness properties to the functions $\varphi_I^0(\mathbf{x})$. Notice that even though formally the sum in the denominator of equation (2.2) runs over all the nodes, only those nodes with $W_J(\mathbf{x}) \neq 0$ are actually considered. An important observation is that, unlike the "hat" functions in equation (2.1), the Shepard functions do not require a

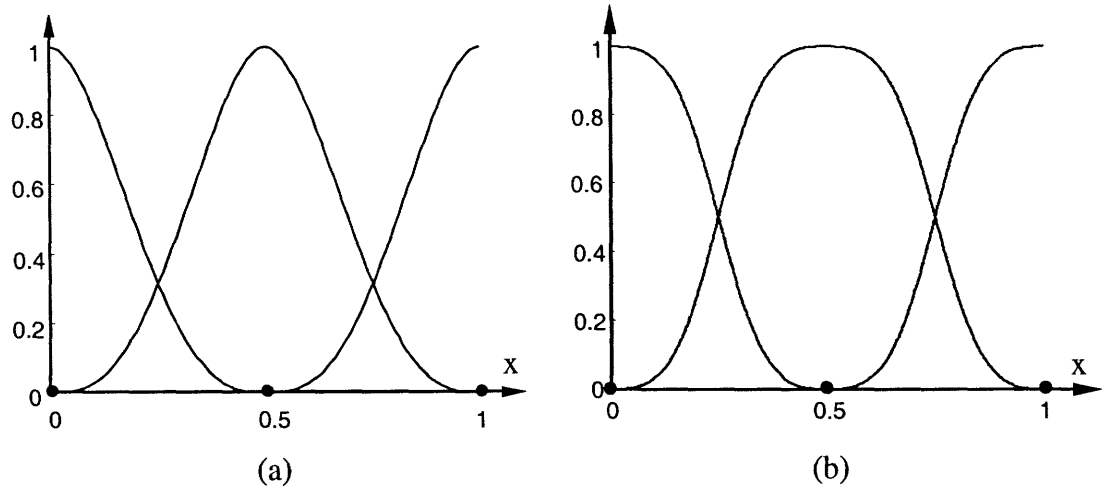


Figure 2-2: Cubic spline weighting functions are shown in (a) on [0,1]. The line is discretized using three nodes. The spheres reduce to line segments of length 1.0. Shepard functions generated using these weighting function are shown in (b).

mesh for their generation and provide a low cost partition of unity for even an arbitrarily scattered set of nodes on a domain. In the method of finite spheres we employ the Shepard partition of unity functions.

Important consideration should be given to the choice of the functions $W_I(\mathbf{x})$ so that low cost partitions of unity are obtained. We concentrate on radial weight functions of the form $W_I(\mathbf{x}) = W(s_I)$ (by an abuse of notation) where $s_I = \frac{\|\mathbf{x}-\mathbf{x}_I\|_0}{r_I}$ and choose a cubic spline weight function of the following form:

$$W(s_I) = \begin{cases} \frac{2}{3} - 4s_I^2 + 4s_I^3 & 0 \leq s_I < \frac{1}{2} \\ \frac{4}{3} - 4s_I + 4s_I^2 - \frac{4}{3}s_I^3 & \frac{1}{2} < s_I \leq 1 \\ 0 & s_I > 1 \end{cases} . \tag{2.3}$$

The justification for this choice will be given in section 2.4. In figure 2-2(b) we show the Shepard partitions of unity functions generated using the cubic spline weight functions (figure 2-2(a)) on a line segment.

2.2.2 The global approximation spaces

The functions $\{\varphi_I^0(\mathbf{x})\}$ satisfy zeroth order consistency, i.e. they ensure that rigid body modes are exactly represented. To attain higher order consistency, at each node I, a local approximation space $V_I^h = span_{m\in\mathcal{I}}\{p_m(\mathbf{x})\}$ is defined, where $p_m(\mathbf{x})$ is a polynomial or other function and $\mathcal{I}$ is an index set (e.g. $V_I^h = span\{1, x, y\}\ \forall I$ provides linear consistency in R^2). The superscript h is a measure of the size of the spheres.

The global approximation space V_h is generated by multiplying the partition of unity function at each node I with the functions from the local basis

$$V_h = \sum_{I=1}^{N} \varphi_I^0 V_I^h. \tag{2.4}$$

Since $V_I^h = span_{m\in\mathcal{I}}(p_m(\mathbf{x}))$, any function $v_I^h \in V_I^h$ can be expressed as $v_I^h(\mathbf{x}) = \sum_{m\in\mathcal{I}} p_m(\mathbf{x})\alpha_{Im}$, for $\alpha_{Im} \in R$. If we multiply each $p_m(\mathbf{x})$ by $\varphi_I^0(\mathbf{x})$, the resulting function has the same support as $\varphi_I^0(\mathbf{x})$. The global approximation space is constructed using such products. Hence, any function $v_h \in V_h$ can now be written as

$$v_h(\mathbf{x}) = \sum_{I=1}^{N} \sum_{m\in\mathcal{I}} h_{Im}(\mathbf{x})\alpha_{Im} \tag{2.5}$$

where

$$h_{Im}(\mathbf{x}) = \varphi_I^0(\mathbf{x})p_m(\mathbf{x}) \tag{2.6}$$

is a basis/shape function associated with the m^{th} degree of freedom α_{Im} of node I.

In figure 2-3(a) a Shepard partition of unity function is shown at a node I on a square domain in R^2. If we multiply this function with $\frac{(x-x_I)}{r_I}$ and $\frac{(y-y_I)}{r_I}$ (r_I is the radius of the sphere with center (x_I,y_I) at node I), we generate two more shape functions h_{I1} and h_{I2} at the same node with the same support.

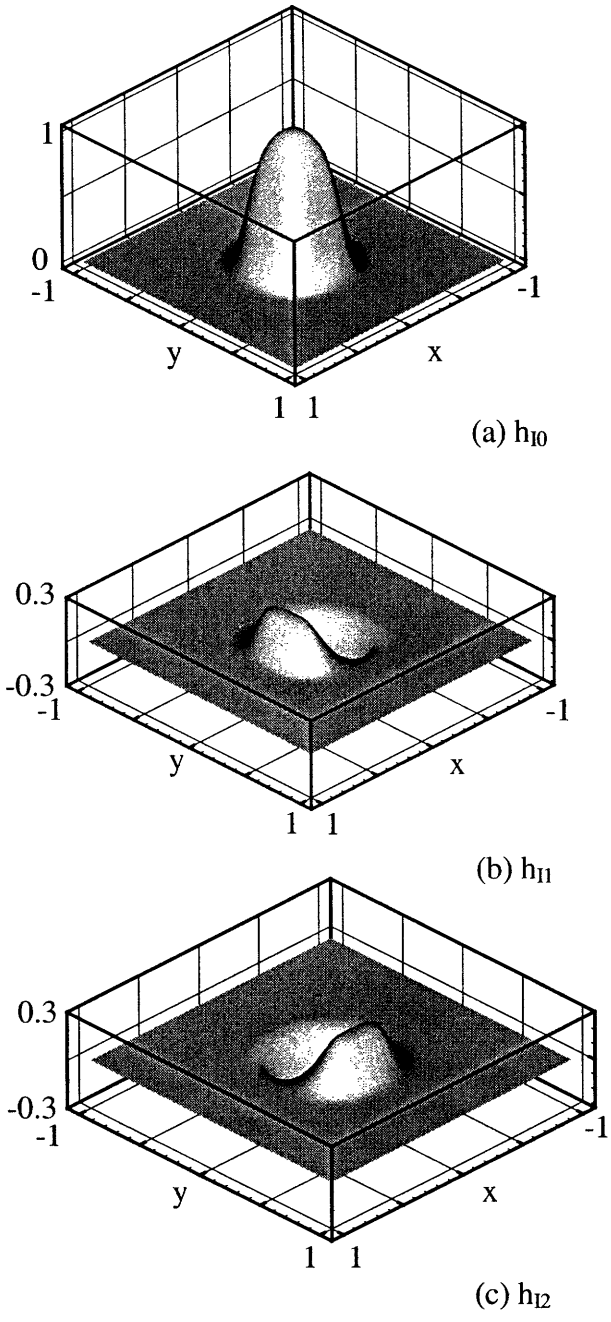


Figure 2-3: Three shape functions (h_{I0}, h_{I1} and h_{I2}) at an interior node are shown. h_{I0} is the Shepard function at the node, while $h_{I1}=\frac{(x-x_I)}{r_I}h_{I0}$ and $h_{I2}=\frac{(y-y_I)}{r_I}h_{I0}$.

2.3 Some properties of the approximation spaces

We now state and prove some important properties of the discretization scheme (see also [17] and [20]). We use the symbol C to denote a generic positive constant which may take different values at successive occurrences (including in the same equation).

Theorem 2.1 Reproducing property: *If any function $p_k(\mathbf{x})$ $(k \in \mathcal{I})$ is included in the local basis of each node, it is possible to exactly reproduce it on the entire domain.*

Proof: Using equations (2.5) and (2.6) we may write

$$v_h(\mathbf{x}) = \sum_{I=1}^{N} \sum_{m \in \mathcal{I}} \varphi_I^0(\mathbf{x}) p_m(\mathbf{x}) \alpha_{Im}. \tag{2.7}$$

If we choose $\alpha_{Im} = \delta_{mk} \, \forall \, I$, where δ_{mk} is the Kronecker delta defined as

$$\delta_{mk} = \begin{cases} 1 & \text{if } m = k \\ 0 & \text{if } m \neq k \end{cases} \tag{2.8}$$

then

$$v_h(\mathbf{x}) = \sum_{I=1}^{N} \varphi_I^0(\mathbf{x}) \sum_{m \in \mathcal{I}} p_m(\mathbf{x}) \delta_{mk} = p_k(\mathbf{x}) \tag{2.9}$$

since $\sum_{I=1}^{N} \varphi_I^0(\mathbf{x}) = 1 \forall \mathbf{x} \in \Omega$.

Corollary 2.1 Consistency: *If $Q_m \subset span(V_I^h)$ $\forall I$, then $Q_m \subset span(V_h)$.*

Theorem 2.1 states that if we include *a-priori* knowledge of the solution in local subdomains, then this knowledge will enhance the approximation capability because the functions representing this knowledge can be reproduced. Corollary 2.1 assures that it is possible to obtain any order of consistency, at least theoretically. It turns out that for our choice of the PU functions, the functions h_{Im} are linearly independent, i.e. $V_h = span(h_{Im}(\mathbf{x}))$,as long as the local bases are linearly independent.

Theorem 2.2 Continuity: *Let $W_I, I = 1, 2, \cdots, N \in C_0^s(B(\mathbf{x}_I, r_I))$ and let $p_m(\mathbf{x}) \in C^l(\Omega)$ for $s, l \geq 0$; then the shape functions $h_{Im}(\mathbf{x})$ satisfy $h_{Im}(\mathbf{x}) \in C_0^{\min(s,l)}(\Omega_I)$ where $\Omega_I = B(\mathbf{x}_I, r_I) \cap \Omega$.*

Proof: The proof is immediate from equations (2.2) and (2.6).

This theorem is used in section 2.4 to obtain a functional form of the weighting functions $W_I(\mathbf{x})$.

Theorem 2.3 Approximation error estimate: *Let u be the function to be approximated, and let the Shepard functions $\varphi_I^0(\mathbf{x})$ satisfy*

$$\|\varphi_I^0(\mathbf{x})\|_{L^\infty(R^d)} \leq C, \tag{2.10}$$

$$\|\nabla\varphi_I^0(\mathbf{x})\|_{L^\infty(R^d)} \leq \frac{C}{r_I}. \tag{2.11}$$

Assume that the local approximation spaces V_I^h have the following properties: On each patch $\Omega_I = B(\mathbf{x}_I, r_I) \cap \Omega$, u can be approximated by a function $v_I^h \in V_I^h$ such that

$$\|u - v_I^h\|_{L^2(\Omega_I)} \leq \epsilon_1(I, h, p, u), \tag{2.12}$$

$$\|\nabla(u - v_I^h)\|_{L^2(\Omega_I)} \leq \epsilon_2(I, h, p, u). \tag{2.13}$$

then there is a function $v_h \in V_h$ satisfying

$$\|u - v_h\|_{L^2(\Omega)} \leq C \left(\sum_{I=1}^{N} (\epsilon_1(I, h, p, u))^2 \right)^{1/2} \tag{2.14}$$

$$\|\nabla(u - v_h)\|_{L^2(\Omega)} \leq \left(\sum_{I=1}^{N} \left(\frac{C}{r_I} \right)^2 (\epsilon_1(I, h, p, u))^2 + C\,(\epsilon_2(I, h, p, u))^2 \right)^{1/2}. \tag{2.15}$$

Proof: Since the functions $\{\varphi_I^0(\mathbf{x})\}_{I=1}^N$ form a partition of unity,

$$\begin{aligned} (u - v_h) &= u - \textstyle\sum_{I=1}^N \varphi_I^0(\mathbf{x}) v_I^h \\ &= \textstyle\sum_{I=1}^N \varphi_I^0(\mathbf{x})(u - v_I^h). \end{aligned}$$

Therefore

$$\begin{aligned}
\|u - v_h\|^2_{L^2(\Omega)} &= \int_\Omega (u - v_h)^2 \, d\Omega \\
&= \int_\Omega \left(\sum_{I=1}^N \varphi_I^0(\mathbf{x})(u - v_I^h) \right)^2 d\Omega \\
&\leq C \sum_{I=1}^N \int_\Omega \left(\varphi_I^0(\mathbf{x})(u - v_I^h) \right)^2 d\Omega \\
&\leq C \| \varphi_I^0(\mathbf{x}) \|^2_{L^\infty} \sum_{I=1}^N \|u - v_I^h\|^2_{L^2(\Omega_I)} \\
&\leq C \sum_{I=1}^N (\epsilon_1(I,h,p,u))^2 . \qquad \text{from (2.10) and (2.12)}
\end{aligned}$$

This proves (2.14). The proof of (2.15) is similar. We notice that

$$\begin{aligned}
\nabla(u - v_h) &= \sum_{I=1}^N \nabla \left(\varphi_I^0(\mathbf{x})(u - v_I^h) \right) \\
&= \sum_{I=1}^N \nabla \varphi_I^0(\mathbf{x})(u - v_I^h) + \sum_{I=1}^N \varphi_I^0(\mathbf{x}) \nabla(u - v_I^h)
\end{aligned}$$

therefore

$$\begin{aligned}
\|\nabla(u - v_h)\|^2_{L^2(\Omega)} &= \| \sum_{I=1}^N \nabla \varphi_I^0(\mathbf{x})(u - v_I^h) + \sum_{I=1}^N \varphi_I^0(\mathbf{x}) \nabla(u - v_I^h) \|^2_{L^2(\Omega)} \\
&\leq 2 \left\{ \| \sum_{I=1}^N \nabla \varphi_I^0(\mathbf{x})(u - v_I^h) \|^2_{L^2(\Omega)} + \| \sum_{I=1}^N \varphi_I^0(\mathbf{x}) \nabla(u - v_I^h) \|^2_{L^2(\Omega)} \right\} \\
&\leq C \left\{ \sum_{I=1}^N \| \nabla \varphi_I^0(\mathbf{x})(u - v_I^h) \|^2_{L^2(\Omega)} + \sum_{I=1}^N \| \varphi_I^0(\mathbf{x}) \nabla(u - v_I^h) \|^2_{L^2(\Omega)} \right\} \\
&\leq C \left\{ \sum_{I=1}^N \| \nabla \varphi_I^0(\mathbf{x}) \|^2_{L^\infty} \, \|(u - v_I^h)\|^2_{L^2(\Omega)} + \sum_{I=1}^N \| \varphi_I^0(\mathbf{x}) \|^2_{L^\infty} \, \|\nabla(u - v_I^h)\|^2_{L^2(\Omega)} \right\} \\
&\leq \sum_{I=1}^N \left\{ \left(\frac{C}{r_I} \right)^2 (\epsilon_1(I,h,p,u))^2 + C \, (\epsilon_2(I,h,p,u))^2 \right\} \qquad \text{from (2.11) and (2.13).}
\end{aligned}$$

Theorem 2.3 is of very general nature and provides an interpolation error estimate if the local approximation behavior is known.

Theorem 2.4 Convergence rate of the h-version: *Let $u \in H^k(\Omega)$, $k \geq 2$. Let V_I^h have the following approximation properties:*

$$\epsilon_1(I,h,p,u) \leq C r_I^{\mu+1} \| u \|_{H^k(\Omega_I)}$$
$$\epsilon_2(I,h,p,u) \leq C r_I^{\mu} \| u \|_{H^k(\Omega_I)}$$

for some appropriate $\mu > 0$. If u_h is the numerical solution, then

$$\| u - u_h \|_{L^2(\Omega)} \leq C h^{\mu+1} \| u \|_{H^k(\Omega)},$$
$$\| \nabla (u - u_h) \|_{L^2(\Omega)} \leq C h^{\mu} \| u \|_{H^k(\Omega)} .$$

Proof: The proof follows directly from Theorem 2.3 and the properties of the Galerkin process (see [1] for these properties).

Theorem 2.4 is an application of the previous theorem to obtain a bound on the solution error. Specifically if a polynomial basis of degree p is used as the local approximation space, and $k = p+1$, then $\mu = p$ and an $\mathcal{O}(h^{p+1})$ convergence in the solution variable is predicted.

2.4 Choice of functions $W_I(\mathbf{x})$

In our implementation we have chosen radial weight functions of the form $W_I(\mathbf{x}) = W(s_I)$ where $s_I = \frac{\|\mathbf{x}-\mathbf{x}_I\|_0}{r_I}$. In this section we justify the choice of the cubic spline function in equation (2.3). The following two statements may be made directly as a consequence of Theorem 2.2

1. **Displacement continuity**: The displacement field is continuous so long as the functions W_I and $p_m(\mathbf{x})$ are continuous.

2. **Stress continuity:** The stress fields, obtained by differentiating the displacement field (2.5), are continuous on Ω if each of the functions W_I has zero slope at the center, $\mathbf{x}_I$, and on the surface, $S(\mathbf{x}_I, r_I)$ of the sphere on which it is defined, provided the functions $p_m(\mathbf{x})$ and their derivatives are sufficiently smooth.

For sufficiently smooth functions $p_m(\mathbf{x})$, the stress fields are continuous provided the derivatives of W_I with respect to the spatial coordinates x_i ($i \in \{1,2,3\}$)

$$\frac{\partial W(s_I)}{\partial x_i} = \frac{x_i - x_{Ii}}{r_I^2}\left[\frac{1}{s_I}\frac{dW(s_I)}{ds_I}\right] \tag{2.16}$$

are continuous in $B(\mathbf{x}_I, r_I)$ and on $S(\mathbf{x}_I, r_I)$.
This derivative exists as $s_I \to 0$ if W_I has zero slope at the center of the sphere. Moreover, the derivative in equation (2.16) is continuous on $S(\mathbf{x}_I, r_I)$, i.e. as $s_I \to 1$ if W_I has zero slope on the surface $S(\mathbf{x}_I, r_I)$.

Equation (2.16) introduces two conditions on the first derivative of the function W_I if a continuous stress field is to be obtained. A third condition arises from the constraint that the function W_I vanishes on $S(\mathbf{x}_I, r_I)$, i.e. $W(s_I = 1) = 0$. To satisfy these three conditions, the function W_I needs to be at least a cubic in s_I. In our implementation, we have therefore chosen cubic spline weighting functions.

Chapter 3

Weak forms for second order linear elliptic problems

In chapter 2 we presented a scheme of generating finite dimensional approximation spaces that are subspaces of an infinite dimensional Sobolev space. In section 3.1 of this chapter, we develop the weak form and discretized equations of the governing differential equation (whose weak solution is assumed to lie in that Sobolev space) by integrating over each d-dimensional sphere (d = 1, 2 or 3) centered around a node. We consider the example of a second-order partial differential equation in a single variable. Extension to multiple variables and higher-order differential operators can be directly achieved.

In the method of finite spheres, the approximation functions do not satisfy the Kronecker delta property at the nodes. In section 3.2 we present efficient techniques of incorporating the boundary conditions in the absence of this property. In section 3.3 we decouple the idea of integration domain and support and present a novel technique of directly discretizing general spherical cavities. Several numerical examples are presented in section 3.4.

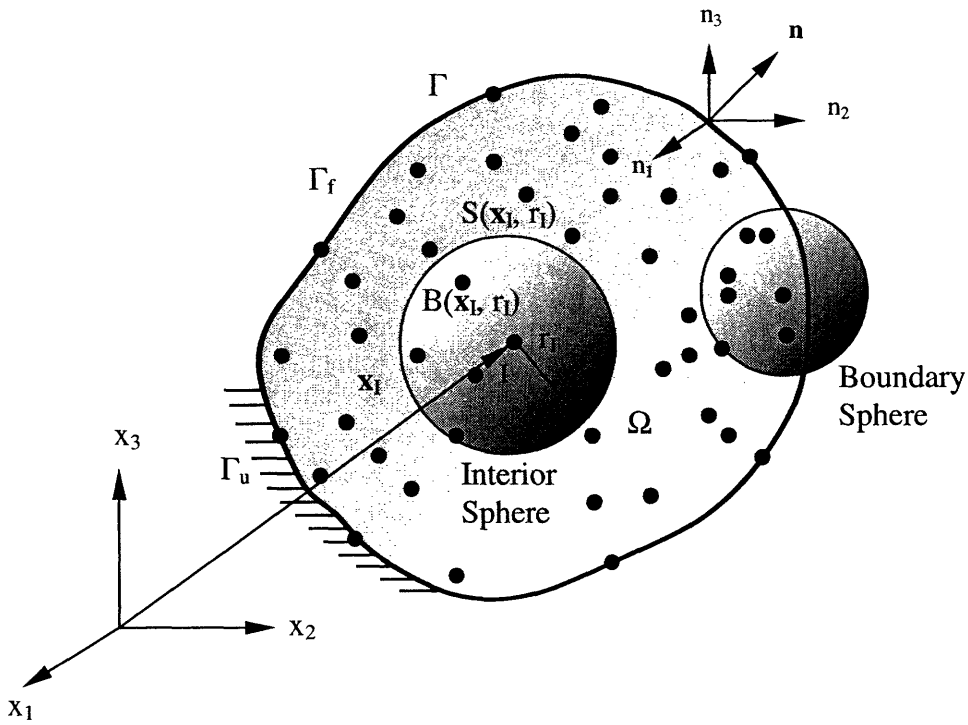


Figure 3-1: General three-dimensional body, Ω, with boundary Γ, discretized using a set of nodes. Γ_u is the portion of the boundary on which Dirichlet boundary conditions are specified whereas Γ_f is the portion of the boundary on which Neumann boundary conditions are specified. $\Gamma = \Gamma_f \cup \Gamma_u$ and $\Gamma_f \cap \Gamma_u = 0$. $\mathbf{n}$ is the outward unit normal to the boundary.

3.1 Galerkin weak form for a d-dimensional sphere

Let $\Omega \in R^d$ ($d = 1, 2$ or 3) be an open bounded domain and let Γ be its boundary (see figure 3-1). Consider the operator equation

$$Au = f \qquad \text{in } \Omega \tag{3.1}$$

where $A : \mathcal{D}_A \subset H^2(\Omega) \longmapsto L^2(\Omega)$ is a second-order symmetric positive definite differential operator with domain of definition $\mathcal{D}_A$ and $f \in L^2(\Omega)$ is the forcing function, with

$$A = -\sum_{i,j=1}^{d} \frac{\partial}{\partial x_i} a_{ij}(\mathbf{x}) \frac{\partial}{\partial x_j} + c(\mathbf{x}) \tag{3.2}$$

where d is the dimensionality of the problem, $a_{ij}(\mathbf{x})$ and $c(\mathbf{x})$ are bounded measurable coefficients. Assume that Neumann boundary conditions are prescribed over the boundary

Γ_f

$$\sum_{i,j=1}^{d} a_{ij}(\mathbf{x})\frac{\partial u}{\partial x_j}n_i = f^s \qquad \text{on } \Gamma_f \tag{3.3}$$

where n_i is the component of the outward unit normal on the boundary along the i^{th} direction (see figure 3-1), and Dirichlet boundary conditions are provided on the boundary Γ_u

$$u = u^s \qquad \text{on } \Gamma_u \tag{3.4}$$

where $\Gamma = \Gamma_u \cup \Gamma_f$ and $\Gamma_u \cap \Gamma_f = 0$. In the Bubnov-Galerkin procedure, we find the approximation $u_h \in V_h$ to the true solution u by making the residual $(Au_h - f)$ orthogonal to the basis functions $\{h_{Im}\}$. Hence, corresponding to node I, we generate the following set of equations:

$$(Au_h - f, h_{Im}) = 0, \qquad m \in \mathcal{I}.$$

Using $u_h = \sum_{J=1}^{N}\sum_{n\in\mathcal{I}} h_{Jn}(\mathbf{x})\alpha_{Jn}$, where h_{Im} s are the shape functions discussed in the previous chapter and $\alpha_{Im} \in R$, and applying Green's Theorem, we obtain the m^{th} equation corresponding to the I^{th} node as

$$\boxed{\sum_{J=1}^{N}\sum_{n\in\mathcal{I}} K_{ImJn}\alpha_{Jn} = f_{Im} + \hat{f}_{Im}} \tag{3.5}$$

where

$$K_{ImJn} = a(h_{Im}, h_{Jn}) = \int_{\Omega_I} c(\mathbf{x})h_{Im}h_{Jn}d\Omega + \sum_{i,j=1}^{d}\int_{\Omega_I} a_{ij}(\mathbf{x})\frac{\partial h_{Im}}{\partial x_i}\frac{\partial h_{Jn}}{\partial x_j}d\Omega,$$

$$f_{Im} = \int_{\Omega_I} f h_{Im}d\Omega, \tag{3.6}$$

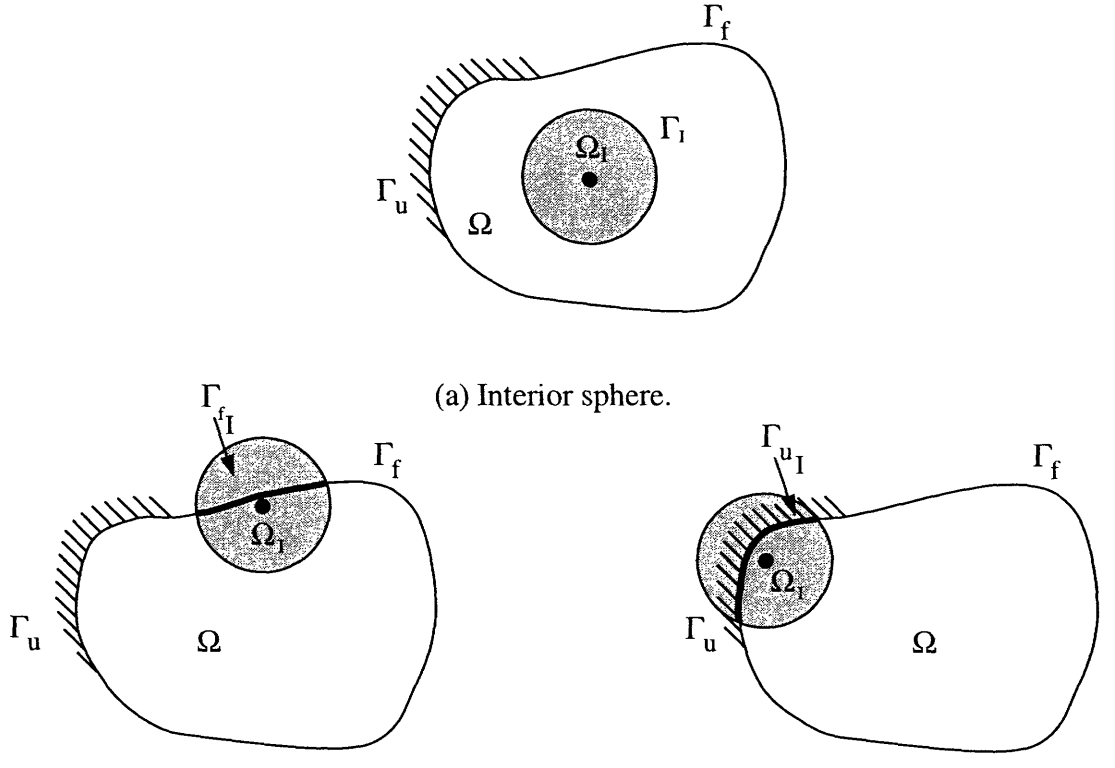
$$\hat{f}_{Im} = \sum_{i,j=1}^{d}\int_{\Gamma_I} h_{Im}n_i a_{ij}(\mathbf{x})\frac{\partial u_h}{\partial x_j}d\Gamma$$

where $\Omega_I = B(\mathbf{x}_I, r_I) \cap \Omega$ and $\Gamma_I = B(\mathbf{x}_I, r_I) \cap \Gamma$

An interior sphere has zero intercepts with the boundary, (see figure 3-2(a)). A boundary sphere has a nonzero intercept with the boundary (see figure 3-2(b) and 3-2(c)). For an

interior sphere, therefore, $\hat{f}_{Im} = 0$ due to compact support and equation (3.5) reduces to:

$$\sum_{J=1}^{N} \sum_{n \in \mathcal{I}} K_{ImJn} \alpha_{Jn} = f_{Im}. \tag{3.7}$$



(a) Interior sphere.

(b) Sphere on Neumann boundary.

(c) Sphere on Dirichlet boundary.

Figure 3-2: Figure showing "interior spheres" (a) and "boundary spheres" (b) & (c). The intersection of the Neumann boundary, Γ_f, with the I^{th} sphere is denoted as Γ_{f_I} (see (b)).The intersection of the Dirichlet boundary, Γ_u, with the I^{th} sphere is denoted as Γ_{u_I} (see (c)). Volume integration is performed on $\Omega_I = B(\mathbf{x}_I, r_I) \cap \Omega$.

3.2 Imposition of boundary conditions

In this section we discuss how the boundary conditions, given by equations (3.3) and (3.4), can be incorporated efficiently.

3.2.1 Neumann boundary conditions

In the finite element/finite volume method, due to the Kronecker delta property of the shape functions, only the nodes on the boundary are subjected to the applied boundary conditions. But in the MFS, the basis functions, defined on the spheres, do not satisfy the Kronecker delta condition and hence, any sphere, with nonzero intercept with the boundary contributes to the boundary integral. Let Γ_{f_I} be the intercept of the sphere I with the boundary Γ_f, see figure 3-2(b), then $\Gamma_f = \cup_{I \in \mathcal{N}_f} \Gamma_{f_I}$, where $\mathcal{N}_f$ is the index set of nodes considered. For such a sphere, equation (3.5) applies with

$$\hat{f}_{Im} = \int_{\Gamma_{f_I}} h_{Im} f^s d\Gamma. \tag{3.8}$$

3.2.2 Dirichlet boundary conditions

In the Galerkin formulation, the governing differential equation is not satisfied point-wise in the interior of the domain. Point-wise satisfaction of the essential boundary condition on a general Dirichlet boundary is similarly not possible. In the finite element techniques, the shape functions satisfy the Kronecker delta property at the nodes (i.e. the shape function at any node is unity at that node and is zero at all other nodes). Furthermore, only those nodes that lie on the Dirichlet boundary participate in the imposition of the Dirichlet boundary conditions.

Therefore, along element edges on the Dirichlet boundary, homogeneous Dirichlet boundary conditions can be exactly satisfied. When a nonhomogeneous boundary condition is prescribed the finite element approximation along the element edges on the Dirichlet boundary $u^h(\mathbf{x}) \mid_{S_u} = \sum h_I(\mathbf{x}) \mid_{S_u} u_I$ (where $h_I(\mathbf{x}) \mid_{S_u}$ is the trace of the finite element shape function $h_I(\mathbf{x})$ on the Dirichlet boundary S_u and u_I is the prescribed boundary condition at node I) converges to the applied boundary condition in a weak sense. Hence, if the finite elements exhibit polynomial consistency of order p then $\|u - u^h\|_0 \leq Ch^{p+1}$ (h denotes the element size and C is a constant depending on the problem considered but is independent of h).

In the method of finite spheres the shape functions do not satisfy the Kronecker delta property at the nodes. This is also true for the MLS (and related) shape functions. Moreover, nodes not lying on the Dirichlet boundary but with nonzero intercepts of their spheres with

the Dirichlet boundary are also involved in enforcing the boundary conditions. Indeed, to retain the flexibility of sprinkling the nodes relatively arbitrarily on the domain, we should be able to satisfy the Dirichlet conditions (in some sense) without even a single node directly on the Dirichlet boundary (so long as the spheres cover the domain). Therefore, rather than trying to satisfy the Dirichlet boundary conditions point-wise at the nodes it is more important to be able to enforce them in a weak sense along the boundary. We have therefore not considered in our work the so-called collocation techniques [37, 38].

Some of the other procedures of imposing Dirichlet boundary conditions that have been employed in the context of meshless methods are techniques involving Lagrange multipliers [11], penalty formulations [28], use of finite elements along Dirichlet boundaries [39] and modified variational principles [12]. The use of Lagrange multipliers results in indefinite systems of equations and increases the number of unknowns considerably. Penalty formulations result in ill-conditioned matrices. The use of finite elements along the Dirichlet boundaries destroys the meshless character of the approximation.

A technique of much potential is the use of modified variational principles. In this section we show how this technique may be used to impose the Dirichlet boundary conditions. This technique enforces the Dirichlet boundary conditions in a weak sense without increasing the number of unknowns. We also show that a specific arrangement of nodes on the boundary may emulate Kronecker-delta-like properties.

Referring to figure 3-2(c) we note that any node with nonzero intercept of its sphere with the boundary Γ_u contributes to the boundary integral in equation (3.5). Let Γ_{u_I} be the intercept of the sphere I with the boundary Γ_u, then $\Gamma_u = \cup_{I \in \mathcal{N}_u} \Gamma_{u_I}$, where $\mathcal{N}_u$ is the index set of nodes considered. Making use of the chain rule of differentiation, we may now write $\hat{f}_{Im}$ as

$$\hat{f}_{Im} = \sum_{J=1}^{N} \sum_{n \in \mathcal{I}} KU_{ImJn} \alpha_{Jn} - fU_{Im} \ ; \tag{3.9}$$

where

$$KU_{ImJn} = \sum_{i,j=1}^{d} \int_{\Gamma_{u_I}} \frac{\partial}{\partial x_j} \left(a_{ij}(\mathbf{x}) h_{Im} h_{Jn} n_i \right) d\Gamma, \tag{3.10}$$

$$fU_{Im} = \sum_{i,j=1}^{d} \int_{\Gamma_{u_I}} u^s \frac{\partial}{\partial x_j} (a_{ij}(\mathbf{x}) n_i h_{Im}) \, d\Gamma. \tag{3.11}$$

We note that KU_{ImJn} is a symmetric stiffness term ($KU_{ImJn} = KU_{JnIm}$) and fU_{Im} is a (known) forcing term. Hence, equation (3.5) becomes

$$\sum_{J=1}^{N} \sum_{n \in \mathcal{I}} (K_{ImJn} - KU_{ImJn}) \alpha_{Jn} = f_{Im} - fU_{Im}. \tag{3.12}$$

This procedure for imposing the Dirichlet boundary conditions is quite general but may be somewhat difficult to implement. Namely, if the nodes are distributed on and near the boundary at random and the boundary is a complex (d-1) dimensional surface, then the computation of the intercepts of the spheres with the boundary surface may become computationally intensive.

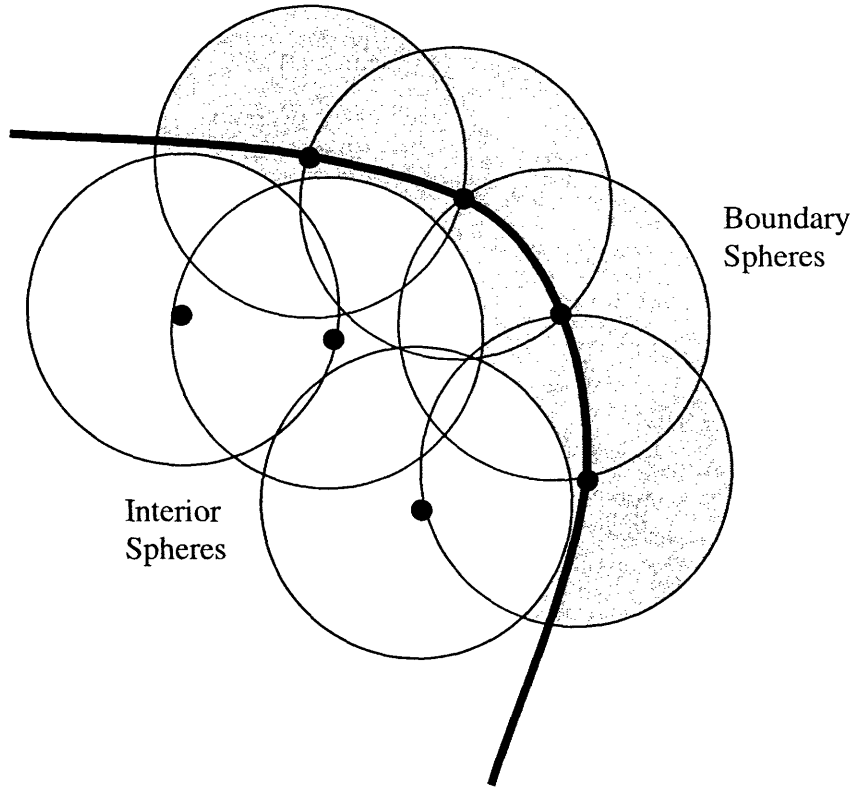


Figure 3-3: Nodal arrangement for easy incorporation of Dirichlet boundary conditions.

To circumvent this difficulty, we propose the special distribution of the boundary spheres

shown in figure 3-3. In this construction, we assume that the nodes are placed on the boundary such that the distance between two successive nodes is the radius of the spheres and that there are no nodes whose spheres intercept the boundary other than those that are on the boundary. This nodal arrangement overcomes the problem of finding the intercept of the boundary spheres with complex boundaries. The arrangement also gives rise to a Kronecker delta-like property. Then at any such boundary node I, the basis functions h_{Im} are such that

$$h_{I0}(\mathbf{x}_I) = \varphi^0_I(\mathbf{x}_I)$$

$$\text{and} \qquad h_{Im}(\mathbf{x}_I) = 0, \qquad m \neq 0.$$

By definition,

$$\varphi^0_I(\mathbf{x}_I) = \frac{W_I}{\sum_{J=1}^{N} W_J} = 1.$$

Hence, the basis function h_{I0} at node I enjoys the Kronecker delta property

$$h_{I0} = \begin{cases} 1 \text{ at node I} \\ 0 \text{ at all other nodes} \end{cases}$$

whereas the higher-order basis functions exhibit the property

$$h_{Im} = \begin{cases} 0 \text{ at node I} \\ 0 \text{ at all other nodes} \end{cases} \qquad \text{for } m \neq 0.$$

Hence, in equation (2.5)

$$v_h(\mathbf{x} = \mathbf{x}_I) = \alpha_{I0}.$$

Thus the specified value of the field variable u at node I on the Dirichlet boundary is taken up by the coefficient of h_{I0}. The implications are that for specified homogeneous (zero) Dirichlet conditions, we simply remove, from the stiffness matrix, all the rows and columns corresponding to the Shepard functions associated with the nodes that are on the Dirichlet boundary and solve the resulting set of reduced equations (3.12). If inhomogeneous Dirichlet conditions are specified, we also remove the rows and columns corresponding to the Shepard functions associated with the nodes on the Dirichlet boundary but need to bring the effect of the nonzero prescribed displacements to the right hand side of the governing equations.

Hence, equation (3.12) becomes ($m \neq 0$ with $I \in \mathcal{N}_u$)

$$\sum_{J=1}^{N} \sum_{\substack{n \in \mathcal{I} \\ if\ J \in \mathcal{N}_u \\ n \neq 0}} (K_{ImJn} - KU_{ImJn})\alpha_{Jn} = f_{Im} - fU_{Im.} - \overline{fU}_{Im} \tag{3.13}$$

where

$$\overline{fU}_{Im} = \sum_{J \in \mathcal{N}_u} (K_{ImJ0} - KU_{ImJ0})\, u^s(\mathbf{x}_J) \tag{3.14}$$

and $\mathbf{x}_J$ is the coordinate of node J. Of course $\overline{fU}_{Im} = 0$ when zero Dirichlet conditions are prescribed.

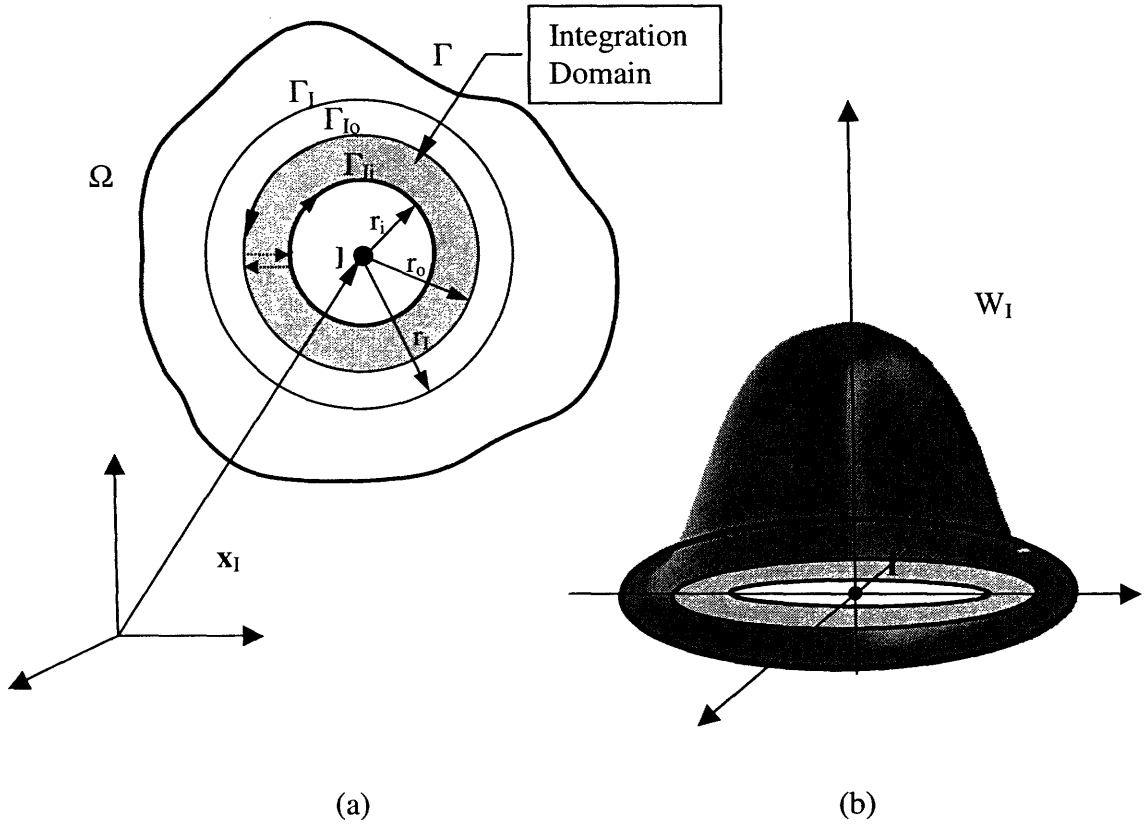


Figure 3-4: A domain Ω with a spherical cavity of radius r_i is shown in (a). Node I is placed at the center of the cavity. The weight function, W_I, at node I has a support radius of r_I (see (b)) . The integration domain associated with the node I is a spherical shell of inner radius r_i and outer radius r_o.

3.3 Doubly-connected domains: the d-dimensional spherical shell

So far we have concentrated on nodes whose integration domains are singly-connected and therefore coincide with the support. There are certain situations, however, for example, a hole in a plate, or a spherical cavity inside a three-dimensional continuum, when it would be effective to be able to directly model doubly-connected domains. The error introduced in modeling the boundaries of these cavities by placing nodes along their periphery is then eliminated and hence less nodes are required to model such geometries. Also, the known behavior of the solution of the governing equations can be included in the local bases of these nodes and thus higher convergence rates can be attained. To be able to model doubly-connected domains, we decouple the regions of support and integration.

Assume that there is a spherical cavity of radius r_i and center $\mathbf{x}_I$ inside the domain Ω (see figure 3-4(a)). We place a node, I, at the center of the cavity and associate with it a weight function W_I such that $supp(W_I) = B(\mathbf{x}_I, r_I)$, but we choose the integration domain for this node as

$$\Omega_I = B(\mathbf{x}_I, r_o) \backslash B(\mathbf{x}_I, r_i)$$

for some $r_i < r_o \leq r_I$. We see that equation (3.5) applies with the integral in $\hat{f}_{Im}$ written as the sum of two integrals (applying contour integration as shown in figure 3-4(a))

$$\hat{f}_{Im} = \sum_{i,j=1}^{d} \oint_{\Gamma_{Io}} a_{ij}(\mathbf{x}) h_{Im} n_i \frac{\partial u_h}{\partial x_j} d\Gamma + \sum_{i,j=1}^{d} \oint_{\Gamma_{Ii}} a_{ij}(\mathbf{x}) h_{Im} n_i \frac{\partial u_h}{\partial x_j} d\Gamma, \tag{3.15}$$

where $\Gamma_{Io} = S(\mathbf{x}_I, r_o)$ and $\Gamma_{Ii} = S(\mathbf{x}_I, r_i)$. We consider two cases:

Case (1) $r_o = r_I$: The first integral in equation (3.15) is zero due to the property of compact support and we have

$$\hat{f}_{Im} = \sum_{i,j=1}^{d} \oint_{\Gamma_{Ii}} a_{ij}(\mathbf{x}) h_{Im} n_i \frac{\partial u_h}{\partial x_j} d\Gamma. \tag{3.16}$$

Usually we have some boundary data prescribed on the inside surface of the cavity which can be incorporated using the techniques described in the previous section.

Case (2) $r_o < r_I$: In this case we have to use equation (3.15) in its full form.

We present a numerical example demonstrating the technique in this section in chapter 5.

3.4 Numerical examples

In this section we present numerical examples in one and two dimensions demonstrating the above formulation. A simple problem involving a bar with distributed loading is solved in one dimension, followed by a one-dimensional steady-state convection-diffusion problem. In two-dimensions we solve a Poisson problem with mixed boundary conditions. Corresponding to each distinct type of equation solved by the MFS, a patch test was performed and the method passed the patch test in each case.

3.4.1 The MFS in R^1 : a bar with distributed loading

Formulation

In R^1 the "spheres" reduce to line-segments (as shown in figure 3-5(a)). We solve the following problem of a bar of unit length, subjected to a distributed loading:

$$\begin{aligned} \frac{d^2u(x)}{dx^2} + f(x) &= 0 && \text{in } \Omega = (0,1) \\ u &= u^s && \text{at } x = 0 \\ \frac{du}{dx} &= f^s && \text{at } x = 1 \end{aligned} \tag{3.17}$$

The parameters u^s and f^s and the function $f(x)$ are chosen so that the analytical solution u is given by the following expression:

$$u(x) = \frac{1}{2}(x - \frac{x^3}{3}) + 2x + 1.$$

At each node I, the following shape functions were used;

$$\{\varphi_I^0(x), \varphi_I^0(x)(x - x_I)/r_I\}$$

where $\varphi_I^0(x)$ is the Shepard function at node I.

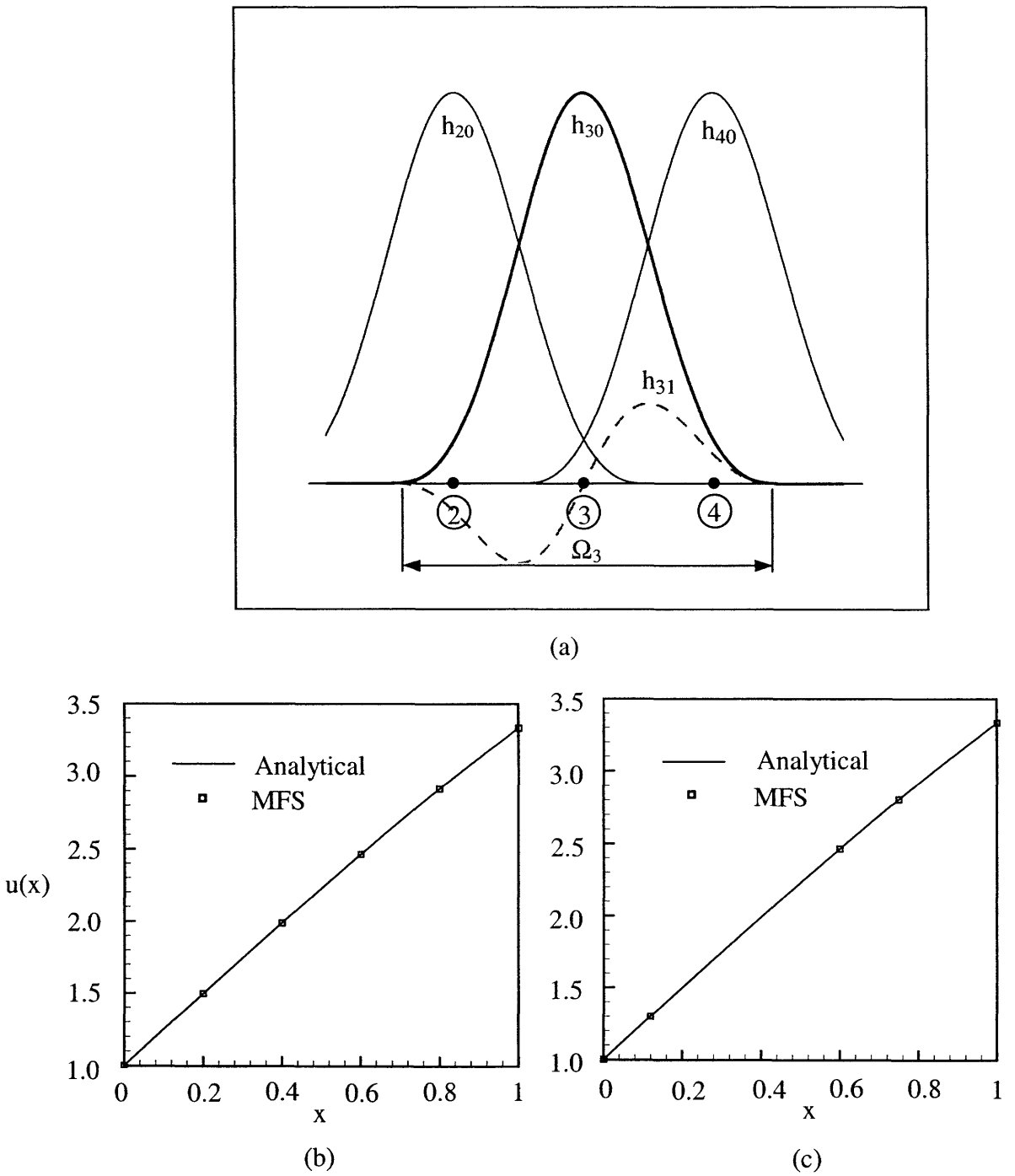


Figure 3-5: A bar of unit length with distributed loading. In (a) a part of the bar is shown with 3 nodes. The Shepard functions h_{I0} (I = 2, 3, 4) are plotted at each node. At node 3, a higher order shape function $h_{31} = \frac{(x-x_3)}{r_3} h_{30}$ is also plotted. The sphere at each node I (Ω_I) reduces to a line segment in one-dimension. In (b) and (c) the displacement field u(x) is plotted as a function of the distance along the bar corresponding to the boundary conditions and loading given in the text. The numerical result in (b) corresponds to a regular distribution of 6 nodes on the bar while that in (c) corresponds to an arbitrary distribution of 5 nodes.

Figure 3-5(a) shows a plot of these shape functions for a typical node within the domain. The discretized equation corresponding to the I^{th} node and m^{th} degree of freedom is given by equation (3.5) where the integrals are:

$$\begin{aligned} K_{ImJn} &= \int_{x_1}^{x_2} \frac{dh_{Im}}{dx}\frac{dh_{Jn}}{dx}dx \\ f_{Im} &= \int_{x_1}^{x_2} f(x)h_{Im}dx \\ \hat{f}_{Im} &= 0 && \text{for an "interior sphere"} \\ & f^s h_{Im}(x=1) && \text{for a sphere on the Neumann boundary} \\ & \sum_{J=1}^{N}\sum_{n\in\mathcal{I}} KU_{ImJn}\alpha_{Jn} - fU_{Im} && \text{for a sphere on the Dirichlet boundary} \end{aligned} \tag{3.18}$$

where

$$\begin{aligned} KU_{ImJn} &= -\left[\frac{d}{dx}\left(h_{Im}h_{Jn}\right)\right]_{x=0} \\ fU_{Im} &= -u^s\left[\frac{dh_{Im}}{dx}\right]_{x=0} \end{aligned} \tag{3.19}$$

it being understood that $x_1 = max(0, x_I - r_I)$ and $x_2 = min(1, x_I + r_I)$.

Numerical results

Both regular and arbitrary distributions of nodes have been used to solve the problem. Figures 3-5(b) shows the result when six nodes are distributed regularly. Figure 3-5(c) shows the solution when the same problem is solved using five nodes with arbitrary distances between them.

3.4.2 The MFS in R^1: convection-diffusion problem

Formulation

We consider the following steady-state convection-diffusion problem in one dimension with prescribed fluid velocity v as discussed in [1]. The temperature is prescribed at two points, $x = 0$ and $x = L$ and we intend to compute the temperature in $(0, L)$. The governing differential equation for the temperature, θ, is

$$\frac{d^2\theta}{dx^2} = \frac{Pe}{L}\frac{d\theta}{dx} \qquad \text{in } (0, L) \tag{3.20}$$

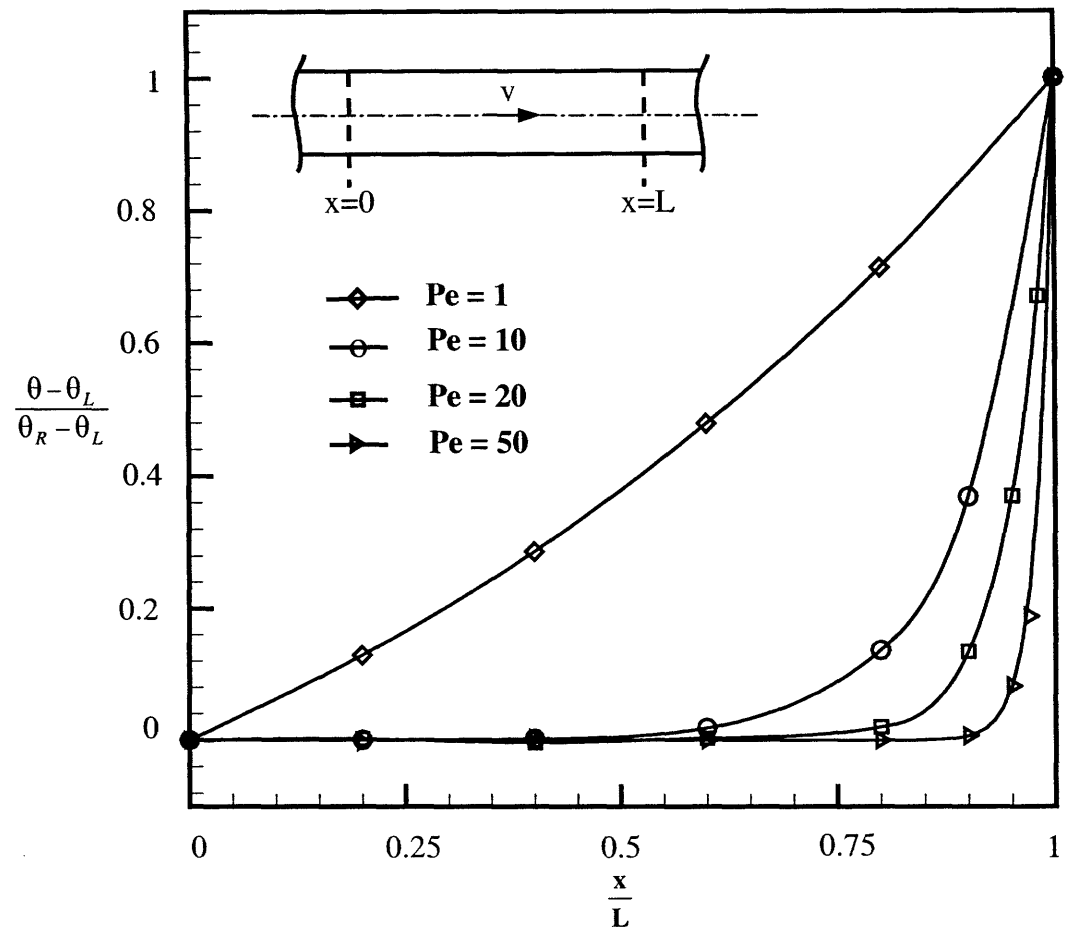


Figure 3-6: Results of the high Peclet number flow problem. The normalized temperature distribution is plotted against normalized distance along the flow direction for four different Peclet numbers (Pe = 1, 10, 20 and 50). Continuous lines correspond to the analytical solution. The solution obtained using the method of finite spheres is plotted corresponding to arbitrary distributions of nodes along the domain.

with the boundary conditions;

$$\begin{aligned} \theta &= \theta_L \qquad \text{at } x = 0 \\ \theta &= \theta_R \qquad \text{at } x = L \end{aligned} \tag{3.21}$$

where the Peclet number is defined as $Pe = \frac{vL}{\alpha}$ (α is the thermal diffusivity of the fluid). The exact solution to this problem is given by

$$\frac{\theta - \theta_L}{\theta_R - \theta_L} = \frac{\exp(\frac{Pe}{L}x) - 1}{\exp(Pe) - 1}. \tag{3.22}$$

With increase in the Peclet number, the solution curve shows a strong boundary layer at $x = L$. The solution using a simple Galerkin finite element scheme leads to severe numerical difficulties and a variety of upwind-type procedures have been proposed to solve the problem (see [1]). In this work we apply the MFS and simply use as our local approximation spaces

$$V_I^h = span\{1, \exp(\frac{Pe}{L}x)\}.$$

Numerical results

Figure 3-6 shows the numerical solutions obtained using the MFS, with quite arbitrarily spaced nodes, plotted on the analytical solution curves for $Pe = 1, 10, 20$ and 50. Due to the solution space chosen we expect a very accurate response with no "wiggles", and this is the case, see Figure 3-6. In Appendix A we present a stability analysis of the one-dimensional convection-diffusion problem using numerical inf-sup tests.

3.4.3 The MFS in R^2: the Poisson equation on the bi-unit square

Formulation

We consider a problem in a single field variable defined on R^2. We seek a function $u(x, y)$ satisfying the Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + f(x, y) = 0 \qquad \text{in } \Omega = (-1, 1) \times (-1, 1) \tag{3.23}$$

subject to the boundary conditions (see figure 3-7)

$$\begin{array}{rcll} u & = & u^s(x, y) & \text{on } \Gamma_a \\ \frac{\partial u}{\partial y} & = & 0 & \text{on } \Gamma_b \\ \frac{\partial u}{\partial x} & = & -f^s(x, y) & \text{on } \Gamma_c \\ \frac{\partial u}{\partial y} & = & 0 & \text{on } \Gamma_d. \end{array} \tag{3.24}$$

The functions $f(x, y), f^s(x, y)$ and $u^s(x, y)$ are chosen such that the analytical solution $u(x, y)$ is

$$u(x, y) = (7x + x^7)\cos(\pi y). \tag{3.25}$$

The discretized equation corresponding to the m^{th} degree of freedom of the I^{th} node is given

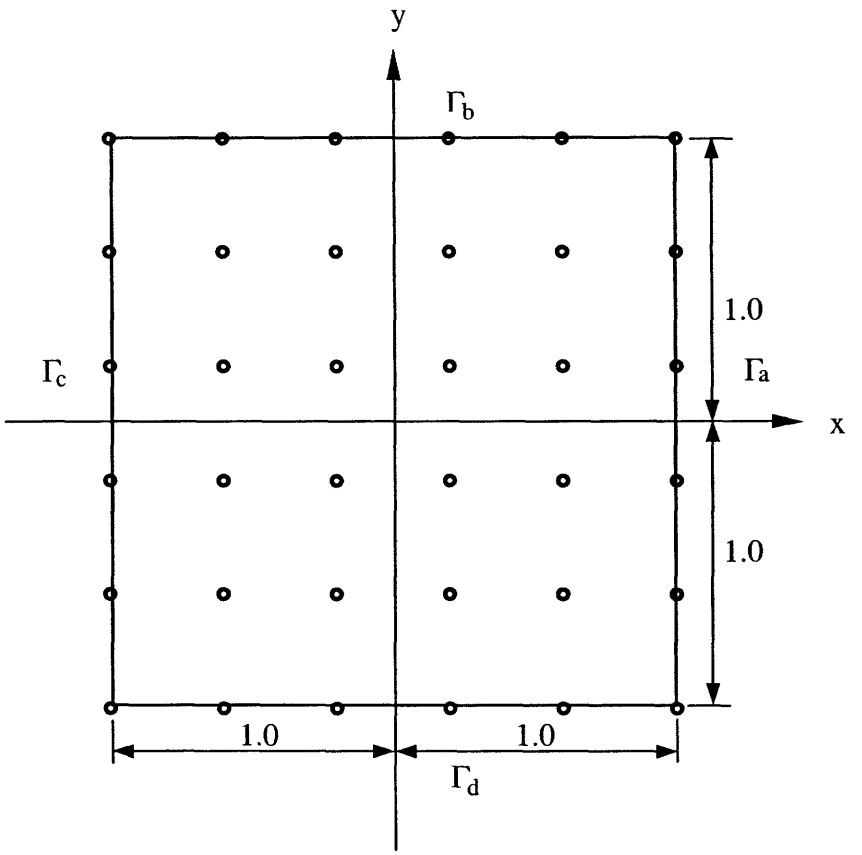


Figure 3-7: A regular arrangement of 36 nodes is shown on the domain on which the Poisson problem is defined.

by equation (3.5) where the integrals are

$$\begin{aligned}
K_{ImJn} &= \int_{\Omega_I} \left(\frac{\partial h_{Im}}{\partial x}\frac{\partial h_{Jn}}{\partial x} + \frac{\partial h_{Im}}{\partial y}\frac{\partial h_{Jn}}{\partial y} \right) d\Omega \\
f_{Im} &= \int_{\Omega_I} f(x,y) h_{Im}\, d\Omega \\
\hat{f}_{Im} &= \begin{cases} 0 & \text{for an ``interior sphere''} \\ 0 & \text{for spheres on } \Gamma_b \text{ and } \Gamma_d \\ \int_{\Gamma_{c_I}} f^s h_{Im}\, d\Omega & \text{for a sphere on } \Gamma_c \\ \sum_{J=1}^{N} \sum_{n\in\mathcal{I}} KU_{ImJn}\alpha_{Jn} - fU_{Im} & \text{for a sphere on } \Gamma_a \end{cases}
\end{aligned} \tag{3.26}$$

where

$$\begin{aligned}
KU_{ImJn} &= \int_{\Gamma_{a_I}} \frac{\partial}{\partial x}\left(h_{Im}h_{Jn}\right) d\Gamma \\
fU_{Im} &= \int_{\Gamma_{a_I}} u^s \frac{\partial h_{Im}}{\partial x}\, d\Gamma
\end{aligned} \tag{3.27}$$

and $\Gamma_a = \cup_{I \in \mathcal{N}_a} \Gamma_{a_I}$ and $\Gamma_c = \cup_{I \in \mathcal{N}_c} \Gamma_{c_I}$, $\mathcal{N}_a$ and $\mathcal{N}_c$ being the index set of nodes whose spheres have nonzero intercepts with the boundaries Γ_a and Γ_c respectively.

Numerical results

Figure 3-7 shows the discretization of the domain using a regular arrangement of 36 nodes. In figure 3-8(a) we present the MFS solution $u(x)$ superposed on the analytical one along two lines $y = 0$ and $y = 1$. In figure 3-8(b) the solution $u(y)$ (as computed by the MFS) is shown as a function of the spatial coordinate y along the boundary Γ_a (i.e. $x = 1$) together with the analytical solution. Note that the essential boundary condition prescribed along this boundary is exactly satisfied at the nodes but only approximately (in a weak sense) in-between the nodes.

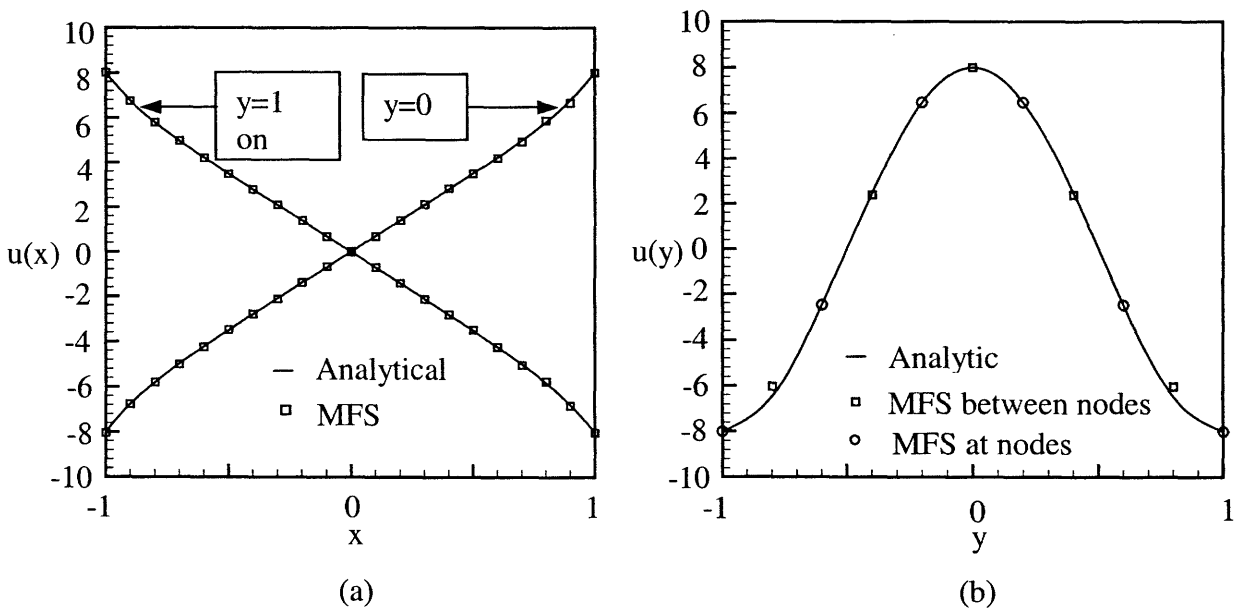


Figure 3-8: Results of the Poisson problem. In (a) the field variable u(x) is plotted as a function of x for two values of y (= 0, 1). In (b) the Dirichlet boundary Γ_a is considered. The MFS solution is plotted as a function of y along with the analytical solution.

Chapter 4

Numerical integration rules

The key to a successful meshless numerical technique lies in the choice of efficient numerical integration rules. In this chapter we discuss this very important issue.

Numerical quadrature in one dimension is quite well understood. There are only three types of domains to be considered; a finite line segment, a semi-infinite line segment and the entire real line. Moreover, in one dimension, the theory of numerical quadrature is closely related to the theory of orthogonal polynomials which is quite well understood for one variable.

In multiple dimensions, however, there are infinitely many bounded connected regions that are not equivalent under affine transformation. For example, the circle, square and rectangle are not equivalent under such transformation and therefore distinct integration rules have to be developed for each of these regions. Moreover, the theory of orthogonal polynomials is not straightforward in multiple dimensions, especially if the geometry is complex. Therefore, in multiple dimensions, the task of developing integration rules is not simple.

One of the primary reasons for the success of the finite element technique is the ease with which numerical integration can be performed in multiple dimensions. The functions to be integrated are usually polynomials (or mapped polynomials), the elements do not overlap and they can be mapped to d-dimensional cubes. Hence Gauss-Legendre product rules are used for numerical integration with relatively low computational cost and high accuracy.

In the method of finite spheres, however, the shape functions are rational (nonpolynomial) functions (in figures 4-1 to 4-3 we show some of the functions that are evaluated numerically on [0,1] discretized using five nodes) and the integration domains are spheres, spherical shells or general sectors. Moreover, the spheres overlap giving rise to general "lens" shaped regions. Hence, a separate class of integration rules is required. Exact integration of the stiffness terms is not possible since the integration rules can, at best, guarantee polynomial accuracy. Hence the focus is on obtaining rules that use minimal number of integration points for a given accuracy.

In this chapter we discuss only Gaussian product rules of cubature for the method of finite spheres. Nonproduct rules either require more points for same polynomial accuracy or need the generation of complicated orthogonal polynomials on the complex geometries we encounter in the method of finite spheres. The classical Monte Carlo method is ill famous for its $O(1/\sqrt{N})$ accuracy (where N is the number of integration points used). While the quasi-Monte Carlo techniques provide better accuracy, they are not recommended for integration in two or three dimensions where the integrand is well behaved [40].

In our work we have considered integration rules on two-dimensional domains. In Appendix B we state and prove a Gaussian product rule of cubature on a two-dimensional annular sector. Even though cubature rules exist for circular annuli as well as spheres and hyperspheres [40, 41, 42, 43, 44], a rule for annular sectors seem to be an original contribution of this thesis. In section 4.1 we present a specialized version of this rule for integration on interior disks. In section 4.2 we discuss product rules on boundary sectors. In section 4.3 we consider the problem of integration on the "lens" shaped region of overlap of two disks.

4.1 Integration on an interior disk

Peirce [41] presented a general product rule with polynomial accuracy k on the planar annulus. This rule may also be derived as a special case of the general rule presented in Appendix B. In this section we are concerned with integration on the planar disk and

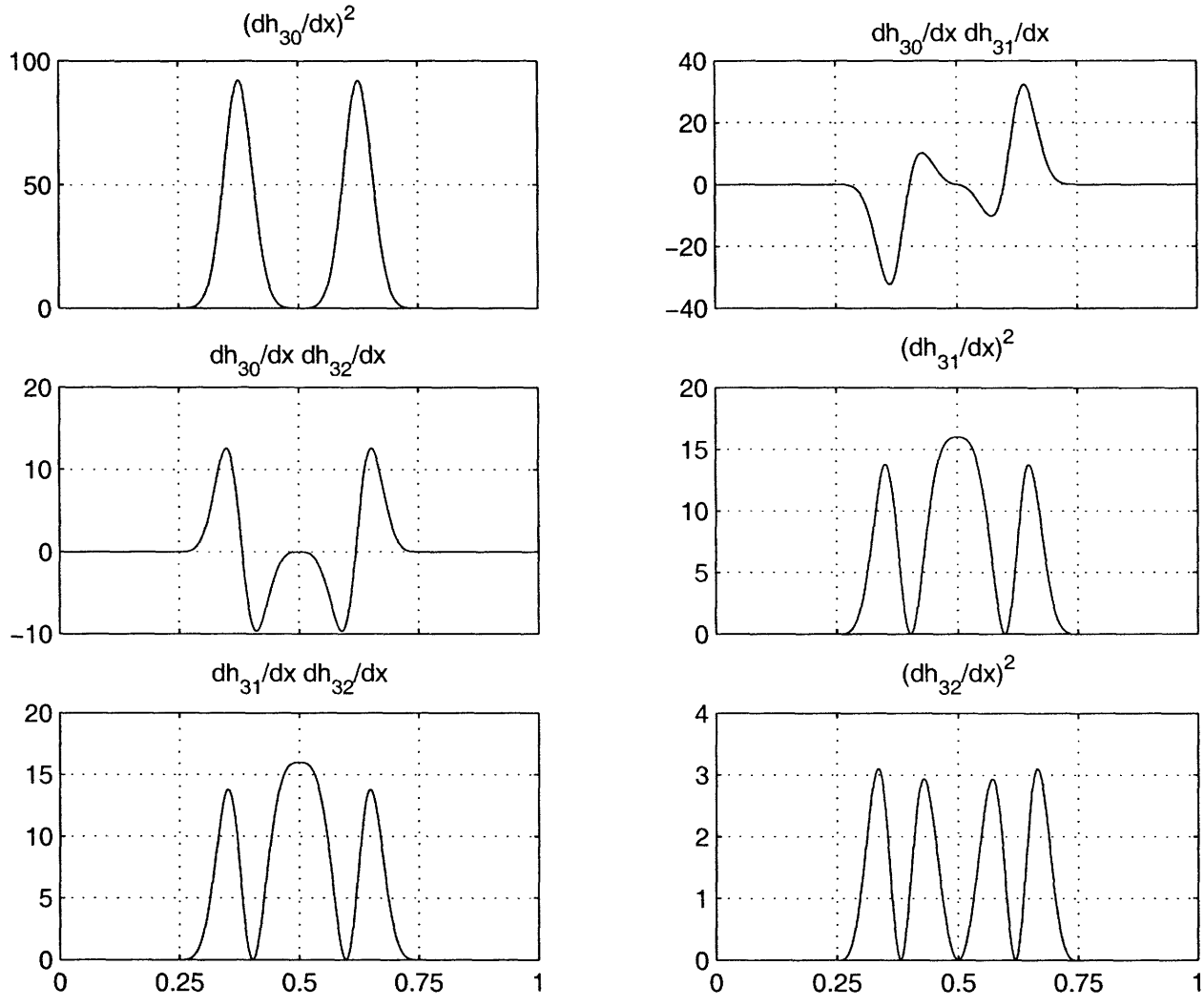


Figure 4-1: Examples of some functions that are numerically integrated for the one-dimensional bar problem in chapter 3. The domain, [0,1], is discretized using 5 nodes spaced 0.25 apart.

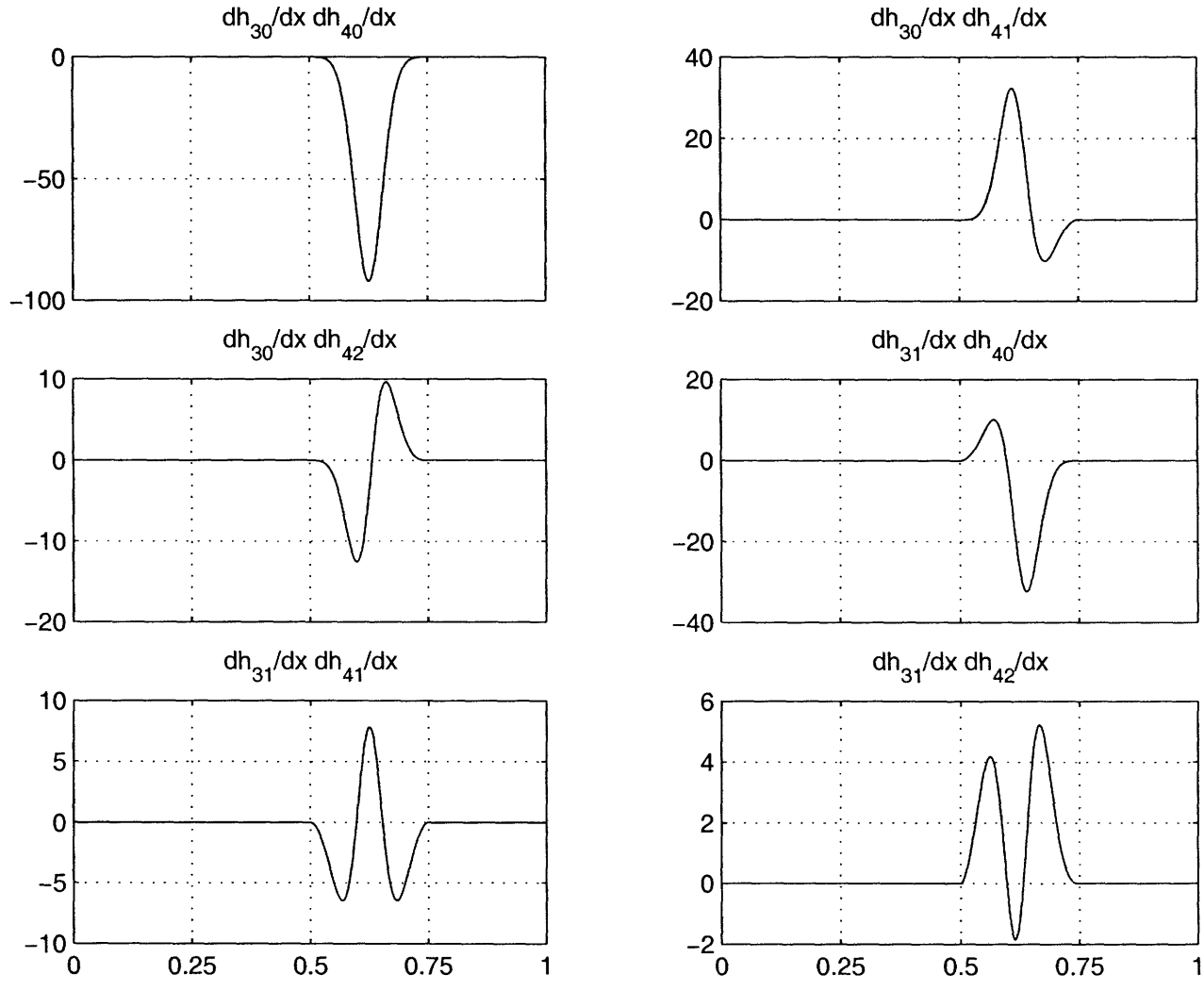


Figure 4-2: Some more functions on the same domain as in figure 4-1

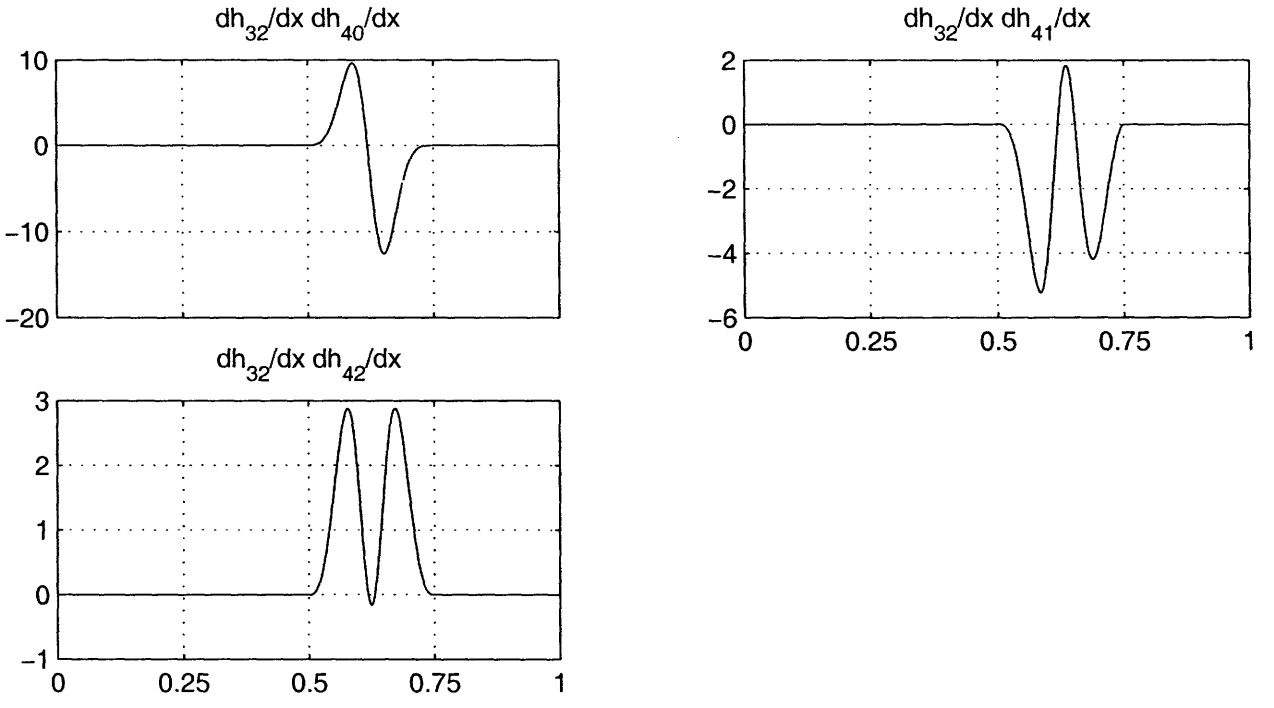


Figure 4-3: Some more functions on the same domain as in figure 4-1

require a rule of the form

$$\iint_{\Omega} f(x,y)dxdy \doteq \sum_i \sum_j D_{ij} f(x_i, y_j). \tag{4.1}$$

The region, Ω, under consideration is a disk with radius R_o (see figure 4-4). The dot over the equality signifies that the relationship is a strict equality if the function $f(x,y)$ is a polynomial of order at most k in x and y, otherwise it is an approximation. The following theorem sates the relevant integration rule:

Theorem 4.1 : *If it is required that the rule in equation (4.1) have accuracy k=4m+3, m=0,1,2,..., in* $x = r\cos\theta$ *and* $y = r\sin\theta$ *, and if it is required to have a minimum number of evaluation points which are taken at the intersection of concentric arcs (radius* r_j*) with rays emanating from the origin (angle* θ_i*), then it is both necessary and sufficient for the existence of a unique set of weights* $D_{ij} \in R$ *that*

$$\theta_{i+1} - \theta_i = \frac{2\pi}{k+1} \qquad\qquad i = 1, 2, \ldots, k \tag{4.2}$$

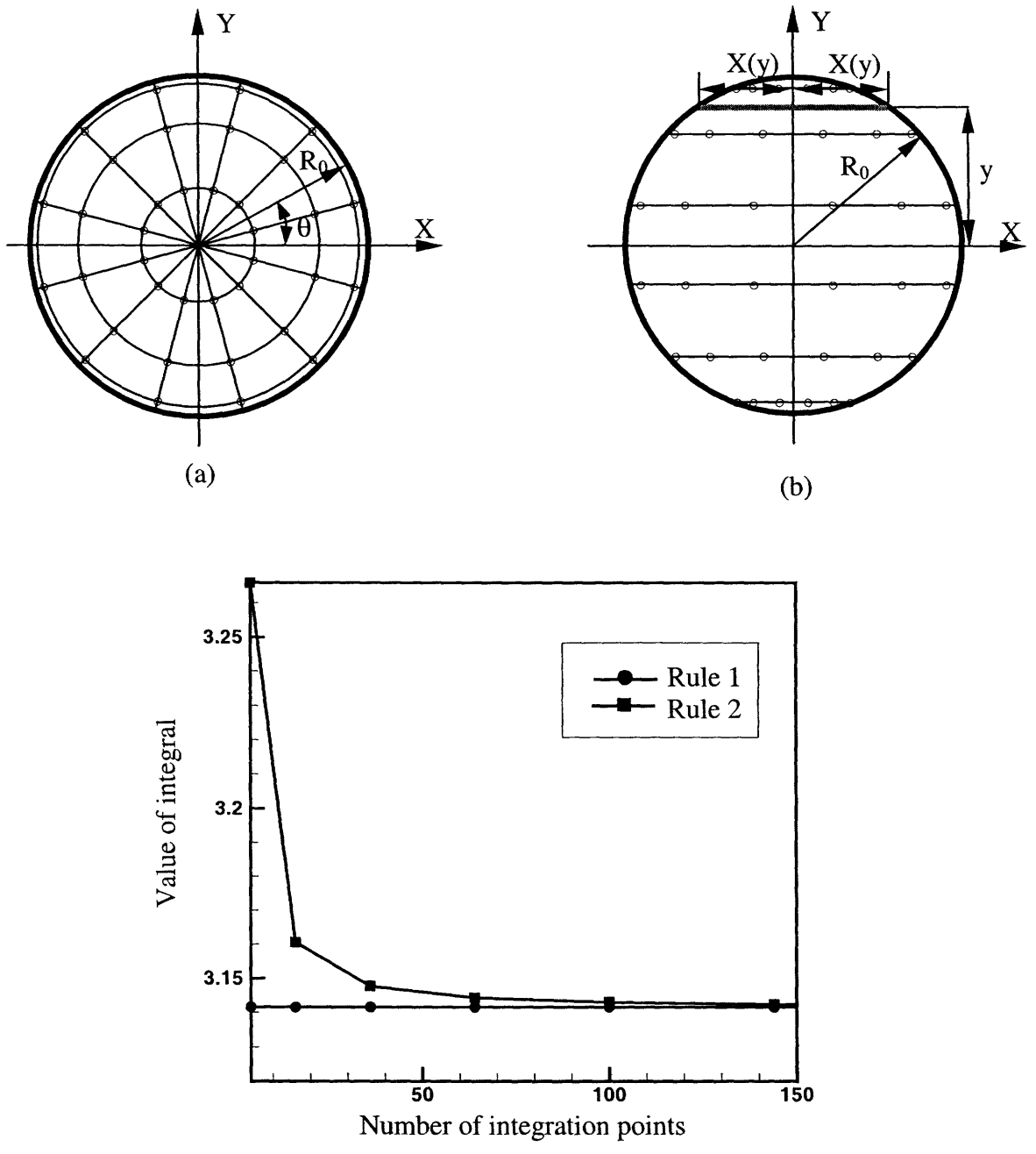


Figure 4-4: Integration on an "interior" disk of radius 1.0. In (a) integration points corresponding to the rule in Theorem 4.1 (Rule 1)are shown. To integrate a polynomial of degree 11 exactly 36 integration points are required. In (b) integration stations corresponding to a Gauss-Legendre product rule (Rule 2) are shown. In (c) Rules 1 and 2 are used to evaluate the area of the disk.

and that the r_i^2 be the $m+1$ zeros of $P_{m+1}(r^2)$, the Legendre polynomial in r^2 of degree $m+1$, orthogonalized on $[0, R_o^2]$. The (unique) weights D_{ij} are of the form $A_i B_j$, where

$$A_i = \frac{2\pi}{k+1} \qquad i = 1, 2, \ldots, k+1 \tag{4.3}$$

and

$$B_j = \frac{1}{2P'_{m+1}(r_j^2)} \int_0^{R_o^2} \frac{P_{m+1}(r^2)}{r^2 - r_j^2} dr^2 \qquad j = 1, 2, \ldots, m+1 \tag{4.4}$$

The integration points are on equally spaced radii and the integration weights are independent of angular position (Gauss-Chebyshev rule in the θ-direction).

To demonstrate the efficiency of this rule we compare it with the Gauss-Legendre product rule on the disk

$$\iint_{\Omega} f(x,y)dxdy = \int_{y=-R_o}^{R_o} \int_{x=-X(y)}^{X(y)} f(x,y)\, dxdy \simeq \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} D_{ij} f(x_i, y_j) \tag{4.5}$$

where N_x and N_y are the number of integration points chosen along the x and y-directions, respectively, and $D_{ij} = W_i^x W_j^y$ is the product of the usual Gaussian weights W_i^x and W_j^y in the x and y-directions. It is not possible to guarantee exact integration of polynomials of any degree using this rule. For demonstration we consider the simple problem of computing the area of a unit circle (where $f(x,y) = 1$) in figure 4-4(c).

4.2 Integration on a boundary sector

We categorize the boundary sectors into two major groups depending on the angle $\varphi_0 = 2\theta_0$ that the radii joining the center of the disk to the two intercepts of the disk on S make interior to the domain:

Type I sector: $\varphi_0 \leq \pi$ (see figure 4-5(a)). The rule that allows us to perform numerical cubature on this sector to any desired order of accuracy is computationally expensive since it requires the evaluation of the zeros of a different orthogonal polynomial for every φ_0. We,

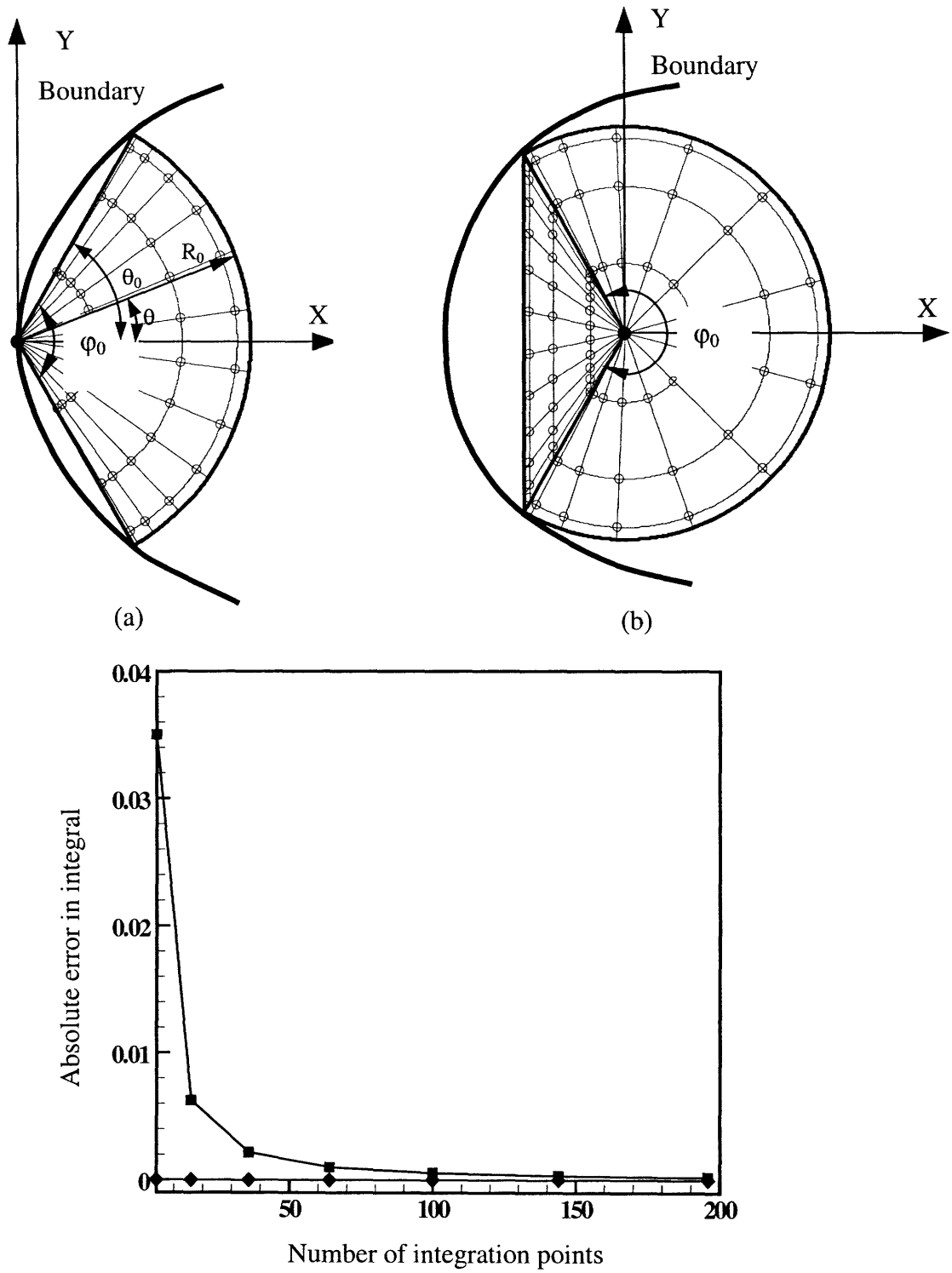


Figure 4-5: Integration points on a boundary sector. (a) Type I boundary sector with $\varphi_0 \leq \pi$. (b) Type II boundary sector with $\varphi_0 > \pi$. (c) Absolute error as a function of number of integration points when $f(x,y) = 1$ (filled diamonds) and $f(x,y) = x$ (filled squares) are integrated on a Type I sector of unit radius and $\varphi_0 = 2\pi/3$ using the rule in section 4.2

therefore, propose the following "engineering solution"

$$\iint_{\Omega} f(x,y)dxdy \simeq \sum_{i=1}^{m+1} \sum_{j=1}^{4(m+1)} D_{ij} f(r_i \cos\theta_j, r_i \sin\theta_j) \tag{4.6}$$

where $D_{ij} = A_i B_j$ with A_i being the usual Gaussian weights on an interval $[-\theta_0, \theta_0]$ and B_j as in equation (4.4). The integration points r_i are the positive square roots of the zeros of the Legendre polynomial $P_{m+1}(r^2)$ in $r^2$of degree $m+1$ ($k = 4m+3$), orthogonalized on $[0, R_o^2]$ as in Theorem 4.1. However, the radial coordinates of the integration points θ_j are chosen as the zeros of the Legendre polynomial $P_{4(m+1)}(\theta)$ in θ of degree $4(m+1)$, orthogonalized on $[-\theta_0, \theta_0]$.

This rule guarantees arbitrary polynomial accuracy if the integrand in (4.6) is a purely radial function. Hence, as figure 4-5(c) shows, the area of the sector ($f(x,y) = 1$) can be exactly integrated but if the integrand is $f(x,y) = x$, for example, exact integration is not possible. Notice, however, that this rule is better than a pure Gauss-Legendre product rule on the sector as in equation (4.5).

Type II sector: $\varphi_0 > \pi$ (see figure 4-5(b)). This type of boundary sector is more expensive to handle. We decompose a Type II sector into a sector for which the rules of the Type I sector can be used and a triangle as shown in figure 4-5(b). For the triangle we use a product rule based on Gauss-Legendre quadrature.

4.3 Integration on the lens

To be able to integrate efficiently on the lens shaped region of intersection of two disks, we need specialized integration rules. In this section we propose and compare two schemes for numerically evaluating $\iint_{\Omega_{IJ}} f(x,y)dxdy$ where $\Omega_{IJ} = B(\mathbf{x}_I, r_I) \cap B(\mathbf{x}_J, r_J) \neq 0$.

Scheme 1 (see figure 4-6(a)): In this scheme we use a Gauss-Legendre product rule of the

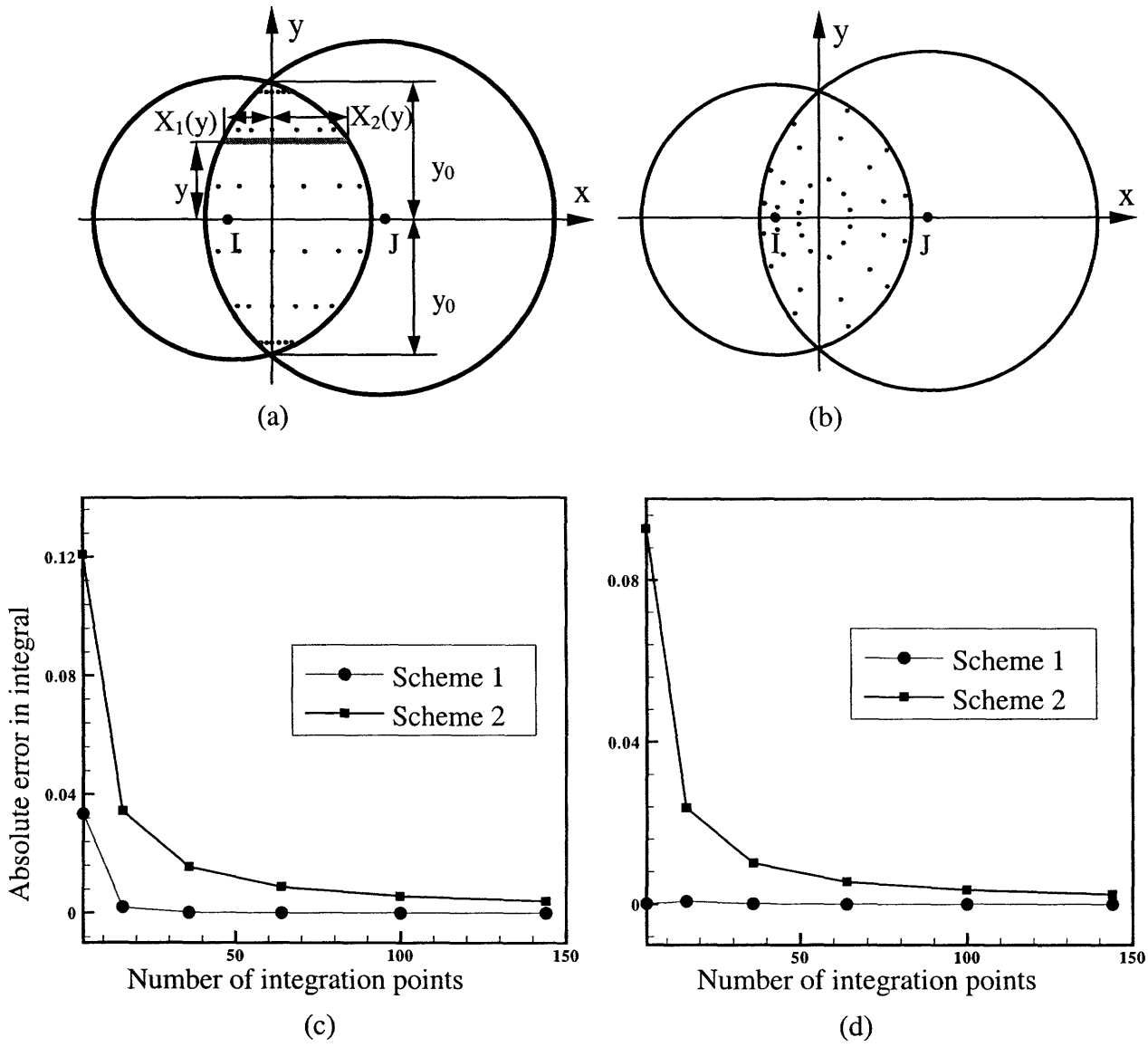


Figure 4-6: Integration on the lens. Some integration points generated using Schemes 1, see (a) , and 2, see (b), of section 4.3 are shown on the intersection of two disks of radii 0.8 and 1.0, respectively, with center-to-center distance of 0.9. Absolute errors as a function of number of integration points when $f(x,y) = 1$, see (c) and $f(x,y) = (1 - x^2 - y^2)^{1/2}$, see (d), are integrated on the intersection region using Schemes 1 and 2 are also shown.

form

$$\iint_{\Omega_{IJ}} f(x,y)dxdy = \int_{y=-y_o}^{y_o} \int_{x=-X_1(y)}^{X_2(y)} f(x,y)\, dxdy \simeq \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} D_{ij} f(x_i, y_j) \tag{4.7}$$

where N_x and N_y are the number of integration points chosen along the x and y-directions, respectively and $D_{ij} = W_i^x W_j^y$ is the product of the usual Gaussian weights W_i^x and W_j^y in the x and y-directions.

Scheme 2 (see figure 4-6(b)): In this scheme, we map the domain Ω_{IJ} onto a unit disk and compute the resulting integral using the scheme in Theorem 4.1

$$\iint_{\Omega_{IJ}} f(x,y)dxdy = \int_{\rho=0}^{1} \int_{\theta=0}^{2\pi} F(\rho,\theta) J\, \rho\, d\rho\, d\theta \tag{4.8}$$

where J is the Jacobian of the transformation.

We note that none of the schemes can guarantee polynomial accuracy for the integrand $f(x,y)$. This is because the integration limits in rule (4.7) are not simple. On the other hand, the Jacobian, in equation (4.8), is a complex function to integrate. In figure 4-6(c) the absolute error in integrating the area of the intersection region is shown as a function of the number of integration points. Note that neither scheme can integrate $f(x,y) = 1$ exactly on the domain. In figure 4-6(d) the absolute value of the error in integrating a more complex function $f(x,y) = \sqrt{1-x^2-y^2}$ is shown as a function of the number of integration points. Such numerical experiments show that scheme 1 requires fewer integration points for the same accuracy than scheme 2.

Chapter 5

Linear elasticity problems in R^2 : displacement-based formulation

Linear elasticity occupies an important role in a variety of engineering problems. In this chapter we apply the method of finite spheres to solve such problems. Attention is focused on problems in R^2 where the integration rules developed in chapter 4 are directly applicable. We adopt a pure displacement-based formulation [1] where the displacements are interpolated and the strain and stress fields are computed from the displacement fields. While, in this chapter, we develop formulations for plane strain and plane stress problems, extension to axisymmetric and full three-dimensional analysis is straightforward. Moreover, the formulations presented in this chapter directly carry over to problems in fluid mechanics (velocities replacing displacements).

In section 5.1 we briefly summarize the governing equations for a linear elastic continuum. In section 5.2 we present the variational formulation of the elasticity problem. The nodal discretization scheme and the discretized set of equations are presented in sections 5.3 and 5.4 respectively. Finally, in section 5.5 we present numerical examples.

5.1 Governing equations

The system of governing differential equations and the boundary conditions for a linear elastic continuum $\Omega \in R^2$ with boundary Γ can be written as:

Equilibrium Equations :

$$\partial_\epsilon^T \boldsymbol{\tau} + \mathbf{f}^B = \mathbf{0} \qquad \text{in } \Omega \tag{5.1}$$

Strain – Displacement Relationships :

$$\boldsymbol{\epsilon} = \partial_\epsilon \mathbf{u} \qquad \text{in } \Omega \tag{5.2}$$

Linear Elastic Constitutive Relationship :

$$\boldsymbol{\tau} = \mathbf{C}\boldsymbol{\epsilon} \qquad \text{in } \Omega \tag{5.3}$$

Boundary Conditions :

$$\mathbf{N}\boldsymbol{\tau} = \mathbf{f}^S \qquad \text{on } \Gamma_f \tag{5.4}$$

$$\mathbf{u} = \mathbf{u}^S \qquad \text{on } \Gamma_u \tag{5.5}$$

In the equations (5.1) to (5.5), $\mathbf{u}$, $\boldsymbol{\epsilon}$ and $\boldsymbol{\tau}$ are the displacement, stress and strain vectors, $\mathbf{C}$ is the elasticity matrix, $\mathbf{f}^S$ is the prescribed traction vector on the Neumann boundary Γ_f, $\mathbf{u}^S$ is the vector of prescribed displacements on the Dirichlet boundary Γ_u (note that the domain boundary $\Gamma = \Gamma_f \cup \Gamma_u$), $\mathbf{f}^B$ is the body force vector (including inertia terms), ∂_ϵ is a linear gradient operator and $\mathbf{N}$ is the matrix of direction cosine components of a unit normal to the domain boundary (positive outwards). In R^2 these vectors and matrices are written as:

$$\mathbf{u} = [u(x,y) \;\; v(x,y)]^T \tag{5.6}$$

$$\boldsymbol{\epsilon} = [\varepsilon_{xx} \;\; \varepsilon_{yy} \;\; \gamma_{xy}]^T \tag{5.7}$$

$$\boldsymbol{\tau} = [\tau_{xx} \;\; \tau_{yy} \;\; \tau_{xy}]^T \tag{5.8}$$

$$\mathbf{f}^S = [f_x^S(x,y) \;\; f_y^S(x,y)]^T$$

$$\mathbf{u}^S = [u^S(x,y) \;\; v^S(x,y)]^T$$

$$\boldsymbol{\partial}_\epsilon = \begin{bmatrix} \partial/\partial x & 0 \\ 0 & \partial/\partial y \\ \partial/\partial y & \partial/\partial x \end{bmatrix} \tag{5.9}$$

$$\mathbf{N} = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \tag{5.10}$$

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & 0 \\ c_{12} & c_{11} & 0 \\ 0 & 0 & c_{33} \end{bmatrix}$$

where for *plane stress* conditions

$$c_{11} = \frac{E}{1-\nu^2},\ c_{12} = \frac{E\nu}{1-\nu^2} \text{ and } c_{33} = \frac{E}{2(1+\nu)}$$

and for *plane strain* conditions

$$c_{11} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)},\ c_{12} = \frac{E\nu}{(1+\nu)(1-2\nu)} \text{ and } c_{33} = \frac{E}{2(1+\nu)}$$

E and ν being the Young's modulus and Poisson's ratio of the material, respectively.

5.2 Variational form

We consider the following variational indicator [1]

$$\Pi^*(\mathbf{u}) = \int_\Omega \frac{1}{2}\boldsymbol{\epsilon}^T(\mathbf{u})\mathbf{C}\boldsymbol{\epsilon}(\mathbf{u})\, d\Omega - \Re. \tag{5.11}$$

The term $\Re$ accounts for the externally applied body forces, surface tractions and applied displacements,

$$\Re = \int_\Omega \mathbf{u}^T\mathbf{f}^B\, d\Omega + \int_{\Gamma_f} \mathbf{u}^T\mathbf{f}^S\, d\Gamma + \int_{\Gamma_u} \mathbf{f}^{uT}(\mathbf{u}-\mathbf{u}^S)\, d\Gamma \tag{5.12}$$

where $\mathbf{f}^u$ is the traction vector on the Dirichlet boundary and may be expressed as

$$\mathbf{f}^u = \mathbf{N}\mathbf{C}\boldsymbol{\epsilon}(\mathbf{u}). \tag{5.13}$$

Here $\mathbf{f}^u$ is the vector of Lagrange multipliers enforcing the Dirichlet boundary conditions.

Invoking the stationarity of Π^* we obtain the following weak form

$$\begin{aligned} &\int_\Omega \boldsymbol{\epsilon}^T(\mathbf{v})\mathbf{C}\boldsymbol{\epsilon}(\mathbf{u})\,d\Omega - \int_{\Gamma_u}\left[\boldsymbol{\epsilon}^T(\mathbf{v})\mathbf{C}\mathbf{N}^T\mathbf{u} + \mathbf{v}^T\mathbf{N}\mathbf{C}\boldsymbol{\epsilon}(\mathbf{u})\right]\,d\Gamma \\ &= \int_\Omega \mathbf{v}^T\mathbf{f}^B\,d\Omega + \int_{\Gamma_f}\mathbf{v}^T\mathbf{f}^S\,d\Gamma - \int_{\Gamma_u}\boldsymbol{\epsilon}(\mathbf{v})^T\mathbf{C}\mathbf{N}^T\mathbf{u}^S\,d\Gamma \quad \forall \mathbf{v}\in H^1(\Omega) \end{aligned} \tag{5.14}$$

where $H^1(\Omega)$ is the first order Hilbert space

5.3 Nodal interpolations

We have the following approximation for the displacement field

$$\mathbf{u}(x,y) = \sum_{J=1}^{N}\sum_{n\in\mathcal{I}}\mathbf{H}_{Jn}(x,y)\boldsymbol{\alpha}_{Jn} = \mathbf{H}(x,y)\mathbf{U}, \tag{5.15}$$

where

$$\mathbf{U} = [\boldsymbol{\alpha}_{10}\ \boldsymbol{\alpha}_{11}\ \boldsymbol{\alpha}_{12}\ \ldots\boldsymbol{\alpha}_{Jn}\ \ldots]^T$$

is the vector of nodal unknowns (not nodal displacements unless the Kronecker delta property is satisfied by the shape functions), and

$$\boldsymbol{\alpha}_{Jn} = [u^{Jn}\quad v^{Jn}]$$

is the vector of nodal unknowns at node J corresponding to the n^{th} degree of freedom (u^{Jn} and v^{Jn} are the nodal variables for the x and y direction displacements at node J corresponding to the n^{th} degree of freedom). The nodal shape function matrix corresponding to the n^{th} degree of freedom is

$$\mathbf{H}_{Jn}(x,y) = \begin{bmatrix} h_{Jn}(x,y) & 0 \\ 0 & h_{Jn}(x,y) \end{bmatrix}. \tag{5.16}$$

Hence, the discretized versions of equations (5.7) and (5.8) are

$$\boldsymbol{\epsilon}(x,y) = \sum_{J=1}^{N}\sum_{n\in\mathcal{I}}\mathbf{B}_{Jn}(x,y)\boldsymbol{\alpha}_{Jn} = \mathbf{B}(x,y)\mathbf{U} \tag{5.17}$$

and

$$\boldsymbol{\tau}(x,y) = \sum_{J=1}^{N} \sum_{n \in \mathcal{I}} \mathbf{CB}_{Jn}(x,y)\boldsymbol{\alpha}_{Jn} = \mathbf{CB}(x,y)\mathbf{U}, \tag{5.18}$$

where the strain-displacement matrix $\mathbf{B}(x,y)$ in equation (5.17) is partitioned as

$$\mathbf{B}(x,y) = [\mathbf{B}_{10}(x,y)\, \mathbf{B}_{11}(x,y)\, \ldots \mathbf{B}_{Jn}(x,y) \ldots]$$

where

$$\mathbf{B}_{Jn}(x,y) = \boldsymbol{\partial}_{\epsilon}\mathbf{H}_{Jn}(x,y) = \begin{bmatrix} \partial h_{Jn}/\partial x & 0 \\ 0 & \partial h_{Jn}/\partial y \\ \partial h_{Jn}/\partial y & \partial h_{Jn}/\partial x \end{bmatrix}. \tag{5.19}$$

5.4 Discrete equations

Using equations (5.15)-(5.18) in (5.14) we obtain the discretized system of algebraic equations corresponding to node I and degree of freedom m

$$\boxed{\sum_{J=1}^{N} \sum_{n \in \mathcal{I}} \mathbf{K}_{ImJn}\boldsymbol{\alpha}_{Jn} = \mathbf{f}_{Im} + \hat{\mathbf{f}}_{Im}} \tag{5.20}$$

In equation (5.20), the various matrices and vectors are as follows;

$$\mathbf{K}_{ImJn} = \int_{\Omega_I} \mathbf{B}_{Im}^{T}\mathbf{CB}_{Jn}\, d\Omega \tag{5.21}$$

$$\mathbf{f}_{Im} = \int_{\Omega_I} \mathbf{H}_{Im}\mathbf{f}^{B}\, d\Omega \tag{5.22}$$

where $\Omega_I = \Omega \cap B(\mathbf{x}_I, r_I)$.

If I is a node associated with an "internal sphere", then

$$\hat{\mathbf{f}}_{Im} = \mathbf{0}$$

from compact support.

If I is a node with an annular integration domain (see section 3.3), with internal boundary

Γ_{Ii} and external boundary Γ_{Io}, then

$$\hat{\mathbf{f}}_{Im} = \oint_{\Gamma_{Io}} \mathbf{H}_{Im}\mathbf{N}\tau \, d\Gamma + \oint_{\Gamma_{Ii}} \mathbf{H}_{Im}\mathbf{N}\tau \, d\Gamma. \tag{5.23}$$

If the sphere corresponding to node I has nonzero intercept on the Neumann boundary, then from equation (5.4),

$$\hat{\mathbf{f}}_{Im} = \int_{\Gamma_{f_I}} \mathbf{H}_{Im}\mathbf{f}^S \, d\Gamma \tag{5.24}$$

where $\Gamma_f = \cup_{I \in \mathcal{N}_f} \Gamma_{f_I}$, $\mathcal{N}_f$ being the index set of such nodes.

On the other hand, if the sphere corresponding to node I has nonzero intercept on the Dirichlet boundary, then

$$\hat{\mathbf{f}}_{Im} = \sum_{J=1}^{N} \sum_{n \in \mathcal{I}} \mathbf{KU}_{ImJn}\boldsymbol{\alpha}_{Jn} - \mathbf{fU}_{Im} \tag{5.25}$$

where

$$\mathbf{KU}_{ImJn} = \int_{\Gamma_{u_I}} \mathbf{H}_{Im}\mathbf{NCB}_{Jn} \, d\Gamma + \int_{\Gamma_{u_I}} \mathbf{B}_{Im}^T\mathbf{CN}^T\mathbf{H}_{Jn} \, d\Gamma \tag{5.26}$$

and

$$\mathbf{fU}_{Im} = \int_{\Gamma_{u_I}} \mathbf{B}_{Im}^T\mathbf{CN}^T\mathbf{u}^S \, d\Gamma \tag{5.27}$$

where $\Gamma_u = \cup_{I \in \mathcal{N}_u} \Gamma_{u_I}$, $\mathcal{N}_u$ being the index set of such nodes. Note that the stiffness matrix $\mathbf{KU}$ is symmetric.

A point to note is that we may incorporate the Dirichlet conditions by the special arrangement of nodes on the boundary as discussed in section 3.2.

5.5 Numerical results

The formulations presented in this chapter were used to solve typical problem in two-dimensional linear elastostatics. Patch tests were performed on a bi-unit square (both in plane stress and in plane strain conditions) and they were passed with as few as four nodes placed at the corners. The results of the patch tests are not shown here. However, two rep-

resentative example problems are presented. One involving a thick walled pressure vessel in plane stress conditions and the other involving a cantilever plate in plane strain conditions with a uniformly distributed loading.

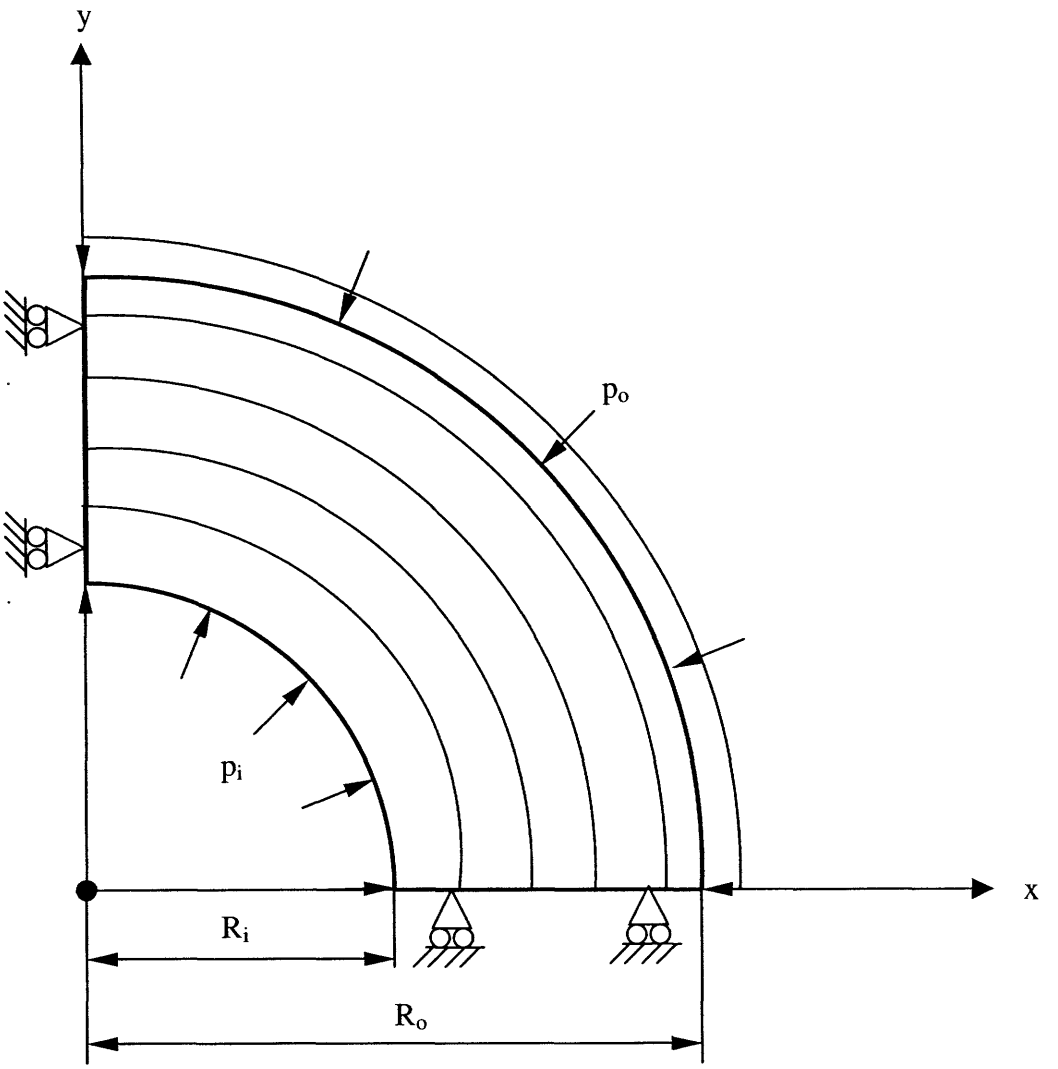


Figure 5-1: A quadrant of the thick-walled pressure vessel (in plane stress). All the nodes are placed at the origin of the coordinate system The integration domain corresponding to each node is an annular sector of inner radius R_i.

Thick walled pressure vessel in plane stress : We consider a thick-walled pressure vessel of external radius $R_o(= 10)$ and internal radius $R_i(= 5)$, subjected to uniform internal pressure p_i and external pressure p_o. The material properties of the cylinder wall are chosen as $E = 100$ and $\nu = 0.3$. As shown in Figure 5-1, one quadrant of the cylinder is discretized.

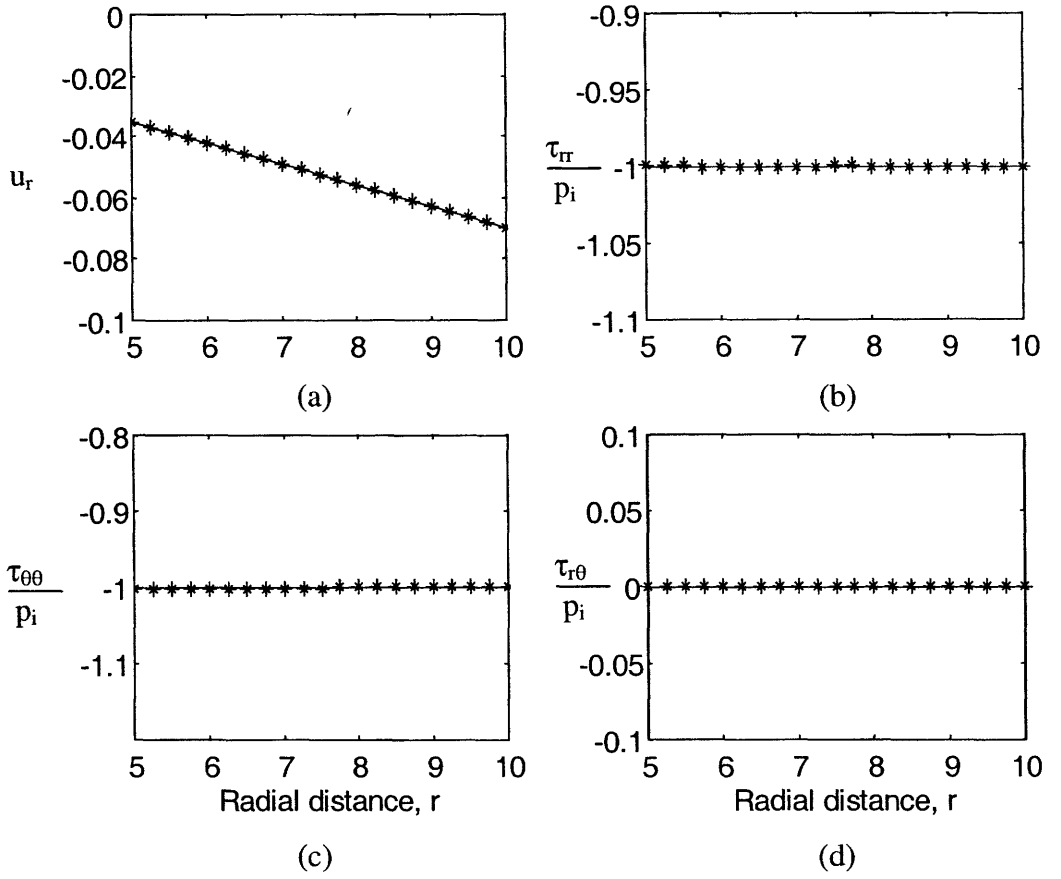


Figure 5-2: Results of the thick cylinder problem for $p_i = p_o = 1.0$. The radial displacement (u_r), the normalized radial stress (τ_{rr}), hoop stress ($\tau_{\theta\theta}$) and shear stress ($\tau_{r\theta}$) are plotted along a radius of the cylinder in (a), (b), (c) and (d) respectively. Continuous lines indicate analytical solution whereas the MFS solution is plotted with asterisks.

The nodes for all the disks are superimposed at the origin. All nodes are associated with annular integration domains, and the integration annuli have the same inner radius R_i but different outer radii. To be able to incorporate the Neumann boundary conditions at R_o, some annuli are chosen to have outer radii greater than R_o (see figure 5-1). Figures 5-2 and 5-3 present results of two numerical experiments.

In the first one (figure 5-2) $p_o = p_i = 1.0$. In the second experiment (figure 5-3) $p_i = 10$ and $p_o = 0$. The computed radial displacement field is plotted against the analytical solution in figure 5-3(a). Figure 5-3(b) shows the radial (σ_{rr}) and hoop ($\sigma_{\theta\theta}$) stresses normalized with the internal pressure (p_i). In this solution 12 nodes were used.

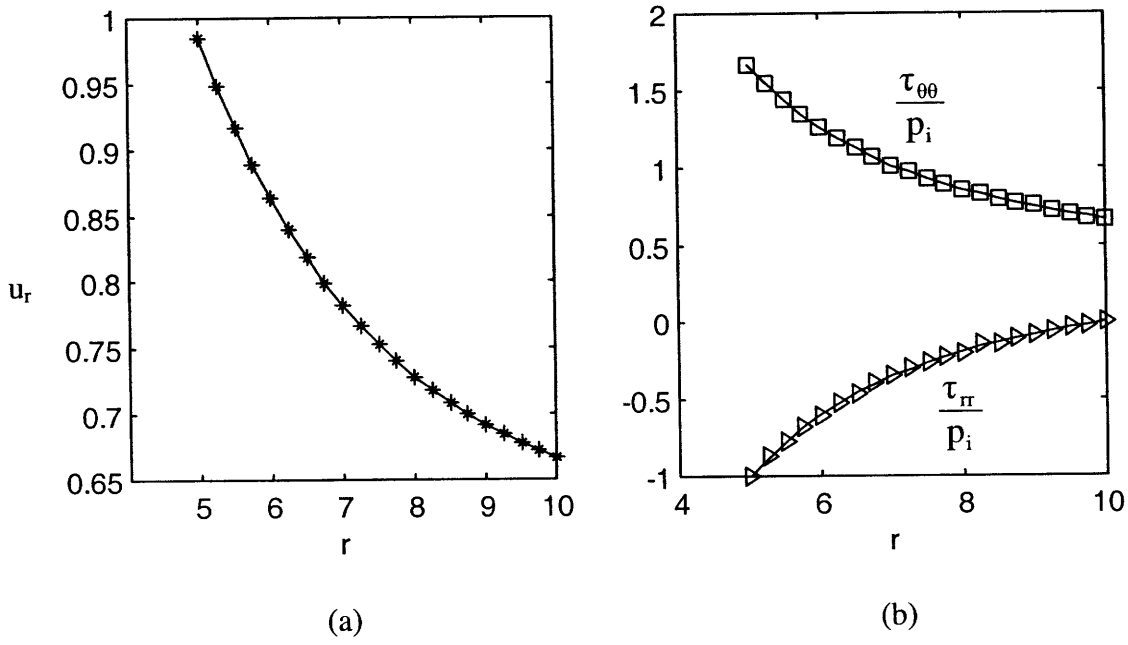


Figure 5-3: (a) Radial displacement field u_r and (b) normalized radial (τ_{rr}) and hoop ($\tau_{\theta\theta}$) stresses in the cylinder wall corresponding to $p_i = 10$ and $p_o = 0$. Continuous lines indicate analytical solution whereas the MFS solution is indicated by asterisks, triangles or squares.

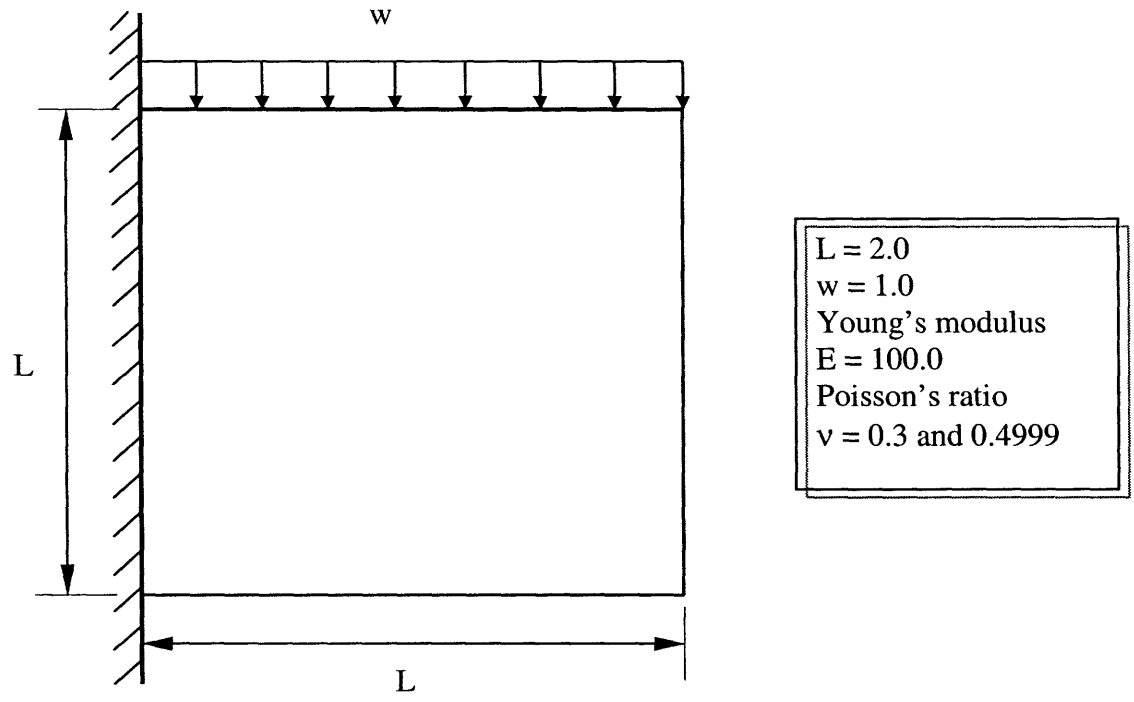
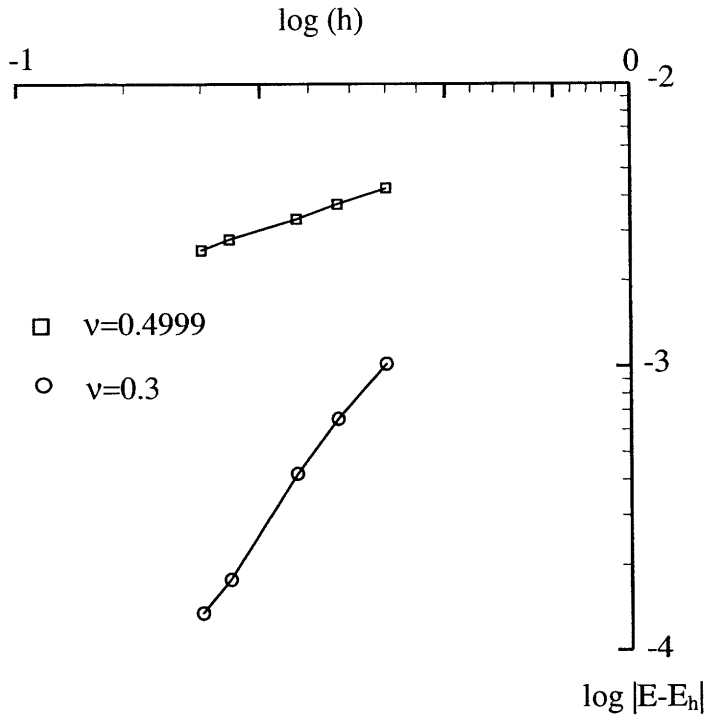


Figure 5-4: The cantilever problem considered in the text.

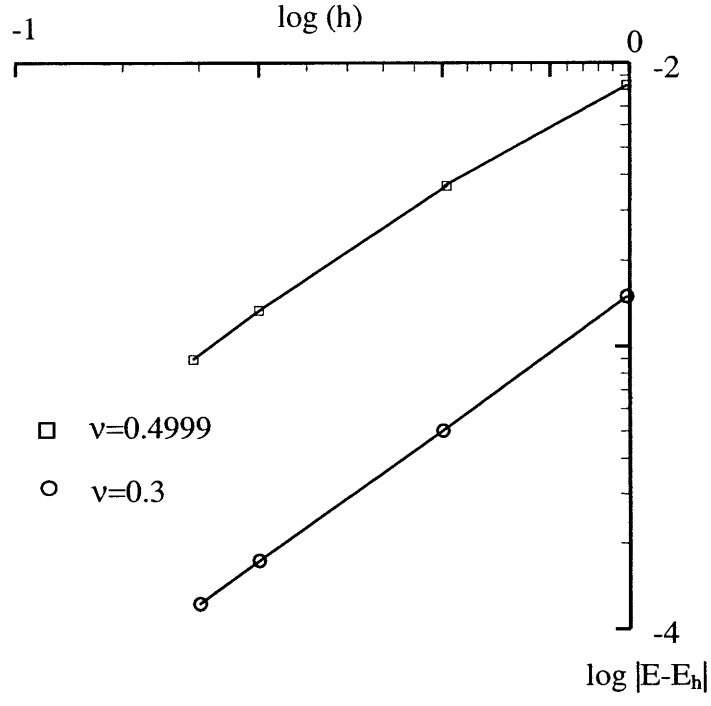
A cantilever plate in plane strain with uniformly distributed loading : In this problem we consider a cantilever plate in plane strain conditions as shown in figure 5-4. The material properties of the plate are chosen as Young's Modulus = 100 and Poisson's ratio $\nu = 0.3$ & 0.4999. A uniformly distributed load of magnitude $w = 1.0$ per unit length is applied as shown. Figure 5-5(a) shows the convergence in strain energy when a h-type refinement is performed using the method of finite spheres. In this analysis the local basis of each node has been chosen as a complete polynomial of second degree. The strain energy of the limit solution of the system is obtained by solving the same problem using a 50×50 mesh of 9-noded finite elements. Figure 5-5(b) shows the convergence curves when the same problem is solved using 9 -noded finite elements.

Two important observations can be made from figure 5-5. First, for a Poisson's ratio of 0.3 the method of finite spheres exhibits a much higher rate of convergence than the one obtained in the classical finite element analysis when ungraded meshes are used. This is due to the fact that continuous approximations are used in the method of finite spheres and might indicate a robustness of the method. The second observation is that when

the Poisson's ratio is increased to 0.4999, the finite element solution as well as the solution obtained from the method of finite spheres "lock", as evidenced by a deterioration in the rate of convergence. This implies that the displacement-based method of finite spheres suffers from exactly the same problem as the classical finite element techniques in the analysis of (almost) incompressible media. We develop a displacement/pressure mixed formulation in the next chapter to overcome the problem of locking.



(a)



(b)

Figure 5-5: Convergence in strain energy. (a) Method of finite spheres. (b) Finite element method (using 9-noded displacement-based finite elements).

Chapter 6

Displacement/pressure mixed formulation

In chapter 5 we observed that the displacement-based method of finite spheres exhibits volumetric locking in the case of almost or fully incompressible media. This case arises frequently in engineering applications, namely, during the modeling of rubber-like solids, isochoric plastic flow or the solution of the Stoke's problem in fluid mechanics. In this chapter we discuss this issue in greater detail.

The problem of "locking" is well known in the standard displacement-based finite element techniques [1]. The reason why a purely displacement-based formulation "locks" is the following. In a pure displacement-based formulation the computed displacement field needs to satisfy the constraint of very small volumetric strains (which become zero as the condition of total incompressibility is approached) while the pressure is of the order of the boundary tractions. The displacement approximation space is not rich enough to accommodate this constraint without a drastic reduction in the rate of convergence [1].

To overcome "locking" effectively, it is necessary to use a mixed formulation [1] in which separate approximation spaces are used for the displacement and pressure fields. While, in principle, numerous mixed formulations may be developed, only those that are stable are useful in practice. The solvability, stability and optimality of mixed formulations are expressed in the ellipticity requirement and the inf-sup condition [1]. The ellipticity condi-

tion is relatively easy to satisfy. The analytical proof whether the inf-sup condition is also satisfied for a specific formulation is, however, difficult. Hence a numerical inf-sup test was designed [1, 45].

Over the past decade several meshless techniques have been proposed but the issue of locking has not been studied in depth. Until recently, it was stated that meshfree methods are immune to locking. Indeed it was reported that the element free Galerkin method has the advantage that it does not suffer from locking [11, 12, 46, 47]. Furthermore, the element free Galerkin method has been actually proposed as a solution to the locking problem in isochoric elasto-plastic analyses especially when a sufficiently large support size for the interpolation functions is chosen [48].

In the context of another meshless scheme, the reproducing kernel particle method, a similar claim was made confirming the absence of volumetric locking when analyzing large deformation behavior of nearly incompressible hyperelastic materials [49]. More recently, Li and Liu [50] have reported that they have been able to simulate shear band formation successfully in inelastic solids using an explicit displacement-based formulation and the reproducing kernel particle method. They have linked the absence of locking to the choice of higher order polynomial interpolation and the use of reduced order integration. In these publications the conclusions were drawn primarily by only studying the analysis results of a few example problems, and not considering the inf-sup condition.

Only recently it has been reported that the element free Galerkin method does indeed suffer from locking in incompressible deformations [51]. The locking is pronounced when moving least squares interpolants with small support size are used. Functions with small support size are necessary in practice for better localization and a lower bandwidth of the system matrices. In their paper Dolbow and Belytschko [51] have proposed a mixed displacement/pressure formulation and selective reduced integration to alleviate locking. However, with a linear displacement field and a constant pressure field, the scheme does not pass the numerical inf-sup test. Chen et. al. note that the use of large support size is computationally expensive and, moreover, cannot remove pressure oscillations [52, 53]. They propose a pressure projection method to remove locking and pressure oscillations in nearly

incompressible finite elasticity problems together with a reduced integration scheme. No inf-sup tests were however performed to test the stability of the proposed scheme.

In this chapter we introduce displacement/pressure mixed formulations to overcome "locking" and test the stability and optimality of several discretization schemes using numerical inf-sup tests. Most of the work presented here has been reported in [58]. In section 6.1 we introduce a displacement/pressure mixed formulation for the analysis of problems involving linear elastic solids. In section 6.2 we discuss the inf-sup condition, introduce a numerical form and analyze the stability of several displacement/pressure discretization schemes. In section 6.3 we give numerical results.

6.1 Mixed displacement/pressure formulation

In chapter 5 the displacement-based or *primal formulation* of the elasticity problem was presented. By introducing pressure as an auxiliary variable, we obtain an equivalent *dual formulation:* the displacement/pressure mixed formulation. It should be noted, however, that this replaces the minimization problem (displacement-based formulation) by a saddle-point problem (the displacement/pressure mixed formulation). In this section we briefly review the governing equations and then derive a displacement/pressure mixed formulation and discretized equations.

6.1.1 Governing equations

Let us consider an open bounded domain $\Omega \subset R^2$. Let Γ be its boundary. The system of governing differential equations and the boundary conditions can be written as:

Equilibrium Equations :

$$\partial_\epsilon^T \boldsymbol{\tau} + \mathbf{f}^B = \mathbf{0} \qquad \text{in } \Omega \tag{6.1}$$

Strain – Displacement Relationships :

$$\boldsymbol{\epsilon} = \partial_\epsilon \mathbf{u} \qquad \text{in } \Omega \tag{6.2}$$

Boundary Conditions :

$$\mathbf{N}\boldsymbol{\tau} = \mathbf{f}^S \qquad \text{on } \Gamma_f \tag{6.3}$$

$$\mathbf{u} = \mathbf{u}^S \qquad \text{on } \Gamma_u \tag{6.4}$$

In equations (6.1)-(6.4), $\mathbf{u}$, $\boldsymbol{\epsilon}$ and $\boldsymbol{\tau}$ are the displacement, stress and strain vectors, $\mathbf{f}^S$ is the prescribed traction vector on the Neumann boundary Γ_f, $\mathbf{u}^S$ is the vector of prescribed displacements on the Dirichlet boundary Γ_u (note that the domain boundary $\Gamma = \Gamma_f \cup \Gamma_u$ and $\Gamma_f \cap \Gamma_u = \{0\}$), $\mathbf{f}^B$ is the body force vector (including inertia terms), $\boldsymbol{\partial}_\epsilon$ is a linear gradient operator and $\mathbf{N}$ is the matrix of direction cosine components of a unit normal to the domain boundary (positive outwards).

If the body is made of an almost incompressible medium, we anticipate that the volumetric strains will be small compared to the deviatoric strains and write the constitutive relationship in the following form

$$\boldsymbol{\tau} = -p\mathbf{I} + 2G\boldsymbol{\epsilon}^D \tag{6.5}$$

where $\mathbf{I}$ is the vector corresponding to the Kronecker delta, G is the shear modulus

$$G = \frac{E}{2(1+\nu)} \tag{6.6}$$

where E and ν are the Young's modulus and the Poisson ratio of the material, respectively. $\boldsymbol{\epsilon}^D$ is the vector of deviatoric strain components,

$$\boldsymbol{\epsilon}^D = \boldsymbol{\epsilon} - \frac{\epsilon_V}{3}\mathbf{I} \tag{6.7}$$

where ϵ_V is the volumetric strain,

$$\epsilon_V = \begin{cases} (\varepsilon_{xx} + \varepsilon_{yy}) & \text{for plane strain conditions,} \\ \frac{1-2\nu}{1-\nu}(\varepsilon_{xx} + \varepsilon_{yy}) & \text{for plane stress conditions.} \end{cases} \tag{6.8}$$

The pressure in the body is

$$p = -\kappa\epsilon_V \tag{6.9}$$

where the bulk modulus κ is

$$\kappa = \frac{E}{3(1-2\nu)}. \tag{6.10}$$

In addition, we note that the vector of deviatoric stresses

$$\boldsymbol{\tau}^D = \boldsymbol{\tau} + p\mathbf{I} \tag{6.11}$$

is related to vector of deviatoric strains by the following constitutive relationship

$$\boldsymbol{\tau}^D = \mathbf{C}^D \boldsymbol{\epsilon}^D. \tag{6.12}$$

6.1.2 Variational form

We consider the following variational indicator [1]

$$\Pi^*(\mathbf{u},p) = \int_\Omega \left[\frac{1}{2} \boldsymbol{\epsilon}^{D^T}(\mathbf{u}) \mathbf{C}^D \boldsymbol{\epsilon}^D(\mathbf{u}) - \frac{1}{2}\frac{p^2}{\kappa} - p\epsilon_V(\mathbf{u}) \right] d\Omega - \Re. \tag{6.13}$$

The term $\Re$ accounts for the externally applied body forces, surface tractions and applied displacements,

$$\Re = \int_\Omega \mathbf{u}^T \mathbf{f}^B d\Omega + \int_{\Gamma_f} \mathbf{u}^T \mathbf{f}^S d\Gamma + \int_{\Gamma_u} \mathbf{f}^{uT}(\mathbf{u} - \mathbf{u}^s) d\Gamma \tag{6.14}$$

where $\mathbf{f}^u$ is the traction vector on the Dirichlet boundary Γ_u and may be expressed as

$$\mathbf{f}^u = \mathbf{N}\mathbf{C}^D \boldsymbol{\epsilon}^D(\mathbf{u}) - p\mathbf{N}\mathbf{I}. \tag{6.15}$$

Here $\mathbf{f}^u$ is a vector of Lagrange multipliers to enforce the Dirichlet boundary conditions.

Invoking the stationarity of Π^* we obtain the following weak forms

$$\begin{aligned} &\int_\Omega \left[\boldsymbol{\epsilon}^{D^T}(\mathbf{v}) \mathbf{C}^D \boldsymbol{\epsilon}^D(\mathbf{u}) - \epsilon_V(\mathbf{v}) p \right] d\Omega \\ &- \int_{\Gamma_u} \left[\boldsymbol{\epsilon}^{D^T}(\mathbf{v}) \mathbf{C}^D \mathbf{N}^T \mathbf{u} + \mathbf{v}^T \mathbf{N} \mathbf{C}^D \boldsymbol{\epsilon}^D(\mathbf{u}) \right] d\Gamma + \int_{\Gamma_u} \mathbf{v}^T \mathbf{N} \mathbf{I} p \, d\Gamma \\ &= \int_\Omega \mathbf{v}^T \mathbf{f}^B d\Omega + \int_{\Gamma_f} \mathbf{v}^T \mathbf{f}^S d\Gamma - \int_{\Gamma_u} \boldsymbol{\epsilon}^{D^T}(\mathbf{v}) \mathbf{C}^D \mathbf{N}^T \mathbf{u}^S d\Gamma \quad \forall \mathbf{v} \in H^1(\Omega) \end{aligned} \tag{6.16}$$

and

$$-\int_\Omega q\left[\epsilon_V(\mathbf{u})+\tfrac{p}{\kappa}\right]d\Omega+\int_{\Gamma_u} q\mathbf{I}^T\mathbf{N}^T\mathbf{u}\,d\Gamma=\int_{\Gamma_u} q\mathbf{I}^T\mathbf{N}^T\mathbf{u}^S d\Gamma \qquad \forall q\in L^2(\Omega) \tag{6.17}$$

where and $H^1(\Omega)$ and $L^2(\Omega)$ are the first order Hilbert space and Lebesgue space of square integrable functions, respectively.

6.1.3 Nodal interpolations

We have the following approximation for the displacement field

$$\mathbf{u}(x,y)=\sum_{J=1}^{N}\sum_{n\in\mathcal{I}}\mathbf{H}_{Jn}(x,y)\boldsymbol{\alpha}_{Jn}=\mathbf{H}(x,y)\mathbf{U} \tag{6.18}$$

where $\mathbf{U}=[\boldsymbol{\alpha}_{10}\ \boldsymbol{\alpha}_{11}\ \boldsymbol{\alpha}_{12}\ \ldots\boldsymbol{\alpha}_{Jn}\ \ldots]^T$ is the vector of nodal unknowns and $\boldsymbol{\alpha}_{Jn}=[u^{Jn}\quad v^{Jn}]$ is the vector of nodal unknowns at node J corresponding to the n^{th} degree of freedom (u^{Jn} and v^{Jn} are the nodal variables for the x and y direction displacements at node J corresponding to the n^{th} degree of freedom). The shape function matrix corresponding to node J and the n^{th} degree of freedom is

$$\mathbf{H}_{Jn}(x,y)=\begin{bmatrix} h_{Jn}(x,y) & 0 \\ 0 & h_{Jn}(x,y)\end{bmatrix}.$$

We choose the following approximation for the pressure field

$$p(x,y)=\sum_{J=1}^{N}\sum_{n\in\mathcal{I}}h^p_{Jn}(x,y)p_{Jn}=\mathbf{H}_p(x,y)\mathbf{P}, \tag{6.19}$$

where $\mathbf{P}=[p_{10}\ p_{11}\ p_{12}\ \ldots p_{Jn}\ \ldots]^T$ is the vector of nodal point unknowns corresponding to the pressure degrees of freedom. The shape function $h^p_{Jn}(x,y)$ at node J corresponding to the n^{th} degree of freedom is also generated using the partition of unity paradigm. This construction of the pressure approximation space results in a continuous pressure field.

The approximations for the strains in equations (6.7) and (6.8) are

$$\epsilon^D(x,y) = \sum_{J=1}^{N} \sum_{n \in \mathcal{I}} \mathbf{B}^D_{Jn}(x,y)\boldsymbol{\alpha}_{Jn} = \mathbf{B}^D(x,y)\mathbf{U} \tag{6.20}$$

and

$$\epsilon_V(x,y) = \mathbf{B}_V(x,y)\mathbf{U}. \tag{6.21}$$

where $\mathbf{B}^D$ and $\mathbf{B}_V$ are the corresponding strain interpolation matrices.

6.1.4 Discrete equations

Using equations (6.18)-(6.21) in equations (6.16)-(6.17) we obtain the following discrete sets of equations corresponding to node I and degree of freedom m

$$\sum_{J=1}^{N} \sum_{n \in \mathcal{I}} \begin{bmatrix} \mathbf{K}_{uu_{ImJn}} & \mathbf{K}_{up_{ImJn}} \\ \mathbf{K}^T_{up_{ImJn}} & \mathbf{K}_{pp_{ImJn}} \end{bmatrix} \begin{Bmatrix} \boldsymbol{\alpha}_{Jn} \\ p_{Jn} \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_{Im} \\ \mathbf{0} \end{Bmatrix} + \hat{\mathbf{f}}_{Im} \tag{6.22}$$

where

$$\mathbf{K}_{uu_{ImJn}} = \int_{\Omega_I} \mathbf{B}^{DT}_{Im} \mathbf{C}^D \mathbf{B}^D_{Jn} \, d\Omega, \tag{6.23}$$

$$\mathbf{K}_{up_{ImJn}} = -\int_{\Omega_I} \mathbf{B}^T_{VIm} h^p_{Jn} \, d\Omega, \tag{6.24}$$

$$\mathbf{K}_{pp_{ImJn}} = -\frac{1}{\kappa} \int_{\Omega_I} h^p_{Im} h^p_{Jn} \, d\Omega \tag{6.25}$$

and

$$\mathbf{f}_{Im} = \int_{\Omega_I} \mathbf{H}_{Im} \mathbf{f}^B \, d\Omega \tag{6.26}$$

where $\Omega_I = \Omega \cap B(\mathbf{x}_I, r_I)$. If I is a node associated with an "interior sphere", then

$$\hat{\mathbf{f}}_{Im} = \mathbf{0}$$

from the property of compact support. If I is a node associated with a "boundary sphere" then $\hat{\mathbf{f}}_{Im}$ allows us to incorporate the prescribed boundary conditions.

If the sphere corresponding to node I has a nonzero intercept on the Neumann boundary

Γ_f, then

$$\hat{\mathbf{f}}_{Im} = \left\{ \begin{array}{c} \int_{\Gamma_{f_I}} \mathbf{H}_{Im} \mathbf{f}^S \, d\Gamma \\ \mathbf{0} \end{array} \right\} \tag{6.27}$$

where $\Gamma_f = \cup_{I \in \mathcal{N}_f} \Gamma_{f_I}$; $\mathcal{N}_f$ being the index set of such nodes.

On the other hand, if the sphere corresponding to node I has a nonzero intercept on the Dirichlet boundary Γ_u, then

$$\hat{\mathbf{f}}_{Im} = \sum_{J=1}^{N} \sum_{n \in \mathcal{I}} \left[\begin{array}{cc} \mathbf{KU}_{uu_{ImJn}} & \mathbf{KU}_{up_{ImJn}} \\ \mathbf{KU}^T_{up_{ImJn}} & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \boldsymbol{\alpha}_{Jn} \\ p_{Jn} \end{array} \right\} - \left\{ \begin{array}{c} \mathbf{fU}_{u_{Im}} \\ \mathbf{fU}_{p_{Im}} \end{array} \right\} \tag{6.28}$$

where

$$\mathbf{KU}_{uu_{ImJn}} = \int_{\Gamma_{u_I}} \mathbf{H}_{Im} \mathbf{N} \mathbf{C}^D \mathbf{B}^D_{Jn} \, d\Gamma + \int_{\Gamma_{u_I}} \mathbf{B}^{DT}_{Im} \mathbf{C}^D \mathbf{N}^T \mathbf{H}_{Jn} \, d\Gamma, \tag{6.29}$$

$$\mathbf{KU}_{up_{ImJn}} = - \int_{\Gamma_{u_I}} \mathbf{H}_{Im} \mathbf{N} \mathbf{I} h^p_{Jn} \, d\Gamma, \tag{6.30}$$

$$\mathbf{fU}_{u_{Im}} = \int_{\Gamma_{u_I}} \mathbf{B}^{DT}_{Im} \mathbf{C}^D \mathbf{N}^T \mathbf{u}^S \, d\Gamma, \tag{6.31}$$

and

$$\mathbf{fU}_{p_{Im}} = - \int_{\Gamma_{u_I}} h^p_{Im} \mathbf{I}^T \mathbf{N}^T \mathbf{u}^S \, d\Gamma \tag{6.32}$$

where $\Gamma_u = \cup_{I \in \mathcal{N}_u} \Gamma_{u_I}$; $\mathcal{N}_u$ being the index set of such nodes.

6.2 Inf-sup tests

While a pure displacement-based formulation is always stable but "locks", a displacement/pressure mixed formulation relieves locking, but is not stable for arbitrary choice of the displacement and pressure approximation spaces. For such a formulation to be reliable for general use, the issue of stability is very important. In this section we state the necessary and sufficient conditions for the stability of displacement/pressure mixed formulations (for details please refer to [1]). These conditions also ensure solvability of the discretized set of equations and optimal convergence rates (for given choices of displacement and pressure approximation spaces).

6.2.1 Inf-sup condition

To keep the discussion simple, let us consider displacement approximation spaces $V_h \subset H_o^1(\Omega)$ (H_o^1 contains functions that are in the first order Hilbert space and satisfy the homogeneous Dirichlet boundary conditions of the problem). $Q_h \subset L^2(\Omega)$ is the usual pressure approximation space. Then, equations (6.16) and (6.17) can be written compactly as

Find $\mathbf{u}_h \in V_h$ and $p_h \in Q_h$ such that

$$\begin{aligned} &\int_\Omega \boldsymbol{\epsilon}^{D^T}(\mathbf{v}_h)\mathbf{C}^D\boldsymbol{\epsilon}^D(\mathbf{u}_h)\,d\Omega - \int_\Omega p_h\, div\,\mathbf{v}_h\; d\Omega \\ &= \int_\Omega \mathbf{v}_h^T\mathbf{f}^B d\Omega + \int_{\Gamma_f} \mathbf{v}_h^T\mathbf{f}^S d\Gamma \quad \forall \mathbf{v}_h \in V_h \end{aligned} \tag{6.33}$$

and

$$-\int_\Omega q_h \left[div\,\mathbf{u}_h + \tfrac{p_h}{\kappa}\right] d\Omega = 0 \qquad \forall q_h \in Q_h \tag{6.34}$$

The corresponding matrix problem is of the following type

$$\begin{bmatrix} (\mathbf{K}_{uu})_h & (\mathbf{K}_{up})_h \\ (\mathbf{K}_{up})_h^T & -\frac{1}{\kappa}\mathbf{T}_h \end{bmatrix} \begin{Bmatrix} \mathbf{U}_h \\ \mathbf{P}_h \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_h \\ \mathbf{0} \end{Bmatrix} \tag{6.35}$$

where

$$\mathbf{T}_{h_{ImJn}} = \int_{\Omega_I} h_{Im}^p h_{Jn}^p\; d\Omega. \tag{6.36}$$

Let P_h be a L^2-projector onto Q_h, i.e.

$$\int_\Omega q_h(P_h(div\,\mathbf{w}_h) - div\,\mathbf{w}_h)\,d\Omega = 0 \qquad \forall \mathbf{w}_h \in V_h \text{ and } \forall q_h \in Q_h \tag{6.37}$$

Comparing equations (6.34) and (6.37) we recognize that

$$p_h = -\kappa\, P_h(div\,\mathbf{u}_h) \tag{6.38}$$

and comparing this with the discretized relation $(\mathbf{K}_{up})_h^T\,\mathbf{U}_h - \frac{1}{\kappa}\mathbf{T}_h\mathbf{P}_h = \mathbf{0}$ in (6.35), we observe that $-\mathbf{T}_h^{-1}\,(\mathbf{K}_{up})_h^T$ is the discrete form of the operator $P_h(div)$.

To obtain a stable and optimal procedure for the selected interpolations, the mixed formulation in equations (6.33)-(6.34) should necessarily satisfy the following two conditions [1, 45]

(i) **Ellipticity condition:** $\exists\, \alpha > 0$ such that

$$\int_\Omega \boldsymbol{\epsilon}^{D^T}(\mathbf{w}_h)\mathbf{C}^D\boldsymbol{\epsilon}^D(\mathbf{w}_h)\, d\Omega \geq \alpha \parallel \mathbf{w}_h \parallel_1^2$$

$$\forall \mathbf{w}_h \text{ such that } \int_\Omega q_h\, div\, \mathbf{w}_h\, d\Omega = 0\ \forall q_h \in Q_h$$

This condition is readily satisfied in this linear analysis since no reduced integration is used.

(ii) **Inf-sup condition** $\exists\, \gamma > 0$, independent of h such that

$$\inf_{q_h \in Q_h} \sup_{\mathbf{v}_h \in V_h} \frac{\int_\Omega q_h\, div \mathbf{v}_h\, d\Omega}{\parallel q_h \parallel_0 \parallel \mathbf{v}_h \parallel_1} = \gamma_h \geq \gamma \tag{6.39}$$

The norms $\parallel \cdot \parallel_0$ and $\parallel \cdot \parallel_1$ are defined as

$$\parallel \cdot \parallel_0^2 = \int_\Omega (\cdot)^2\, d\Omega \tag{6.40}$$

and

$$\parallel \cdot \parallel_1^2 = \int_\Omega \sum_{i,j=1}^{2} \left(\frac{\partial (\cdot)_i}{\partial x_j} \right)^2 d\Omega \tag{6.41}$$

PROPOSITION 6.1 *When the inf-sup condition (6.39) holds (and the ellipticity condition is satisfied), the following relation can be established [45]*

$$\parallel \mathbf{u} - \mathbf{u}_h \parallel_1 + \parallel p - p_h \parallel_0 \leq C \inf_{\mathbf{v}_h \in V_h, q_h \in Q_h} (\parallel \mathbf{u} - \mathbf{v}_h \parallel_1 + \parallel p - q_h \parallel_0)$$

The constant C depends on the problem considered (the solution) and γ (actually as the inverse of γ) and is independent of h and crucial physical parameters of the system (like the bulk modulus). This condition *strongly excludes "locking"* by ensuring that the rate of convergence is independent of the crucial physical parameters of the system and is optimal for the pressure and displacement approximation spaces chosen.

Let us now define the divergence space

$$D_h = \{div\, \mathbf{w}_h \mid \mathbf{w}_h \in V_h\}.$$

We notice that $P_h(D_h) \subset Q_h$. If $P_h(D_h)$ is strictly included in Q_h then

$$\exists q_h \in P_h^{\perp}(D_h) \text{ such that } \int_\Omega q_h\, div\, \mathbf{v}_h = \int_\Omega q_h\, P_h(div\, \mathbf{v}_h) = 0 \quad \forall \mathbf{v}_h \in V_h$$

where $P_h^{\perp}(D_h)$ is the orthogonal subspace to $P_h(D_h)$ in Q_h. Hence the inf-sup parameter γ_h in (6.39) is zero and therefore the inf-sup condition is of course not satisfied. The elements in $P_h^{\perp}(D_h)$ are referred to as the *spurious pressure modes*. Spurious pressure modes are not desirable because of the following reasons (see [1] for further details)

(i) In the discretized set of equations, *spurious pressure modes belong to the null space of* $(\mathbf{K}_{up})_h$ and for the case of total incompressibility ($\kappa \to \infty$) they render the system matrix in equation (6.35) singular.

(ii) For finite κ, large spurious pressures may be generated if non-homogeneous boundary conditions are applied.

It should be noted, however, that so long as $Q_h = P_h(D_h)$, there can be no spurious pressure modes, since $\forall q_h \in P_h(D_h)\ \exists\, \mathbf{v}_h \in V_h$ such that $q_h = P_h(div\, \mathbf{v}_h)$ and therefore

$$\int_\Omega q_h\, div\, \mathbf{v}_h = \int_\Omega q_h\, P_h(div\, \mathbf{v}_h) = \| q_h \|_0^2 > 0.$$

Therefore the inf-sup condition (6.39) may be used to test the following

1. *Spurious pressure modes* if tested with $q_h \in Q_h$, and
2. *Locking* if tested with $q_h \in P_h(D_h)$ and the inf-sup parameter $\gamma_h \to 0$ with increase in refinement.

At this point it is important to consider the rank verification test prescribed by Zienkiewicz [55] and make the following remark

Remark 6.1 *The rank verification test prescribed by Zienkiewicz [55] for the "stability" of the system (6.35) is just a necessary condition for the solvability of the system equations*

under the condition of complete incompressibility and does not address the issue of stability of the mixed form.

To see this, let us assume that in the method of finite spheres using N nodes there are n_u displacement degrees of freedom and n_p pressure degrees of freedom per node. Then, $\mathbf{K}_{uu} \in R^{Nn_u \times Nn_u}$, $\mathbf{K}_{up} \in R^{Nn_u \times Nn_p}$ and $\mathbf{K}_{pp} \in R^{Nn_p \times Nn_p}$.Therefore, the constraint ratio

$$\frac{\text{Total number of displacement degress of freedom}}{\text{Total number of pressure degress of freedom}} = \frac{n_u}{n_p}$$

is *independent of the number of nodes* used in the discretization. This ratio is > 1 as long as $n_u > n_p$. This is true, for example, when a first degree polynomial approximation is used for the displacement and pressure approximation spaces ($n_u = 6$ and $n_p = 3$, in R^2 for this case). But, as we shall see in the next section, such a discretization scheme is not stable.

However, the rank verification test does provide a necessary condition for the solvability of the system of equations for the case of complete incompressibility. We have from equation (6.35)

$$\begin{bmatrix} (\mathbf{K}_{uu})_h & (\mathbf{K}_{up})_h \\ (\mathbf{K}_{up})_h^T & \mathbf{0} \end{bmatrix} \begin{Bmatrix} \mathbf{U}_h \\ \mathbf{P}_h \end{Bmatrix} = \begin{Bmatrix} \mathbf{f}_h \\ \mathbf{0} \end{Bmatrix}. \tag{6.42}$$

It is straightforward to show that the *necessary and sufficient* conditions for *solvability* of this system are

(i) $\mathbf{V}_h^T (\mathbf{K}_{uu})_h \mathbf{V}_h > 0 \quad \forall \mathbf{V}_h \in R^{Nn_u}$ satisfying $(\mathbf{K}_{up})_h^T \mathbf{V}_h = 0$; i.e. $(\mathbf{K}_{uu})_h$ must be elliptic on the kernel of $(\mathbf{K}_{up})_h^T$.

(ii) $(\mathbf{K}_{up})_h \mathbf{Q}_h = 0 \Longleftrightarrow \mathbf{Q}_h = \mathbf{0}$; i.e. the matrix $(\mathbf{K}_{up})_h$ must be full rank.

For $(\mathbf{K}_{up})_h^T \in R^{Nn_p \times Nn_u}$ to be rank deficient (the actual displacement solution to the problem lies in the space spanned by the vectors in the null space of $(\mathbf{K}_{up})_h^T$) and $(\mathbf{K}_{up})_h \in R^{Nn_u \times Nn_p}$ to be full rank, it is certainly *necessary* to have $n_u > n_p$.

6.2.2 Numerical inf-sup test

While it is highly desirable to obtain an analytical proof that a given discretization scheme satisfies the inf-sup condition (6.39), such proofs are quite difficult due to the complex (rational) nature of the interpolation functions used in the method of finite spheres. Hence we adopt a numerical inf-sup test [1].

In the numerical inf-sup test, the inf-sup parameter γ_h in (6.39) is computed for a sequence of discretizations of a model problem. A mixed formulation with given displacement/pressure interpolations is said to have passed the inf-sup test if the inf-sup parameter asymptotically approaches a positive value greater than zero as the discretization is refined. The numerical inf-sup test is used for mixed displacement/pressure formulations just as the numerical patch test is used for incompatible displacement discretizations. Our experience is that if the inf-sup test is passed for a well-chosen problem, the inf-sup condition is satisfied.

In order to obtain the inf-sup parameter, γ_h, numerically for a given discretization, we express the relationship (6.39) in matrix form

$$\inf_{\mathbf{W}_h} \sup_{\mathbf{V}_h} \frac{\mathbf{W}_h^T \mathbf{G}_h \mathbf{V}_h}{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h}\sqrt{\mathbf{V}_h^T \mathbf{S}_h \mathbf{V}_h}} = \gamma_h \geq \gamma > 0 \tag{6.43}$$

with

$$\| \mathbf{v}_h \|_1^2 = \mathbf{V}_h^T \mathbf{S}_h \mathbf{V}_h. \tag{6.44}$$

where $\mathbf{W}_h \in R^{Nn_u}$ and $\mathbf{V}_h \in R^{Nn_u}$ are vectors of nodal parameters. $\mathbf{G}_h$ is defined by

$$\mathbf{W}_h^T \mathbf{G}_h \mathbf{V}_h = \int_\Omega q_h \, div \, \mathbf{v}_h \, d\Omega.$$

If $q_h \in P_h(D_h)$, then $\exists \, \mathbf{w}_h \in V_h$ such that $q_h = P_h(div \, \mathbf{w}_h)$. Therefore

$$\begin{aligned} \int_\Omega q_h \, div \, \mathbf{v}_h \, d\Omega &= \int_\Omega P_h(div \, \mathbf{w}_h) \, div \, \mathbf{v}_h \, d\Omega \\ &= \int_\Omega P_h(div \, \mathbf{w}_h) \, P_h(div \, \mathbf{v}_h) \, d\Omega \\ &= \int_\Omega div \, \mathbf{w}_h P_h(\, div \, \mathbf{v}_h) \, d\Omega \end{aligned}$$

and hence

$$\mathbf{G}_h = (\mathbf{K}_{up})_h \, \mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \tag{6.45}$$

can be directly deduced from the observation that $-\mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T$ is the discrete form of the operator $P_h(div)$, whereas the operator div has the discrete form $-(\mathbf{K}_{up})_h^T$. Moreover,

$$\| \, q_h \, \|_0^2 = \int_\Omega (P_h(div \, \mathbf{w}_h))^2 \, d\Omega = \mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h \tag{6.46}$$

We observe that the matrices $\mathbf{S}_h$ and $\mathbf{T}_h$ are *symmetric* and *positive definite.* On the other hand, $\mathbf{G}_h$ is *symmetric* but *positive semi-definite.* To see that the last statement is true, let us consider the quadratic form

$$\begin{aligned} \mathbf{V}_h^T \mathbf{G}_h \mathbf{V}_h &= \mathbf{V}_h^T \, (\mathbf{K}_{up})_h \, \mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \, \mathbf{V}_h \\ &= \left((\mathbf{K}_{up})_h^T \, \mathbf{V}_h \right)^T \mathbf{T}_h^{-1} \left((\mathbf{K}_{up})_h^T \, \mathbf{V}_h \right). \end{aligned}$$

Since the solvability condition for the system (6.42) requires that $(\mathbf{K}_{up})_h^T$ have a nontrivial null space, $\mathbf{V}_h^T \mathbf{G}_h \mathbf{V}_h \geq 0$.

We consider a sequence of discretizations in which all the displacement degrees of freedom corresponding to spheres that have nonzero intercepts with the Dirichlet boundary are set to zero. For each discretization, the computation of the inf-sup parameter γ_h is made possible by the following results [1]

PROPOSITION 6.2 *The inf-sup parameter* $\gamma_h = \sqrt{\lambda_p}$, *where* λ_p *is the minimum nonzero eigenvalue of the generalized eigenvalue problem*

$$\mathbf{G}_h \boldsymbol{\phi}_h = \lambda \mathbf{S}_h \boldsymbol{\phi}_h \tag{6.47}$$

Proof: Let us define

$$f(\mathbf{W}_h, \mathbf{V}_h) = \frac{\mathbf{W}_h^T \mathbf{G}_h \mathbf{V}_h}{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h} \sqrt{\mathbf{V}_h^T \mathbf{S}_h \mathbf{V}_h}}$$

and let

$$\mathbf{S}_h = \mathbf{L}_h^T \mathbf{L}_h$$

be the Cholesky decomposition of $\mathbf{S}_h$. We also define $\mathbf{X}_h = \mathbf{L}_h \mathbf{V}_h$. Hence,

$$f(\mathbf{W}_h, \mathbf{V}_h) = \frac{\mathbf{X}_h^T \mathbf{L}_h^{-T} \mathbf{G}_h \mathbf{W}_h}{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h} \sqrt{\mathbf{X}_h^T \mathbf{X}_h}}$$

We now use Cauchy-Schwarz inequality

$$\left| \mathbf{X}_h^T \mathbf{L}_h^{-T} \mathbf{G}_h \mathbf{W}_h \right| \leq \|\mathbf{X}_h\| \ \left\| \mathbf{L}_h^{-T} \mathbf{G}_h \mathbf{W}_h \right\|$$

where the Euclidean norm $\|\mathbf{X}_h\| = \sqrt{\mathbf{X}_h^T \mathbf{X}_h}$. Therefore

$$\begin{aligned} f(\mathbf{W}_h, \mathbf{V}_h) &\leq \frac{\left\| \mathbf{L}_h^{-T} \mathbf{G}_h \mathbf{W}_h \right\|}{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h}} \\ &= \frac{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{L}_h^{-1} \mathbf{L}_h^{-T} \mathbf{G}_h \mathbf{W}_h}}{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h}} \\ &= \frac{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{S}_h^{-1} \mathbf{G}_h \mathbf{W}_h}}{\sqrt{\mathbf{W}_h^T \mathbf{G}_h \mathbf{W}_h}} \end{aligned}$$

If λ_p is the minimum nonzero eigenvalue in (6.47) then

$$\inf_{\mathbf{W}_h} \sup_{\mathbf{V}_h} f(\mathbf{W}_h, \mathbf{V}_h) = \sqrt{\lambda_p}.$$

PROPOSITION 6.3 *The inf-sup parameter $\gamma_h = \sqrt{\lambda_p}$, where λ_p is the minimum nonzero eigenvalue of the generalized eigenvalue problem*

$$\mathbf{G}_h' \boldsymbol{\psi}_h = \lambda \mathbf{T}_h \boldsymbol{\psi}_h \tag{6.48}$$

where

$$\mathbf{G}_h' = (\mathbf{K}_{up})_h^T \, \mathbf{S}_h^{-1} \, (\mathbf{K}_{up})_h \, . \tag{6.49}$$

Proof: From (6.47)

$$\mathbf{G}_h \boldsymbol{\phi}_h = \lambda \mathbf{S}_h \boldsymbol{\phi}_h.$$

But $\mathbf{G}_h = (\mathbf{K}_{up})_h \, \mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T$, hence

$$\begin{aligned} (\mathbf{K}_{up})_h \, \mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \, \boldsymbol{\phi}_h &= \lambda \mathbf{S}_h \boldsymbol{\phi}_h \\ (\mathbf{K}_{up})_h^T \, \mathbf{S}_h^{-1} \, (\mathbf{K}_{up})_h \, \mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \, \boldsymbol{\phi}_h &= \lambda \, (\mathbf{K}_{up})_h^T \, \boldsymbol{\phi}_h \\ \mathbf{G}_h' \left(\mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \, \boldsymbol{\phi}_h \right) &= \lambda \mathbf{T}_h \left(\mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \, \boldsymbol{\phi}_h \right). \end{aligned}$$

Defining $\boldsymbol{\psi}_h = \left(\mathbf{T}_h^{-1} \, (\mathbf{K}_{up})_h^T \, \boldsymbol{\phi}_h \right)$, we obtain the eigenvalue problem (6.48).

The eigenvalue problems (6.47) and (6.48) have the same nonzero eigenvalues. However, there are two advantages to using (6.48);

(i) Eigenvalue problem (6.48) is posed in the pressure space ($\boldsymbol{\psi}_h \in R^{Nn_p}$) and since the pressure degrees of freedom are less than the displacement degrees of freedom, the size of the eigenvalue problem that needs to be solved is smaller.

(ii) The matrix $\mathbf{G}_h'$ is *symmetric*. For $\mathbf{Q}_h \in R^{Nn_p}$

$$\begin{aligned} \mathbf{Q}_h^T \mathbf{G}_h' \mathbf{Q}_h &= \mathbf{Q}_h^T \, (\mathbf{K}_{up})_h^T \, \mathbf{S}_h^{-1} \, (\mathbf{K}_{up})_h \, \mathbf{Q}_h \\ &= \left((\mathbf{K}_{up})_h \, \mathbf{Q}_h \right)^T \mathbf{S}_h^{-1} \left((\mathbf{K}_{up})_h \, \mathbf{Q}_h \right). \end{aligned}$$

Therefore, *in the absence of spurious pressure modes*, $(\mathbf{K}_{up})_h$ has a trivial null space (as opposed to $(\mathbf{K}_{up})_h^T$) and $\mathbf{G}_h'$ is *positive definite* and it is sufficient to search for the minimum eigenvalue of (6.48). The number of zero eigenvalues, if any, indicates the number of spurious pressure modes.

6.3 Results

In the finite element method, the inf-sup test has been employed successfully in identifying specific displacement/pressure mixed interpolation schemes that are stable and result in optimal convergence [1]. We follow a similar approach in the context of the method of finite spheres and identify useful displacement/pressure interpolation schemes. The numerical inf-sup test is used to test several discretizations in two-dimensional plane strain analysis. Throughout the tests, the same system is used: the simple cantilevered square block, shown in figure 6-1. This problem has been used to identify several effective finite element dis-

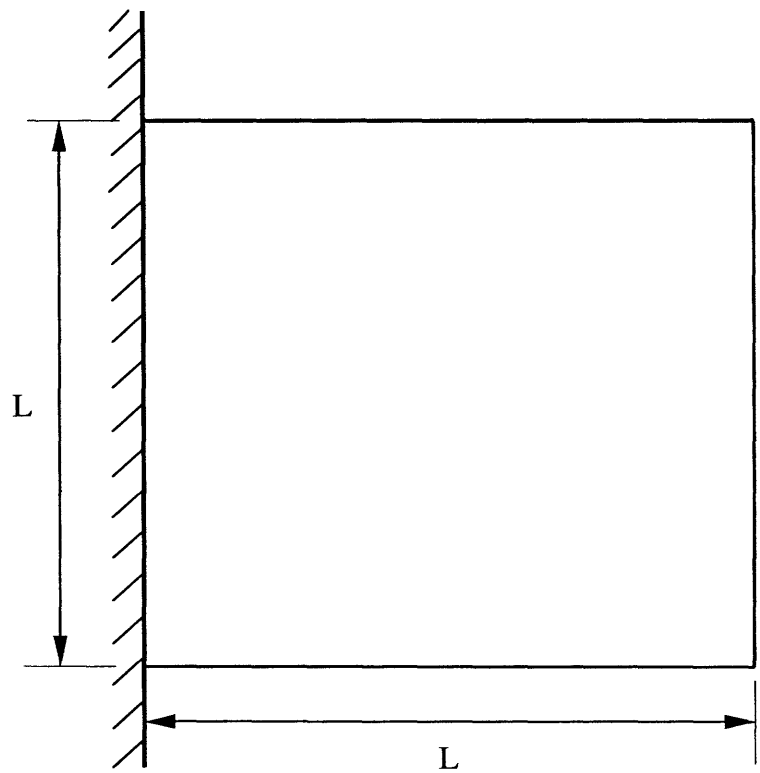


Figure 6-1: Problem considered for the inf-sup experiments: a cantilever plate (L=2.0) in plane strain.

cretization schemes [1]. In every instance, the domain is discretized using a sequence of regular arrangement of nodes. The results are plotted in the form $\log(\gamma_h) = f(\log h)$, where γ_h is the inf-sup parameter and h is the radius of each sphere.

In the finite element technique regular and irregular discretizations have been widely used for numerical inf-sup tests [1, 56]. However, in the method of finite spheres, the approximation space corresponding to a finer discretization does not contain the approximation space corresponding to a previous coarser discretization, and hence using even a regular grid is quite a severe test. But, of course, additional studies using irregular nodal arrangements would be valuable.

In order to satisfy the inf-sup condition (6.39), the inf-sup parameter must be bounded above zero with increase in refinement. Therefore, when a steady decrease of $\log(\gamma_h)$ is observed on the graph, the discretization scheme is said to fail the inf-sup test, whereas, if the inf-sup value approaches a value greater than zero with increase in refinement, the test is passed.

We focus attention on polynomial local approximation spaces. To label the various mixed interpolation function spaces we introduce the following notation. Let P_n and Q_n denote, respectively, the space of complete polynomials of degree 'n' and tensor product polynomials of degree 'n' in R^2 (e.g. $P_1 = span\{1, x, y\}$ and $Q_1 = span\{1, x, y, xy\}$). In the method of finite spheres we use the approximation spaces $P_n^S = \sum_{i=1}^N \varphi_i^0 P_n$ and $Q_n^S = \sum_{i=1}^N \varphi_i^0 Q_n$, and refer to a mixed interpolation scheme using, for example, P_n^S for displacement approximation and Q_n^S for the pressure approximation simply as the 'P_n^S/Q_n^S interpolation'.

We have considered seven different displacement/pressure discretization schemes. Figures 6-2 to 6-4 show the numerical results. The conclusions regarding whether the inf-sup test is passed or not are readily drawn and are summarized in Table 6.1. It is interesting to observe that while the inf-sup parameter corresponding to certain discretizations that fail the inf-sup test (e.g. the P_1^S/P_1^S discretization) steadily decreases, the inf-sup parameter corresponding to certain other discretizations (e.g. the P_2^S/P_2^S discretization) initially increases and then starts to decrease. Another interesting point to note is that while in the

Table 6.1: Inf-sup numerical predictions

Discretization scheme	Numerical inf-sup prediction
P_1^S/P_0^S	PASS
P_1^S/P_1^S	FAIL
P_1^S/Q_1^S	FAIL
P_2^S/P_0^S	PASS
P_2^S/P_1^S	PASS
P_2^S/Q_1^S	PASS
P_2^S/P_2^S	FAIL

finite element context, the simple 3/1 (P_1/P_0) element fails the inf-sup test, the P_1^S/P_0^S discretization scheme passes the test.

Once more we consider the problem in figure 5-1. Figure 6-5 shows the convergence in strain energy when a uniform h-type refinement is performed corresponding to two values of the Poisson's ratio, ν, equal to 0.3 and 0.4999. We observe that the P_2^S/Q_1^S displacement/pressure mixed discretization scheme alleviates locking as expected (see Table 6.1) and results in optimal convergence rate. The strain energy of the reference solution was obtained from a finite element analysis of the same problem using a 50×50 mesh of nine noded finite elements (9/3 displacement/pressure elements for the nearly incompressible case [1]). Figure 6-6 shows the computed pressure over the domain when two different mixed discretization schemes, P_1^S/P_0^S and P_1^S/P_1^S are used. Note that the discretization scheme P_1^S/P_0^S which passes the inf-sup test, results in a smooth pressure plot, while the pressure plot obtained by using a P_1^S/P_1^S discretization scheme, which fails the numerical inf-sup test, is very irregular.

Another important observation is that in none of the test cases zero eigenvalues were encountered indicating that no spurious pressure modes were present.

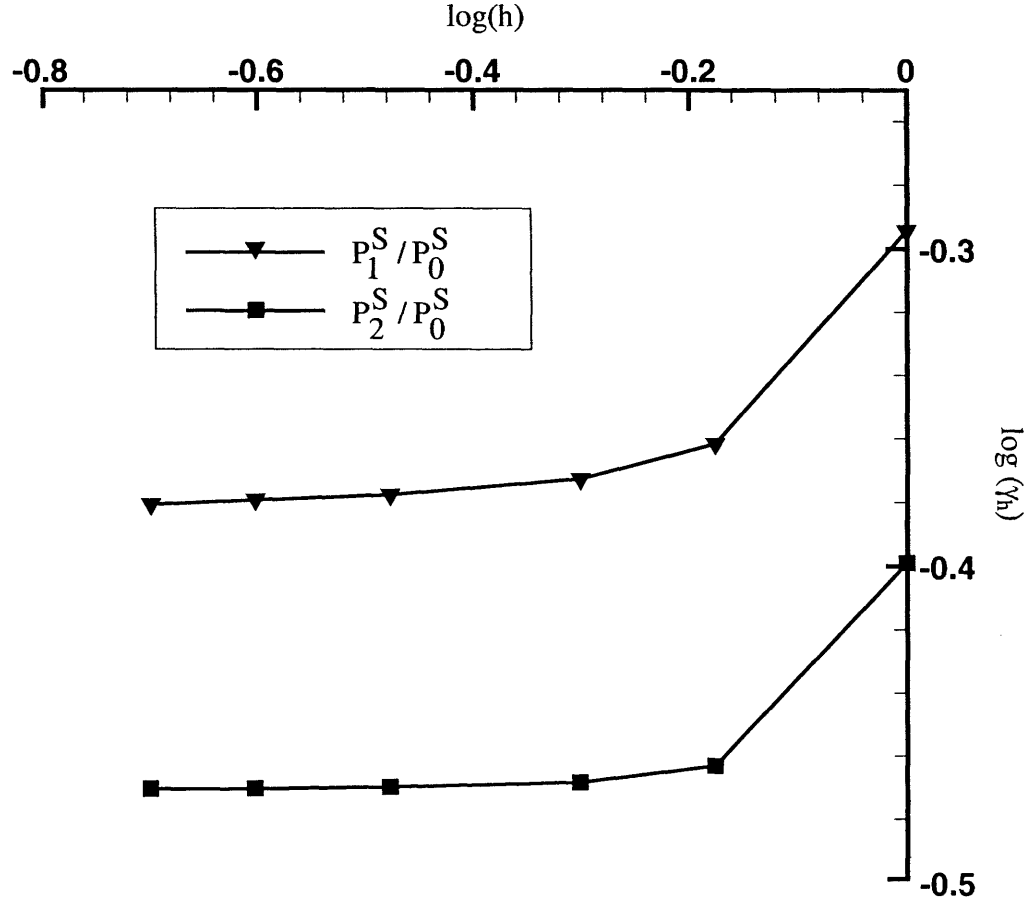


Figure 6-2: Inf-sup test results, P_1^S/P_0^S and P_2^S/P_0^S discretizations.

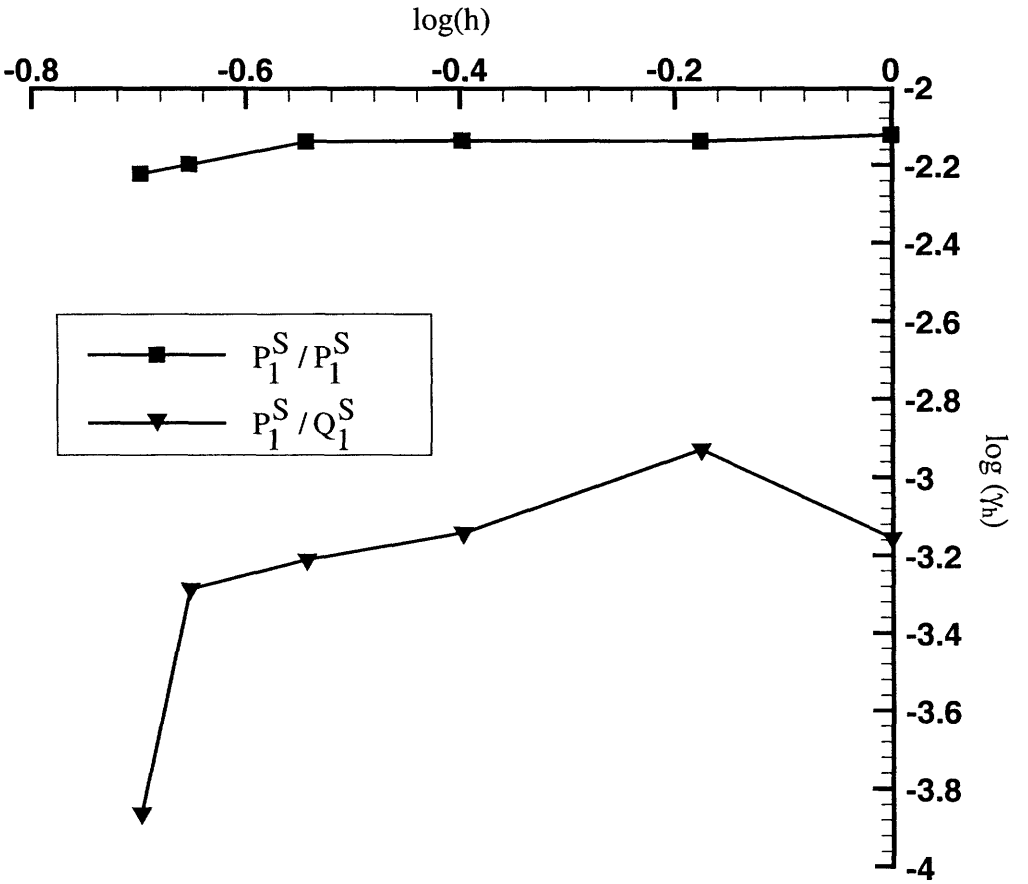


Figure 6-3: Inf-sup test results, P_1^S/P_1^S and P_1^S/Q_1^S discretizations.

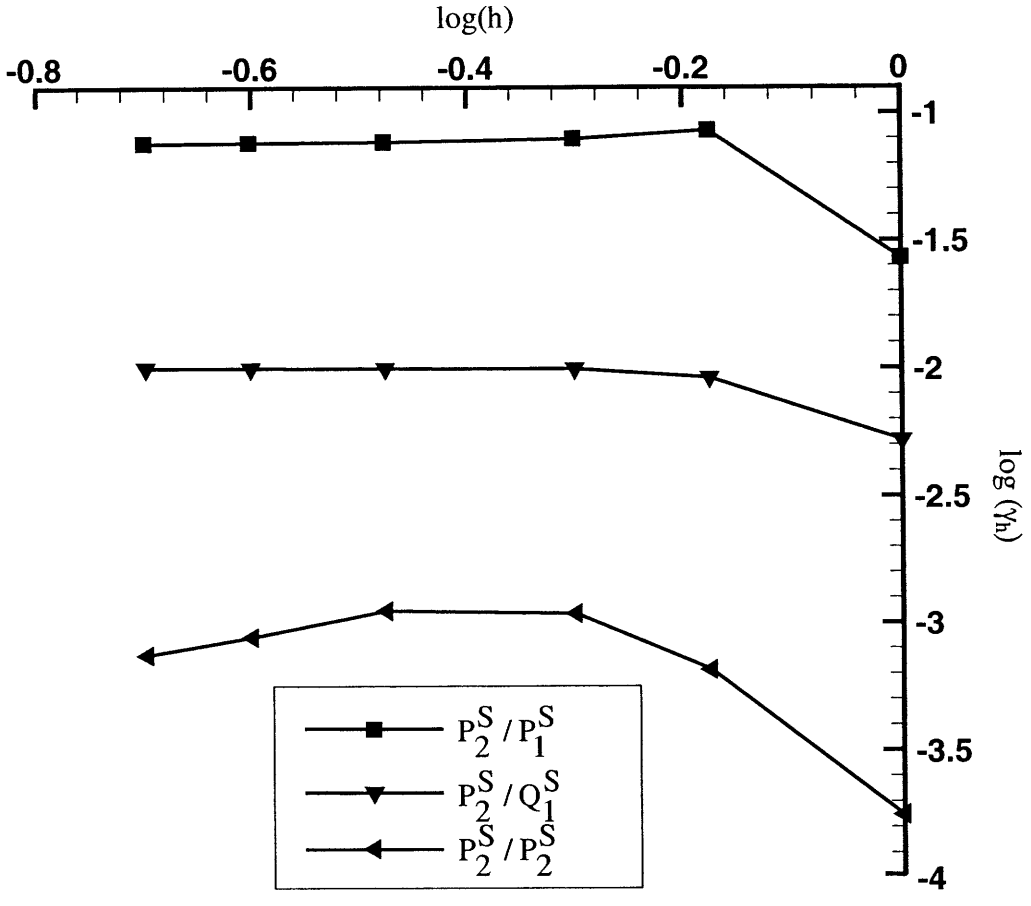


Figure 6-4: Inf-sup test results, P_2^S/P_1^S and P_2^S/Q_1^S and P_2^S/P_2^S discretizations.

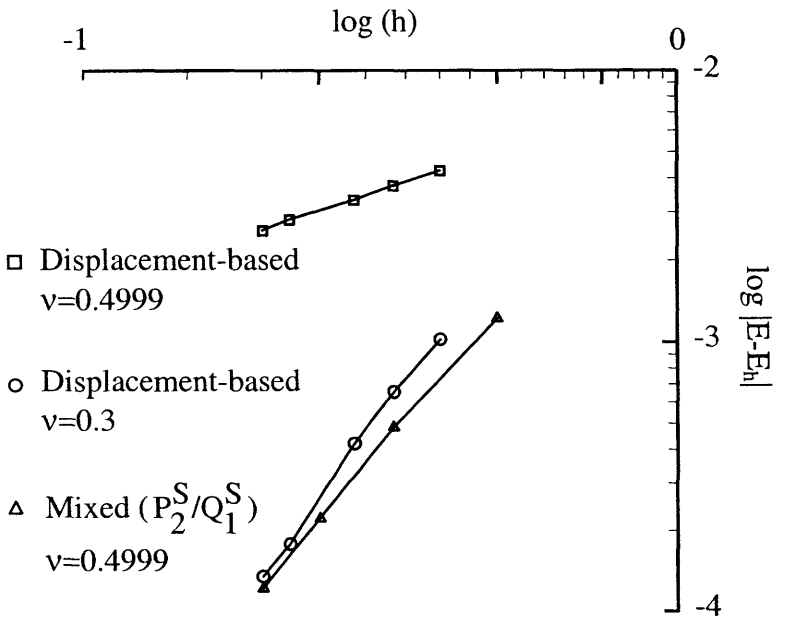
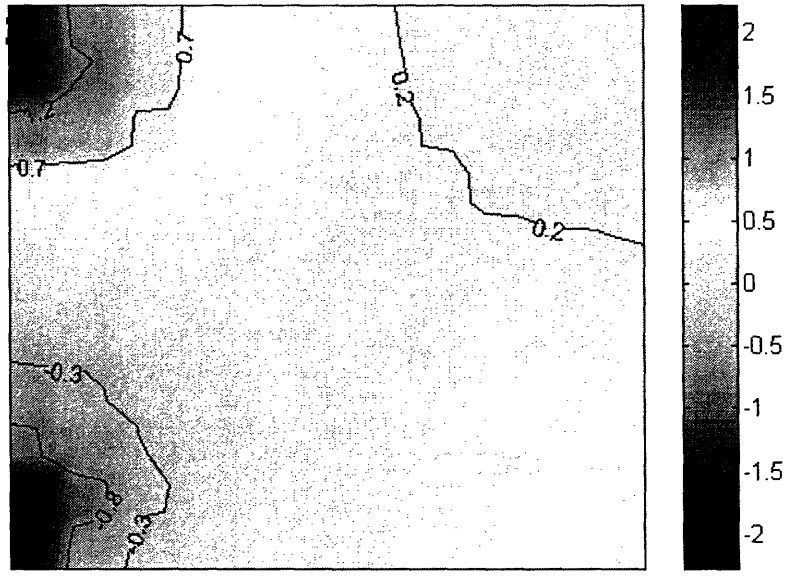
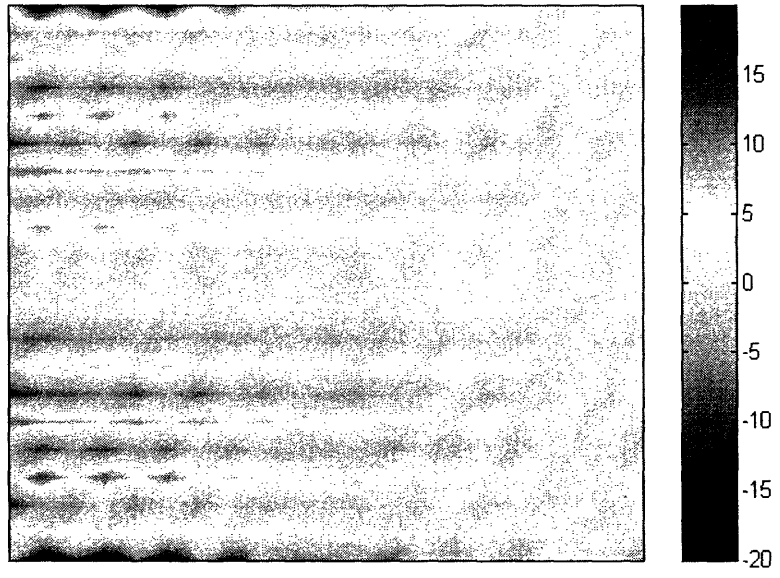


Figure 6-5: The convergence in strain energy (E_h) with decrease in radius of support (h) is shown for two different Poisson's ratios 0.3 and 0.4999. The pure displacement-based formulation is observed to lock when $\nu = 0.4999$. A mixed formulation using both pressure and displacement interpolations remedies locking (refer to the text for an explanation of the symbols used).



(a)



(b)

Figure 6-6: Computed pressure using (a) P_1^S/P_0^S and (b) P_1^S/P_1^S discretizations.

Chapter 7

Computational efficiency issues

A review of the literature on meshless techniques reveals that the current trend is towards application of the new techniques to diverse problem areas in engineering. But, it is clear that for general applications none of these methods is computationally as efficient as the traditional finite element/finite volume techniques.

The issue of efficiency has been given due consideration only recently. Breitkopf et. al., for example, have proposed a way of computing the moving least squares interpolants and their derivatives using a consistency based approach [57]. However, for any meshless technique to find eventually wide application, it must be reasonably efficient compared to the now classical finite element/finite volume techniques and it should, of course, be reliable. In this chapter we discuss this very important issue of computational efficiency (see also [58]).

Computational efficiency is achieved by the appropriate choice of interpolation functions and numerical integration rules. In chapter 2 we have presented the interpolation scheme used in the method of finite spheres. We have discussed the issue of numerical integration in chapter 4.

In section 7.1 of this chapter we critically examine some of the other popular interpolation techniques found in literature and show that the one used in the method of finite spheres is very economical. In section 7.2 we perform estimates of computational cost in the method of finite spheres and compare with similar costs in the finite element techniques as well as

another meshless technique based on the moving least squares interpolants.

7.1 Choice of interpolation scheme

The choice of the interpolation scheme is very important in determining the overall computational efficiency of a meshless computational scheme. The functions and their derivatives should be comparatively inexpensive to evaluate and they should give rise to easily integrable integrands in the Galerkin weak form.

The finite element technique uses simple polynomial functions that satisfy these conditions very efficiently. First generation wavelets, on the other hand, provide examples of functions that are mathematically elegant but are computationally very expensive. In the method of finite spheres the interpolation functions are generated using a partition of unity paradigm based on the Shepard partition of unity functions. We have discussed the interpolation scheme used in the method of finite spheres in chapter 2.

The computation of the shape function

$$h_{Im}(\mathbf{x}) = \varphi_I^0(\mathbf{x}) p_m(\mathbf{x}) \tag{7.1}$$

associated with the m^{th} degree of freedom α_{Im} of node I and its derivatives

$$\frac{\partial h_{Im}}{\partial x_i} = \varphi_I^0(\mathbf{x}) \frac{\partial p_m(\mathbf{x})}{\partial x_i} + p_m(\mathbf{x}) \frac{\partial \varphi_I^0(\mathbf{x})}{\partial x_i}, \qquad i = 1, 2 \; or \; 3 \tag{7.2}$$

is rather straightforward. The choice of the radial weighting functions for generating the Shepard functions $\varphi_I^0(\mathbf{x})$ requires important consideration. We have discussed this issue in chapter 2. In this section we briefly review some of the other popular interpolation schemes and show that the one used in the method of finite spheres offers a comparatively less expensive means of generating approximation spaces with a given order of consistency.

7.1.1 Shape functions based on least squares approximations

Of the least squares based approximation functions, the moving least squares (MLS) and weighted least squares (WLS) functions are the most popular.

Moving least squares (MLS) shape functions:

The moving least squares (MLS) method of generating smooth approximation spaces was used by Nayroles et. al. in the context of the diffuse element method [10] and was later adopted by Belytschko et. al. in their element free Galerkin method [30]. The MLS method has also been used by Atluri et. al. in their implementation of the meshless local Petrov-Galerkin method [28]. The MLS and closely related techniques require an expensive matrix inversion / equation solution process at every evaluation point and therefore result in shape functions that are computationally very expensive.

In the moving least squares method, an element v_h of the global approximation space V_h (the subscript h is a measure of the size of the spheres) can be written as

$$v_h(\mathbf{x}) = \sum_{I=1}^{N} h_I(\mathbf{x})\alpha_I \tag{7.3}$$

where $h_I(\mathbf{x})$ is the shape function associated with the nodal degree of freedom α_I. If $v_h(\mathbf{x}) \in span\{p_m(\mathbf{x})\}_{m=1}^{n}\ \forall \mathbf{x} \in \Omega$, where $p_m(\mathbf{x})$ is a polynomial or other function, then the MLS shape function corresponding to node I is expressed as

$$h_I(\mathbf{x}) = W_I(\mathbf{x})\mathbf{P}(\mathbf{x})^T\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x}_I) \tag{7.4}$$

where $W_I(\mathbf{x})$ is the radial weighting function defined in chapter 2,

$$\mathbf{P}(\mathbf{x})^T = [p_1(\mathbf{x})\ p_2(\mathbf{x}) \ldots p_n(\mathbf{x})] \tag{7.5}$$

and

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{N} W_I(\mathbf{x})\mathbf{P}(\mathbf{x}_I)\mathbf{P}(\mathbf{x}_I)^T. \tag{7.6}$$

The derivative of $h_I(\mathbf{x})$ with respect to a spatial variable x_i ($i = 1, 2\ or\ 3$) is

$$\begin{aligned} \frac{\partial h_I(\mathbf{x})}{\partial x_i} &= W_I(\mathbf{x})\frac{\partial \mathbf{P}(\mathbf{x})^T}{\partial x_i}\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x}_I) + \\ &\mathbf{P}(\mathbf{x})^T\mathbf{A}^{-1}(\mathbf{x})\left(\frac{\partial W_I(\mathbf{x})}{\partial x_i} - W_I(\mathbf{x})\frac{\partial \mathbf{A}(\mathbf{x})}{\partial x_i}\mathbf{A}^{-1}(\mathbf{x})\right)\mathbf{P}(\mathbf{x}_I). \end{aligned} \tag{7.7}$$

From equation (7.4) we observe that $supp(h_I) = supp(W_I)$. We make the following remarks

(see [30] for details)

Remark 7.1 Reproducing property: *$p_m(\mathbf{x}) = \sum_{I=1}^{N} h_I(\mathbf{x}) p_m(\mathbf{x}_I)$ for $m = 1, \ldots, n$.*

This directly follows from equations (7.4) and (7.6) since

$$\begin{aligned}
\sum_{I=1}^{N} h_I(\mathbf{x}) p_m(\mathbf{x}_I) &= \sum_{I=1}^{N} W_I(\mathbf{x}) \mathbf{P}(\mathbf{x})^T \mathbf{A}^{-1}(\mathbf{x}) \mathbf{P}(\mathbf{x}_I) p_m(\mathbf{x}_I) \\
&= \mathbf{P}(\mathbf{x})^T \mathbf{A}^{-1}(\mathbf{x}) \left(\sum_{I=1}^{N} W_I(\mathbf{x}) \mathbf{P}(\mathbf{x}_I) p_m(\mathbf{x}_I) \right) \\
&= \mathbf{P}(\mathbf{x})^T \mathbf{e}_m \qquad \text{from (7.6)} \\
&= p_m(\mathbf{x})
\end{aligned}$$

$\mathbf{e}_m$ being the unit vector in the m^{th} Cartesian direction. Hence to obtain linear consistency in R^2 for example, it is sufficient to choose $\mathbf{P}(\mathbf{x})^T = [1, x, y]$.

An interesting case arises when $\mathbf{P}(\mathbf{x}) = \{1\}$ and zeroth order consistency is assured by the shape functions in equation (7.4) which reduce to Shepard functions

$$h_I(\mathbf{x}) = \frac{W_I(\mathbf{x})}{\sum_{J=1}^{N} W_J(\mathbf{x})}. \tag{7.8}$$

Furthermore, if $1 \in \mathbf{P}(\mathbf{x})$ then the shape functions form a partition of unity subordinate to the cover $\{B(\mathbf{x}_I, r_I)\}_{I=1}^{N}$, i.e.

$$\sum_{I=1}^{N} h_I(\mathbf{x}) = 1. \tag{7.9}$$

Remark 7.2 Continuity: *Let $W_I \in C_0^s(B(\mathbf{x}_I, r_I)), I = 1, 2, \cdots, N$ and let $p_i(\mathbf{x}) \in C^l(\Omega)$, $i = 1, \ldots, m$ for $s, l \geq 0$; then the shape functions $h_I(\mathbf{x})$ in equation (7.4) satisfy $h_I(\mathbf{x}) \in C_0^{\min(s,l)}(B(\mathbf{x}_I, r_I) \cap \Omega)$.*

This property follows directly from (7.4).

Remark 7.3 Invertibility of $\mathbf{A}(\mathbf{x})$: *A necessary condition for the matrix $\mathbf{A}(\mathbf{x}) \in R^{n \times n}$ to be invertible at every $\mathbf{x} \in \Omega$ is*

$$\mathbf{x} \in \bigcap_{J \in \mathcal{Q}} B(\mathbf{x}_J, r_J) \tag{7.10}$$

where $\mathcal{Q}$ is an index set with $card\{\mathcal{Q}\} \geq n$.

This directly follows from equation (7.6), which represents $\mathbf{A}(\mathbf{x})$ as the sum of rank one dyadic products. In practice the matrix $\mathbf{A}(\mathbf{x})$ should be invertible at the integration points and therefore at every integration station at least n spheres (to be consistent in the treatment of the various meshless schemes, we assume that the support of the weighting function, W_I, is a sphere throughout our discussion) should have nonzero support. To obtain second order consistency, for example, 6 spheres in R^2 and 10 spheres in R^3 should have nonzero support at every integration point. This is quite a formidable requirement and of course it is not sufficient to ensure invertibility.

If $\mathbf{P}(\mathbf{x}) = \{1\}$ then condition (7.10) is also a *sufficient* condition for invertibility. For $d > 1$ if linear consistency is desired then an additional condition is required, namely, no three nodes $J \in \mathcal{Q}$ should be colinear. If the nodal arrangement is close to this pathological condition, then the matrix $\mathbf{A}(\mathbf{x})$ becomes ill-conditioned resulting in large approximation errors.

The fact that the $n \times n$ matrix $\mathbf{A}(\mathbf{x})$ has to be generated and then inverted at every integration point makes the MLS approach of generating shape functions very expensive.

Weighted least squares (WLS) shape functions:

The weighted least squares (WLS) interpolation scheme used in the finite point method [24] is related to the MLS technique. A multivalued global interpolation is obtained by requiring that

$$\forall \mathbf{x} \in B(\mathbf{x}_I, r_I) \qquad v_h(\mathbf{x}) = \sum_{J=1}^{N} h_J^I(\mathbf{x}) \alpha_J \tag{7.11}$$

where $h_J^I(\mathbf{x})$ and α_J are the shape function and degree of freedom, respectively, at node J such that $\mathbf{x}_J \in B(\mathbf{x}_I, r_I)$ and

$$h_J^I(\mathbf{x}) = W(\mathbf{x}_J - \mathbf{x}_I)\mathbf{P}(\mathbf{x})^T \mathbf{A}_I^{-1} \mathbf{P}(\mathbf{x}_I) \tag{7.12}$$

where

$$\mathbf{P}(\mathbf{x})^T = [p_1(\mathbf{x})\, p_2(\mathbf{x}) \cdots p_n(\mathbf{x})]$$

as before and

$$\mathbf{A}_I = \sum_{I=1}^{N} W(\mathbf{x}_J - \mathbf{x}_I)\mathbf{P}(\mathbf{x}_I)\mathbf{P}(\mathbf{x}_I)^T. \tag{7.13}$$

Note that a discontinuous global approximation is obtained. The generalized finite difference scheme of Liszka and Orkisz [2] may be considered as a special case of the WLS technique where the weights are so chosen that their supports always include the same number of nodal points.

7.1.2 Shape functions based on kernel estimates

The so called "kernel interpolation techniques" like the smoothed particle hydrodynamics (SPH) [6] and the reproducing kernel particle methods (RKPM) [13] generate nodal shape functions that are very similar to those obtained using the MLS technique. The SPH shape functions provide zeroth order consistency and hence if the functions $W_I(\mathbf{x})$ are used as the discrete kernels then the SPH shape functions are identical to the Shepard partition of unity functions. The RKPM shape functions provide higher order consistency by applying a correction to the SPH kernel. Of course, if the same function $W_I(\mathbf{x})$ is used as the kernel function and the consistency requirement of order p is imposed, the resulting RKPM shape functions are identical to the MLS shape functions in equation (7.4) satisfying p^{th} order consistency [30].

7.1.3 hp-clouds shape functions

The general hp-clouds functions of Duarte and Oden [20] use moving least squares functions $\varphi_I^p(\mathbf{x})$ satisfying p^{th} order consistency to provide the partition of unity, i.e.

$$V_h = \sum_{I=1}^{N} \varphi_I^p V_I^h \tag{7.14}$$

where

$$\varphi_I^p(\mathbf{x}) = W_I(\mathbf{x})\mathbf{P}(\mathbf{x})^T\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x}_I) \tag{7.15}$$

with $\mathbf{P}(\mathbf{x})$ containing complete polynomials of order p and $\mathbf{A}(\mathbf{x})$ defined in equation (7.6). It should be recognized that this procedure of generating the partition of unity functions is extremely expensive and we do not consider these functions in our discussion in section 7.2.

However, for $p = 0$ the hp-clouds functions are identical to the functions used in the method of finite spheres. It was also pointed out by Duarte and Oden that the use of Shepard functions to generate the partitions of unity is probably the least expensive for a given level of accuracy.

7.2 Computational costs

In this section we estimate the computational cost in the displacement-based method of finite spheres and compare this cost with the expense of a similar meshless scheme using the moving least squares approximants as well as the classical finite element techniques. We adopt the $\mathcal{O}$-notation to imply the asymptotic upper bound to within a constant, i.e., for a given function $g(n)$, we denote by $\mathcal{O}(g(n))$ the set of functions

$$\mathcal{O}(g(n)) = \{f(n) : \exists \text{ constants } c, n_o > 0 \text{ such that } 0 \leq f(n) \leq cg(n) \forall n \geq n_0\}.$$

For example, if $f(n) = a_o n + a_1$ with $a_o > 0$, then we may write $f(n) \in \mathcal{O}(n)$. A constant is represented as $\mathcal{O}(1)$.

We assume that the major computational cost may be broken down into:

1. cost of computation of the global stiffness matrix, and
2. cost of solution of the resulting set of algebraic equations.

This approach neglects the computation of the loads, the application of the boundary conditions, memory traffic and other overheads associated with the execution of a general numerical software and therefore gives only a rough estimate of the efficiency. Moreover, we do not consider the preprocessing or postprocessing time. However, a main advantage of the meshless techniques over the traditional finite element techniques is that preprocessing time is reduced as no mesh is required. Postprocessing in meshless techniques is also relatively straightforward since, for example, no additional stress smoothening is required.

We consider a general elliptic problem in d dimensions ($d = 1, 2$ or 3) and assume a discretization scheme using N nodal points and a consistency of order p. The superscripts

MLS, FEM and MFS will be used in the following discussion to differentiate the same variable for different methods. We assume banded symmetric matrices with constant column height and a constant half bandwidth m_K. Each node is assumed to have an average connectivity of M, i.e. the support of each node is assumed to have nonzero overlaps with an average of $(M-1)$ other nodal supports.

7.2.1 Cost of computation of the global stiffness matrix

The finite element method:

In the finite element method, we assume that the stiffness matrix has 'dN^{FEM}' rows and 'dM^{FEM}' non-zero columns per row. Hence, the computational time for the global stiffness matrix may be assessed as

$$T_K^{FEM} = \mathcal{O}(d^2 N^{FEM} M^{FEM} T_{K_{ij}}^{FEM}) \tag{7.16}$$

where $T_{K_{ij}}^{FEM}$ is the computational time for a single term of the stiffness matrix, which is assumed to be a volume integral of the sum of inner products of the derivatives of the shape functions. This integral is evaluated using numerical integration over each finite element. Let N_g^{FEM} denote the number of Gaussian integration points per finite element ($N_g^{FEM} = (p+1)^2$ for a tensor product element in R^2 using a complete polynomial of order p). We model $T_{K_{ij}}^{FEM}$ as

$$T_{K_{ij}}^{FEM} = \mathcal{O}(N_g^{FEM} T_h^{FEM}) \tag{7.17}$$

where T_h^{FEM} is the average computational time for a finite element shape function (or its derivative). We neglect the fact that multiple derivatives need be taken and different elements connected to a nodal point contribute. Therefore

$$T_K^{FEM} = \mathcal{O}(d^2 N_g^{FEM} N^{FEM} M^{FEM} T_h^{FEM}). \tag{7.18}$$

Meshless method using moving least squares interpolants:

We consider a meshless method which uses a moving least squares type of interpolant. The

stiffness matrix has 'dN^{MLS}' rows and 'dM^{MLS}' non-zero columns per row. Hence, the computational time for the global stiffness matrix may be modeled as

$$T_K^{MLS} = \mathcal{O}(d^2 N^{MLS} M^{MLS} T_{K_{ij}}^{MLS}) \tag{7.19}$$

where $T_{K_{ij}}^{MLS}$ is the computational time for a single term of the stiffness matrix. This integral is evaluated using numerical integration. Let N_g^{MLS} denote the number of integration points per sphere. We model $T_{K_{ij}}^{MLS}$ as

$$T_{K_{ij}}^{MLS} = \mathcal{O}(N_g^{MLS} T_h^{MLS}) \tag{7.20}$$

where T_h^{MLS} is the average computational time for a MLS shape function (or its derivative). The MLS shape functions and their derivatives are very complex (see equations (7.4) and (7.7)) and involve the inversion of a $n \times n$ matrix at each integration point, where $n = (p+1)(p+2)/2$, if a consistency of order p is required in R^2. We may model T_h^{MLS} as

$$T_h^{MLS} = \mathcal{O}(n^2 M^{MLS} T_W) \tag{7.21}$$

where T_W is the computational cost of evaluating the weighting function or its derivative at a single evaluation point. This is justified since matrix inversion and matrix-matrix multiplies are $\mathcal{O}(n^3)$ operations and $M^{MLS} = \mathcal{O}(n)$ at least to ensure invertibility of the $\mathbf{A}(x)$ matrix in equation (7.6). Therefore

$$T_K^{MLS} = \mathcal{O}(d^2 n^2 N_g^{MLS} N^{MLS} (M^{MLS})^2 T_W). \tag{7.22}$$

Method of finite spheres:

In the method of finite spheres with n functions in the local basis enforcing p^{th} order consistency, the global stiffness matrix has 'dnN^{MFS}' rows and 'dnM^{MFS}' non-zero columns per row. Hence, the computational time for the global stiffness matrix may be modeled as

$$T_K^{MFS} = \mathcal{O}(d^2 n^2 N^{MFS} M^{MFS} T_{K_{ij}}^{MFS}) \tag{7.23}$$

where $T_{K_{ij}}^{MFS}$ is the computational time for a single term of the stiffness matrix. Let N_g^{MFS}

denote the number of integration points per sphere. We model $T_{K_{ij}}^{MFS}$ as

$$T_{K_{ij}}^{MFS} = \mathcal{O}(N_g^{MFS} T_h^{MFS}) \tag{7.24}$$

where T_h^{MFS} is the average computational time for a MFS shape function (or its derivative). The MFS shape functions and their derivatives are much simpler to compute than the MLS shape functions (see equations (7.1) and (7.2)) and do not involve matrix inversions. The most expensive operation is the computation of the Shepard partition of unity functions (and their derivatives) which requires the evaluation of M^{MFS} weighting functions at each integration point. Therefore we may model T_h^{MFS} as

$$T_h^{MFS} = \mathcal{O}(M^{MFS} T_W) \tag{7.25}$$

where T_W is the computational cost of evaluating the weighting function or its derivative at a single evaluation point. Therefore

$$T_K^{MFS} = \mathcal{O}(d^2 n^2 N_g^{MFS} N^{MFS} (M^{MFS})^2 T_W). \tag{7.26}$$

Comparisons:

It is interesting to observe that estimates (7.22) and (7.26) have the same form. It may appear that the cost advantage in computing the simpler stiffness terms is lost in the number of terms that have to be computed in the MFS. This is true if the same number of nodes, connectivity and, of course number of integration points, are used in both techniques. But due to the condition mentioned in remark 7.3 regarding the invertibility of the $\mathbf{A}(x)$ matrix, $M^{MLS} = \mathcal{O}(n)$, while essentially $M^{MFS} = \mathcal{O}(1)$. Indeed, it has been reported that M^{MLS} for the MLS functions used in the element free Galerkin method can be as high as 50 in R^2 [?]. In the method of finite spheres, however, M^{MFS} can be 4. Furthermore, the invertibility of the $\mathbf{A}(x)$ matrix also necessitates that $N^{MLS} \gg N^{MFS}$ for comparable accuracy (for example, in the method of finite spheres, the problem in figure 5(a) may be solved with quadratic consistency using only four nodes at the four corners). Therefore

$$\frac{T_K^{MLS}}{T_K^{MFS}} \gg 1.0. \tag{7.27}$$

Next, we compare the computation time estimates (7.18) and (7.26)

$$\frac{T_K^{MFS}}{T_K^{FEM}} = O\left(\frac{n^2 N_g^{MFS} N^{MFS} (M^{MFS})^2 T_W}{N_g^{FEM} N^{FEM} M^{FEM} T_h^{FEM}}\right).$$

It is true that $T_K^{MFS}/T_K^{FEM} > 1.0$ but the ratio is not very large since $N^{MFS} \ll N^{FEM}$ for the same accuracy in solution. This is because in the method of finite spheres, the observed convergence rate is much better than in the finite element method [34].

Let us consider the example of the square cantilevered plate in plane strain shown in figure 5-4 with uniformly distributed loading on the top surface. The convergence in strain energy corresponding to a uniform h-type refinement is presented in figures 5-5 (a) and (b) when the problem was solved using 9 noded finite elements and the method of finite spheres respectively. It is observed that a 8×8 regular nodal arrangement (with quadratic consistency) provides a solution (in strain energy) which is comparable in accuracy with the solution provided by a 25×25 mesh of nine noded finite elements. Hence, for comparable accuracy and uniform meshes, we require 5202 degrees of freedom in the finite element technique and 768 (64 nodes × 12 degrees of freedom per node) degrees of freedom in the method of finite spheres. We estimate

$$\begin{aligned} &N^{FEM} = 2601;\ N^{MFS} = 64; \\ &N_g^{FEM} = 9;\ N_g^{MFS} = 144; \\ &M^{FEM} \sim 25;\ M^{MFS} \sim 4; \\ &n = 6 \end{aligned} \tag{7.28}$$

and obtain

$$\frac{T_K^{MFS}}{T_K^{FEM}} \sim 9 \tag{7.29}$$

assuming that T_h^{FEM} and T_W are of the same order of magnitude. This estimate is quite close to the one obtained by comparing the actual computational times (the observed ratio is around 8).

7.2.2 Cost of solution including solving the set of algebraic equations

Given the dimensions of the stiffness matrices considered in the previous section, the solution times for the three techniques may be assessed as

The finite element method:

$$T_s^{FEM} = \mathcal{O}(dN^{FEM}\left(m_K^{FEM}\right)^2) \tag{7.30}$$

Meshless method using moving least squares interpolants:

$$T_s^{MLS} = \mathcal{O}(dN^{MLS}\left(m_K^{MLS}\right)^2) \tag{7.31}$$

Method of finite spheres:

$$T_s^{MFS} = \mathcal{O}(dnN^{MFS}\left(m_K^{MFS}\right)^2). \tag{7.32}$$

We are interested in the order of magnitude estimates and not any particular solution technique.

From our discussion above we realize that for a large problem, the half bandwidth $m_K^{MFS} \ll m_K^{MLS}$ and $N^{MFS} \ll N^{MLS}$. Therefore it is reasonable to estimate that $T_s^{MFS} \ll T_s^{MLS}$.

Comparing estimates (7.30) and (7.32) we observe that $N^{MFS} \ll N^{FEM}$ for comparable accuracy and $m_K^{MFS} \ll m_K^{FEM}$. For example, from the data in (7.28) we see that

$$N^{MFS} = 64;\ N^{FEM} = 2601;$$
$$m_K^{FEM} = 209;\ m_K^{MFS} = 119;$$

and therefore

$$\frac{T_s^{MFS}}{T_s^{FEM}} \sim 4.8 \times 10^{-2}. \tag{7.33}$$

Estimates (7.29) and (7.33) are important in comparing the total computational costs of

the method of finite spheres (T_{total}^{MFS}) and the finite element technique (T_{total}^{FEM}). We assume

$$\begin{aligned} T_{total}^{FEM} &= T_K^{FEM} + T_s^{FEM} \\ T_{total}^{MFS} &= T_K^{MFS} + T_s^{MFS}. \end{aligned} \tag{7.34}$$

We observe, by solving the problem in figure 5-4 using a sequence of 9-noded finite elements using the commercial finite element software package ADINA, that for fine discretizations

$$\frac{T_K^{FEM}}{T_s^{FEM}} \sim 1.0. \tag{7.35}$$

From (7.34) we have

$$\frac{T_{total}^{MFS}}{T_{total}^{FEM}} = \frac{\left(T_K^{MFS}/T_K^{FEM}\right)}{1 + T_s^{FEM}/T_K^{FEM}} + \frac{\left(T_s^{MFS}/T_s^{FEM}\right)}{1 + T_K^{FEM}/T_s^{FEM}}. \tag{7.36}$$

From (7.33) it may be observed that the ratio T_s^{MFS}/T_s^{FEM} is very small compared to the denominator (which is of the order of 2). From (7.29) we observe that the ratio T_K^{MFS}/T_K^{FEM} is of the order of 10. Therefore, it is reasonable to estimate that the method of finite spheres is about five (or say ten) times slower than the finite element technique for elastostatic problems in two dimensions.

In summary, it is interesting to note that even though multiple shape functions are used at a node the method of finite spheres is more efficient than the techniques based on the moving least squares interpolants since no matrix inversion is required at every integration point and there are no stringent overlap criteria. In the current form of implementation, when measured roughly theoretically and as seen in an example solution, the method of finite spheres is about five times slower than the traditional finite element techniques for representative problems in two-dimensions since it offers comparable accuracy in solution with considerably fewer nodes on the domain. This is quite encouraging since the preprocessing time is considerably less than in the finite element techniques.

Chapter 8

A specialized application

In this chapter we consider a very specialized application of the method of finite spheres to multi modal medical simulations. The problem is to simulate the mechanical behavior of soft tissue in *real time* using a physically-based model. As we shall see, this application calls for very high computation speeds and relatively low accuracy. To be able to achieve such speeds drastic simplifications and approximations need to be introduced in our technique. Therefore, in this chapter we deviate considerably from the earlier ones and introduce a version of the method of finite spheres using moving least squares interpolants and point collocation (in this form the technique is very similar to the finite point method [24]). Since the application area is relatively new and uncommon, we give more details regarding the background and challenges in the field than in the previous chapters.

In section 8.1 we present the background of the problem and current state-of-the-art. In section 8.2 we discuss the specialized version of the method of finite spheres. We realize that several important computational and implementational issues need to be addressed. In section 8.3 we discuss these issues. In section 8.4 we present several examples demonstrating the applicability of the proposed technique to real time surgical simulation problems. Finally in section 8.5 we show that the point collocation based method of finite spheres is substantially faster than the traditional finite element techniques (since localized discretization is used) and is ideally suited for real time medical simulations.

Figure 8-1: A typical laparoscopic operation.

8.1 Background

Owing to recent advances in computational speeds and interface devices, there has been considerable excitement about the possibility of having medical simulators to train physicians much like the use of flight simulators to train pilots. In addition to the computer such systems have one or more human machine interface(s). In order to be realistic, they should provide *multi sensory* interaction capabilities and be real time. In this chapter we present real time medical simulation techniques that can be used to generate synthetic environments where the user interacts with the objects in the environment using visual and tactile sensory modalities.

An example of medical procedures where both visual and tactile information are important is minimally invasive surgery. Laparoscopic surgery is a particular example of such a procedure. This technology uses a small video camera and a few long slender instruments to perform surgery with minimum incision. The camera and instruments are introduced into the abdomen or chest through small skin incisions that enable the surgeon to explore and operate on the internal cavity without the need for making large openings (see figure 8-1).

The laparoscopic surgeon faces three major types of problems. First, the visualization of the

internal organ is achieved by means of a wide angled camera, but the vision is monoscopic and limited. Second, hand-eye coordination is a problem since the TV monitor presents the mirror images of the actual hand movements and anatomical landmarks. Third, the haptic cues are substantially reduced since he/she interacts with the internal organs with long slender surgical instruments.

This calls for specialized training approaches for the surgical resident to be better prepared for performing such procedures. The success of flight simulators in training pilots has inspired us to develop analogous "surgical simulators" as immersive virtual environment systems that will train medical personnel with virtual patients. This will not only reduce the use of animals and cadavers, that are currently used for such trainings, but also result in customized practice environments for medical students.

An important issue in medical simulation is the modeling of soft tissues. From a purely mechanistic viewpoint, soft tissues exhibit complex material properties [59]. They are nonlinear, anisotropic, viscoelastic and nonhomogeneous (usually layered). Human skin, for example, has an outer layer called "epidermis" and an inner layer called "dermis" with different sublayers having different material properties.

Moreover, soft tissues deform considerably under the application of relatively small loads. For example, strains of the order of 100% are not at all uncommon. The behavior of soft tissues is usually governed by time dependent nonlinear coupled partial differential equations with complex boundary and initial conditions. In addition it is very difficult to obtain *in-vivo* material properties of living tissues. The *in vivo* material properties differ substantially from those obtained from *in vitro* samples. It is therefore a challenging task to develop efficient models for living tissues so that the simulation of tool-tissue interaction may be performed in real time.

For real time visual display an update rate of 30 Hz is sufficient. For haptic display, we use the Phantom [1] haptic interface device. For stable simulation, the haptic loop requires an update rate of about 1kHz [60, 61]. This imposes severe restriction on the complexity

[1] Developed by SensAble Technologies, Inc.

of the models that can be rendered haptically. Therefore simulation speed is the prime consideration. Reasonably accurate simulations at the required speeds is therefore the goal.

Various techniques can be found in literature for the simulation and display of deformable objects. These techniques can be categorized into two main approaches: "geometrically based" approaches and "physically based" approaches [62, 63]. The "geometrically based" modeling approaches, such as Bezier/B-spline based procedures and free form deformation techniques, do not account for the physics of deformation, but are simpler to implement. In contrast, the "physically based" approaches, such as the lumped parameter particle models and finite element based techniques, attempt to model the underlying physics, but are computationally intensive.

A "geometrically based" approach for the modeling of deformable objects is the free form deformation technique [64]. This approach enables the user to modify the shape of the object interactively by manipulating the lattice of control points. Hsu et. al. improved the technique so that the user is able to manipulate the object surfaces directly without using control points [65]. Basdogan et. al. developed a different technique for local deformation of surfaces [66]. In this technique the translation of each vertex point within a certain distance (called the radius of influence) from the collision point is determined using a simple polynomial function. The computation of interaction forces between the laparoscopic tool and the tissue is based on the penetration depth of the tool tip. These approaches, however, do not account for the physics of deformation. An example of a situation where any of these "geometrically based" approaches would not be effective is the simulation of a layer of soft tissue covering a bone.

One of the most popular "physically based" modeling approaches is a lumped parameter technique using masses, springs and dampers. Each node has a point mass associated with it and is connected to its neighbors using linear or nonlinear springs and dampers. Cover et. al. developed the first laparoscopic gall bladder surgery simulator using surface-based spring-mass models [67]. The spring-mass models are also widely used in facial simulations. For example, Terzopoulos and Waters constructed a three-layer network of springs based on three anatomically distinct layers of facial tissue [68].

The spring-mass models are simple and computationally very efficient. However, they suffer from certain limitations. First, the construction of a network of springs in 3D is a complicated process. There is no theoretical basis of how many springs are to be attached to each node or which pairs of nodes are to be connected together to simulate the elastic properties of soft tissues. Secondly, under certain conditions, mass-spring systems may become oscillatory or even go unstable during simulation [69] (unless dampers are added to the system). Finally, the convergence behavior of these models, i.e. the rate at which the modeling error decreases with increase in the number of elements, is not well established.

The finite element technique is widely used in engineering analysis for the simulation of deformable objects [1]. Pieper et. al. developed a planning system for facial plastic surgery using isoparametric finite elements [70]. Bro-Nielsen et. al. applied finite elements for real time surgical simulations using tetrahedral volume elements [71]. The computation time was reduced significantly by using a matrix condensation technique. Cotin et. al. demonstrated a hepatic surgery simulator using similar finite element models [72, 73].

Bro-Nielsen reported that the computation time could be reduced significantly by using a semi-implicit scheme and inverting the condensed stiffness matrix in the precomputation stage [74]. A model of the lower leg with 700 nodes could be deformed with a visual update of 20 Hz. No force feedback was however computed. This technique, however, works only under the condition that the stiffness matrix is diagonal or near diagonal. Moreover, the precomputed data is not reusable when topological changes occur to the object such as during cutting or tearing. Berkely et al. developed a real time finite element model with force feedback for skin surgery [75]. This technique also uses pre-computations extensively.

Although the finite element methods are truly physically based modeling techniques and simulate complex deformations quite accurately, they have certain drawbacks when applied to real time simulations. First, the contact between tool and tissues must occur at nodal points (see figure 8-2 (a) for a situation when this does not happen). Therefore, to prevent loss of resolution, the density of nodal points should be sufficiently high. This requires extensive memory resources and high computational overhead. Second, cutting or tear-

ing requires an expensive remeshing process during simulation. This means precomputed data of the object becomes, at least locally, useless and all data must be computed in real time. The computation time increases approximately as the cubic of the number of nodal unknowns. This poses significant obstacles in real time applications, given the very high frequency of force update required.

A solution to the problems that are faced by the finite element techniques is provided by meshless methods (see figure 8-2 (b)). In this chapter we develop a specialized version of the method of finite spheres for the purpose of real time medical simulations. Nodal points are sprinkled around the surgical tool tip (not the entire domain) and the interpolation is performed by moving least squares functions compactly supported on spheres surrounding the nodes. A point collocation technique is used to generate the discrete equations that are solved in real time.

The localization provided by the finite influence zones of the nodes as well as the elimination of numerical integration results in a highly accelerated numerical scheme. The flexibility in the placement of nodes allows complex operations like cutting to take place relatively easily. In addition, since the differential equations governing the tool tissue interactions are being solved in the vicinity of the tool tip, the solution procedure is physically based.

8.2 Point collocation based method of finite spheres

In this section we assume that soft tissue behavior may be modeled using a linear elastic material assumption. Time dependence of material response (viscoelasticity) may be neglected for tool-tissue interaction times larger than the largest time constant of the system. In the next section we shall see how we may introduce nonlinear tissue behavior approximately in the framework presented in this section.

8.2.1 Governing equations

Let us consider the linear elasticity problem defined on a continuum $\Omega \in R^3$ with boundary Γ (see figure 8-3). The system of governing differential equations is

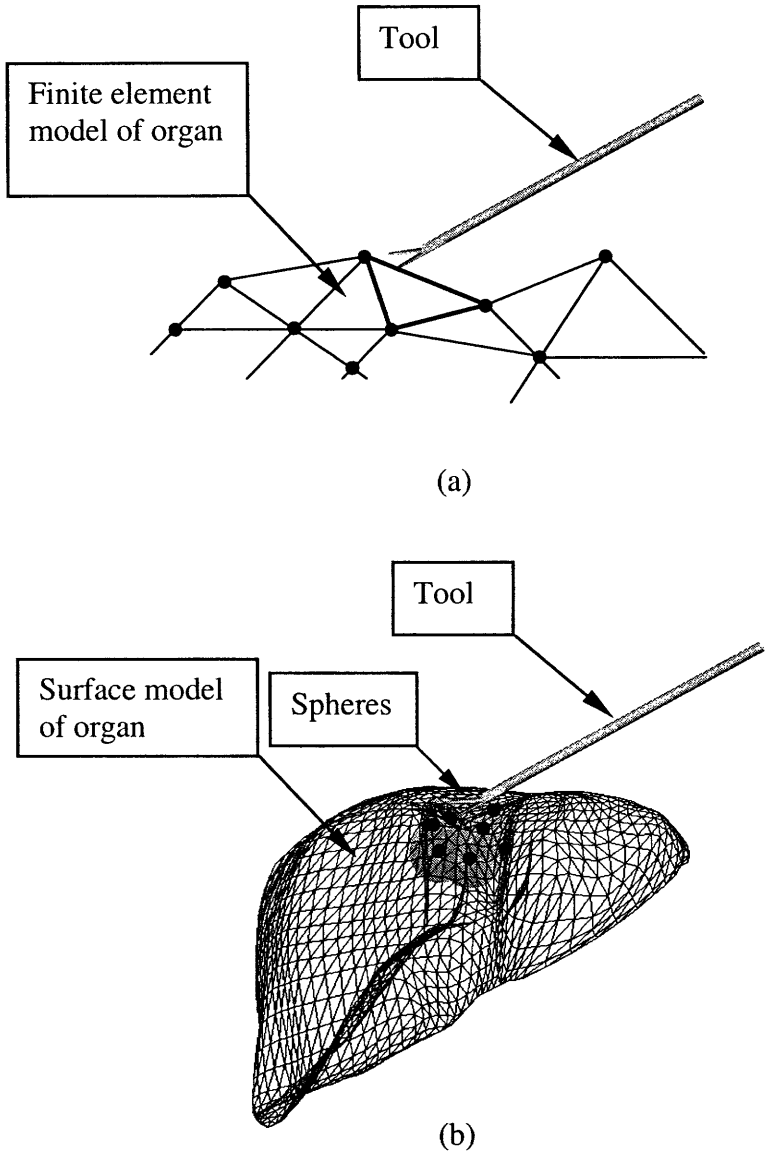


Figure 8-2: In the finite element technique the entire organ has to be meshed. Problem arises when the tool tip is between two nodes (a). In (b) the surface model of a liver is shown. Local discretization is achieved in the vicinity of the tool tip by using the method of finite spheres. Nodes are sprinkled around the tool tip and point collocation is used. The nodal arrangement can travel with the tool tip.

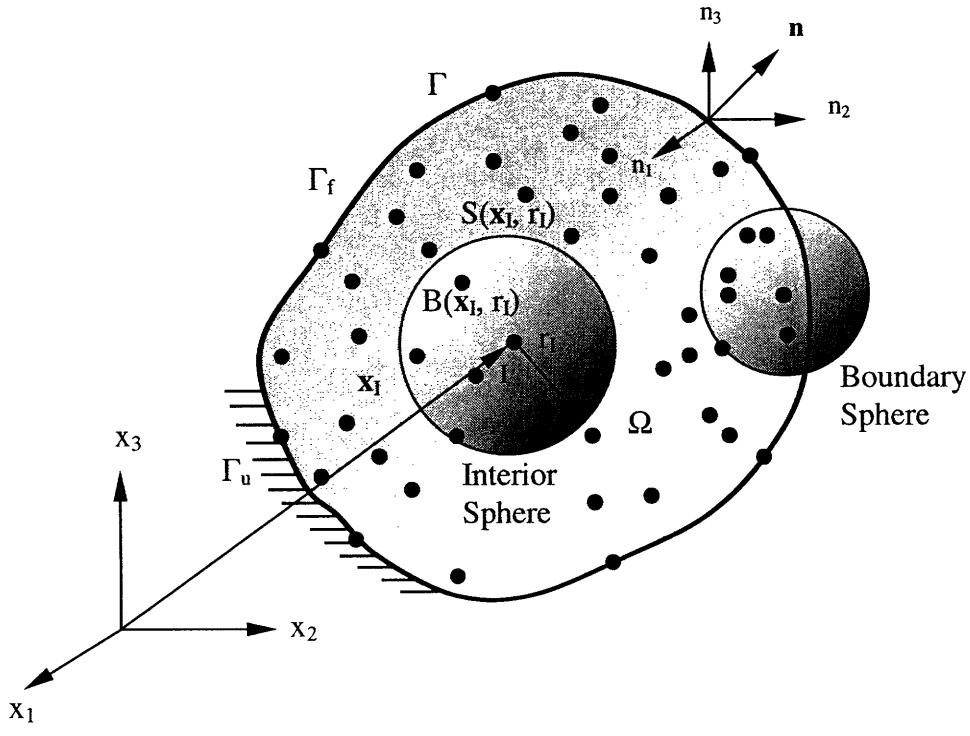


Figure 8-3: General three-dimensional body, Ω, with boundary Γ, discretized using a set of nodes. Γ_u is the portion of the boundary on which Dirichlet (displacement) boundary conditions are specified whereas Γ_f is the portion of the boundary on which Neumann (force) boundary conditions are specified. $\Gamma = \Gamma_f \cup \Gamma_u$ and $\Gamma_f \cap \Gamma_u = 0$. $\mathbf{n}$ is the outward unit normal to the boundary.

$$\boldsymbol{\partial}_\epsilon^T \boldsymbol{\tau}(\mathbf{u}) + \mathbf{f}^B(\mathbf{x}) = \mathbf{0} \qquad \text{in } \Omega \tag{8.1}$$

with force (Neumann) boundary conditions

$$\mathbf{N}\boldsymbol{\tau}(\mathbf{u}) = \mathbf{f}^S(\mathbf{x}) \qquad \text{on } \Gamma_f \tag{8.2}$$

and displacement (Dirichlet) boundary conditions

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}^S(\mathbf{x}) \qquad \text{on } \Gamma_u \tag{8.3}$$

In equations (8.1) to (8.3), $\mathbf{u}$ and $\boldsymbol{\tau}$ are the displacement and stress vectors, $\mathbf{f}^S$ is the prescribed traction vector on the Neumann boundary Γ_f, $\mathbf{u}^S$ is the vector of prescribed displacements on the Dirichlet boundary Γ_u (note that the domain boundary $\Gamma = \Gamma_f \cup \Gamma_u$ and $\Gamma_f \cap \Gamma_u = 0$), $\mathbf{f}^B$ is the body force vector (including inertia terms), $\boldsymbol{\partial}_\epsilon$ is a linear gradient operator, $\mathbf{N}$ is the matrix of direction cosine components of a unit normal to the domain boundary (positive outwards) and $\mathbf{x} = \{x,y,z\}$. In R^3 these vectors and matrices are

$$\mathbf{u} = [u(\mathbf{x}) \; v(\mathbf{x}) \; w(\mathbf{x})]^T \tag{8.4}$$

$$\boldsymbol{\tau} = [\tau_{xx} \; \tau_{yy} \; \tau_{zz} \; \tau_{xy} \; \tau_{yz} \; \tau_{zx}]^T \tag{8.5}$$

$$\mathbf{f}^S = [f_x^S(\mathbf{x}) \; f_y^S(\mathbf{x}) \; f_z^S(\mathbf{x})]^T \tag{8.6}$$

$$\mathbf{u}^S = [u^S(\mathbf{x}) \; v^S(\mathbf{x}) \; w^S(\mathbf{x})]^T \tag{8.7}$$

$$\boldsymbol{\partial}_\epsilon = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix} \tag{8.8}$$

$$\mathbf{N} = \begin{bmatrix} n_x & 0 & 0 & n_y & 0 & n_z \\ 0 & n_y & 0 & n_x & n_z & 0 \\ 0 & 0 & n_z & 0 & n_y & n_x \end{bmatrix} \tag{8.9}$$

The stress-strain constitutive equation may be written as

$$\boldsymbol{\tau}(\mathbf{u}) = \mathbf{C}\boldsymbol{\epsilon}(\mathbf{u}) \tag{8.10}$$

where $\boldsymbol{\epsilon}(\mathbf{u})$ is the vector of strains corresponding to the stresses in (8.5) and

$$\mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{11} & c_{12} & 0 & 0 & 0 \\ c_{12} & c_{12} & c_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{22} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{22} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{22} \end{bmatrix} \tag{8.11}$$

where

$$c_{11} = \frac{E(1-\nu)}{(1+\nu)(1-2\nu)},\ c_{12} = \frac{E\nu}{(1+\nu)(1-2\nu)} \text{ and } c_{22} = \frac{E}{2(1+\nu)}$$

E and ν being the Young's modulus and Poisson's ratio of the material, respectively.

8.2.2 Point collocation

The point collocation technique [1] is a weighted residual scheme in which the displacement solution $\mathbf{u}$ is approximated by $\mathbf{u}_h$ and the governing partial differential equations are evaluated at the nodal points. The discrete set of equations may be written as

$$\left[\boldsymbol{\partial}_\epsilon^T \boldsymbol{\tau}(\mathbf{u}_h)\right]_{\mathbf{x}=\mathbf{x}_I} + \mathbf{f}^B(\mathbf{x}_I) = \mathbf{0} \quad \text{in } \Omega \tag{8.12}$$

with force (Neumann) boundary conditions

$$[\mathbf{N}\boldsymbol{\tau}(\mathbf{u}_h)]_{\mathbf{x}=\mathbf{x}_I} = \mathbf{f}^S(\mathbf{x}_I) \quad \text{on } \Gamma_f \tag{8.13}$$

and displacement (Dirichlet) boundary conditions

$$\mathbf{u}_h(\mathbf{x}_I) = \mathbf{u}^S \qquad \text{on } \Gamma_u \tag{8.14}$$

where $\mathbf{x}_I$ is the position vector of node I.

8.2.3 Nodal interpolation

We choose the moving least squares functions (described in chapter 7) as the trial functions, i.e.

$$\mathbf{u}_h(\mathbf{x}) = \sum_{J=1}^{N} \mathbf{H}_J(\mathbf{x})\boldsymbol{\alpha}_J = \mathbf{H}(\mathbf{x})\mathbf{U}, \tag{8.15}$$

where N nodes are used for discretization and

$$\mathbf{U} = [\boldsymbol{\alpha}_1\ \boldsymbol{\alpha}_2\ \boldsymbol{\alpha}_3\ \ldots]^T$$

is the vector of nodal unknowns (not nodal displacements unless the Kronecker delta property is satisfied by the shape functions), and $\boldsymbol{\alpha}_J = [u^J \quad v^J \quad w^J]$ is the vector of nodal unknowns at node J (u^J, v^J and w^J are the nodal variables for the x, y and z direction displacements at node J). The nodal shape function matrix corresponding to the J^{th} node is

$$\mathbf{H}_J(\mathbf{x}) = \begin{bmatrix} h_J(\mathbf{x}) & 0 & 0 \\ 0 & h_J(\mathbf{x}) & 0 \\ 0 & 0 & h_J(\mathbf{x}) \end{bmatrix}. \tag{8.16}$$

where

$$h_J(\mathbf{x}) = W_J(\mathbf{x})\mathbf{P}(\mathbf{x})^T\mathbf{A}^{-1}(\mathbf{x})\mathbf{P}(\mathbf{x}_J) \tag{8.17}$$

with

$$\mathbf{A}(\mathbf{x}) = \sum_{I=1}^{N} W_I(\mathbf{x})\mathbf{P}(\mathbf{x}_I)\mathbf{P}(\mathbf{x}_I)^T. \tag{8.18}$$

The vector $\mathbf{P}(\mathbf{x})$ contains monomials (local basis functions) ensuring consistency up to a desired order. For a problem in R^3, for example, to ensure zeroth order consistency

$$\mathbf{P}(\mathbf{x})^T = [1]$$

and the resulting moving least squares shape functions in (8.17) are Shepard partition of unity functions (see chapter 2). For first order consistency

$$\mathbf{P}(\mathbf{x})^T = [1,\, x,\, y,\, z]$$

and so on. The function $W_J(\mathbf{x})$ is a compactly supported radial weighting function (discussed in chapter 2) at node J.

8.2.4 Discretized equations

Using (8.15) the discretized stress vector in (8.5) is

$$\tau(\mathbf{x}) = \sum_{J=1}^{N} \mathbf{CB}_J(\mathbf{x})\boldsymbol{\alpha}_J = \mathbf{CB}(\mathbf{x})\mathbf{U}, \tag{8.19}$$

where the strain-displacement matrix $\mathbf{B}(\mathbf{x})$ in equation (8.19) is partitioned as

$$\mathbf{B}(\mathbf{x}) = [\mathbf{B}_1(\mathbf{x})\ \mathbf{B}_2(\mathbf{x})\ \ldots \mathbf{B}_J(\mathbf{x})\ldots]$$

where

$$\mathbf{B}_J(\mathbf{x}) = \boldsymbol{\partial}_\epsilon \mathbf{H}_J(\mathbf{x}) = \begin{bmatrix} \partial h_J/\partial x & 0 & 0 \\ 0 & \partial h_J/\partial y & 0 \\ 0 & 0 & \partial h_J/\partial z \\ \partial h_J/\partial y & \partial h_J/\partial x & 0 \\ 0 & \partial h_J/\partial z & \partial h_J/\partial y \\ \partial h_J/\partial z & 0 & \partial h_J/\partial x \end{bmatrix}. \tag{8.20}$$

The discretized equations (8.12) and (8.13) may therefore be written as

$$\boldsymbol{\partial}_\epsilon^T \mathbf{CB}(\mathbf{x}_I)\mathbf{U} + \mathbf{f}^B(\mathbf{x}_I) \;=\; \mathbf{0} \qquad \text{in } \Omega \tag{8.21}$$

and

$$\mathbf{NCB}(\mathbf{x}_I)\mathbf{U} \;=\; \mathbf{f}^S(\mathbf{x}_I) \qquad \text{on } \Gamma_f \tag{8.22}$$

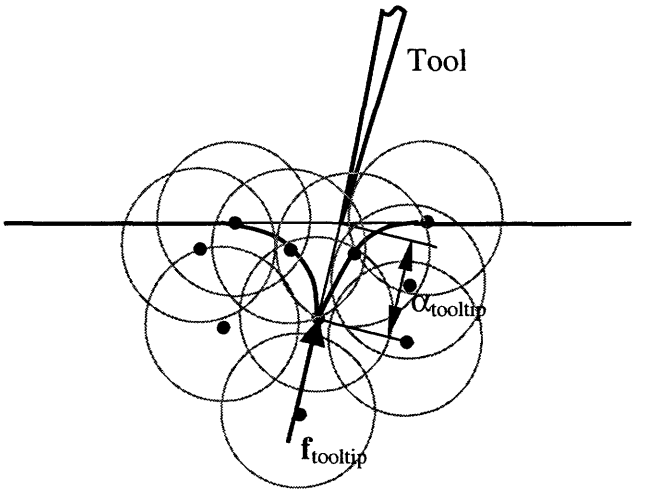


Figure 8-4: Placement of nodes at the tool tip. $\alpha_{tooltip}$ and $\mathbf{f}_{tooltip}$ are the prescribed displacement and reaction force at the tool tip, respectively.

Equations (8.21), (8.22) and (8.14) can be expressed in the compact form

$$\mathbf{KU} = \mathbf{f} \tag{8.23}$$

where $\mathbf{K}$ is the stiffness matrix (nonsymmetric but banded) and $\mathbf{f}$ is the vector containing nodal loads and applied displacements. Notice that no numerical integration is involved in setting up the system matrices.

8.3 Special issues in laparoscopic surgical simulations

In this section we specialize the point collocation approach presented in the last section to the problem of surgical simulation using laparoscopes. We observe that such a specialization substantially simplifies equation (8.23), especially if the nodes are arranged in a special configuration around the tool tip. Then we discuss the issue of real time implementation. Finally we provide algorithms for approximate incorporation of nonlinear behavior of tissues.

8.3.1 Point collocation for laparoscopic surgery simulation

Let us consider the special case of laparoscopic surgery simulation where long slender tools are used to perform surgery. The tool tip and the tissue may be modeled as having point interactions. We sprinkle the nodes around the tool tip in a special fashion. We place a

node right at the tool tip. All other nodes are placed such that their spheres do not intersect the node at the tool tip, or do so only minimally to satisfy the condition of invertibility of the matrix $\mathbf{A}(\mathbf{x})$ (see figure 8-4). From (8.15) this implies that

$$\mathbf{u}_h(\mathbf{x}_{tooltip}) = \boldsymbol{\alpha}_{tooltip} = \mathbf{u}^S \tag{8.24}$$

where $\mathbf{x}_{tooltip}$ is the position vector of the tool tip and $\boldsymbol{\alpha}_{tooltip}$ is the vector of nodal parameters corresponding to the node attached to the tool tip. Therefore the node at the tool tip bears the applied displacement at the tool tip. Furthermore

$$\mathbf{f}^S(\mathbf{x}_I) = 0 \qquad \forall \mathbf{x}_I \neq \mathbf{x}_{tooltip},\ I = 1, \ldots, N \tag{8.25}$$

and

$$\mathbf{f}^B(\mathbf{x}) = 0. \tag{8.26}$$

The stiffness matrix in equation (8.23) may be partitioned as

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ab} \\ \mathbf{K}_{ba} & \mathbf{K}_{bb} \end{bmatrix} \tag{8.27}$$

corresponding to a partitioning of the vector of nodal parameters as

$$\mathbf{U} = [\boldsymbol{\alpha}_{tooltip}\ \mathbf{U}_b]^T \tag{8.28}$$

where $\mathbf{U}_b$ is the vector of nodal unknowns ($\boldsymbol{\alpha}_{tooltip}$ is the *known* displacement at the tool tip) which may be obtained as

$$\mathbf{U}_b = -\mathbf{K}_{bb}^{-1}\mathbf{K}_{ba}\boldsymbol{\alpha}_{tooltip}. \tag{8.29}$$

The reaction force at the tool tip is obtained from the relation

$$\mathbf{f}_{tooltip} = \mathbf{K}_{aa}\boldsymbol{\alpha}_{tooltip} + \mathbf{K}_{ab}\mathbf{U}_b. \tag{8.30}$$

8.3.2 Real time issues

Let us imagine a typical surgical simulation session. The surgeon manipulates the stylus

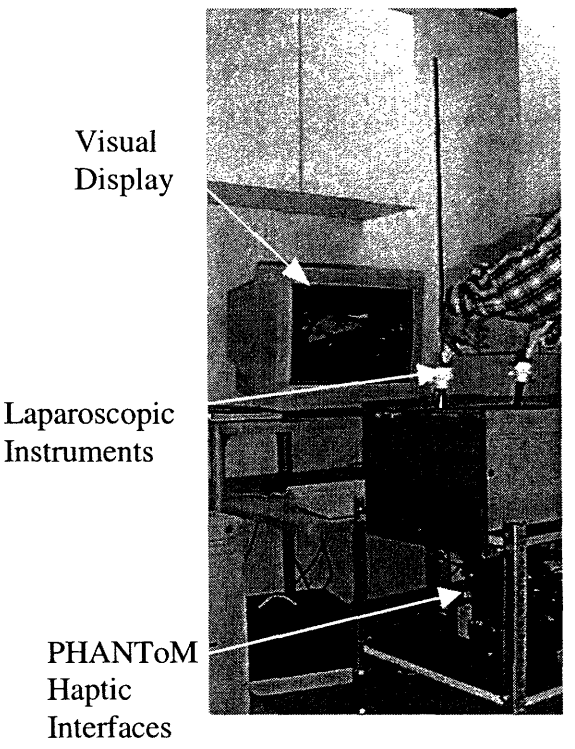


Figure 8-5: A laparoscopic surgery simulator developed at Touch lab, M.I.T.

attached to the haptic interface device (see figure 8-5) and experiences its interactions with the organ models displayed on the computer monitor in front of him. For this to occur in real time four major computational tasks are performed:

1. The collision of the virtual tool tip with the organ model is detected.
2. The MFS nodes are sprinkled around the tool tip.
3. The local displacement profile is computed from equations (8.29) and (8.15) and is displayed.
4. The force at the tool tip is computed from equation (8.30) and fed back to the user through the haptic interface device.

The organ model is usually a polygon (triangular) based surface model. Such models may be obtained from CT scanned images or MRI images of the actual organs. We use a fast collision detection algorithm developed by Ho et. al. [76] where we establish a hierarchical database of geometric primitives, with each primitive having pointers to neighboring primitives. The collision detection time is independent of the total number of polygons in the

model and the process is, of course, real time.

As soon as a collision point is detected, MFS nodes are sprinkled around the tool tip (both on the surface of the organ model as well as inside). This is a computationally intensive process that we prefer to perform off line. We assume that contact occurs at the centroid of the triangle with which collision has occurred and place a MFS node at the centroid. The other MFS nodes are placed by joining the centroid of the triangle with the vertices and projecting on to the surface of the model using the surface normal of this triangle in a manner shown in figure 8-6. The locations of the MFS nodes corresponding to a collision with each and every triangle in the surface model are pre-computed and stored as part of the polygon data structure of the model and may be retrieved very quickly during simulation. The number of spheres we may include in our simulation is completely determined by the speed of the processor. Another important issue is the choice of the radii of the spheres. While spheres with larger radii provide greater covering for fewer spheres, they increase the bandwidth of the stiffness matrix and also result in coarser approximation.

With the hardware setup that we are using currently (Pentium III, 500MHz workstation with a high end graphics accelerator card and running Windows NT) it is possible to perform the matrix setup and solution operations at most at 100Hz using about 30 spheres for the computation of displacements and forces. This implies that we can perform real time visual update. However, the haptic loop cannot be updated in real time (i.e. at 1 kHz). To circumvent this problem, we separate the visual and haptic threads [66] (see figure 8-7). The graphics loop is updated at 30Hz. To be able to update the haptics loop at 1kHz while computing the actual forces at 100Hz, we use an extrapolation technique using a Lagrangian polynomial of degree three (see figure 8-8 for details).

8.3.3 Approximate incorporation of nonlinear tissue behavior

We have already discussed that soft tissue behavior is inherently nonlinear. In this section we briefly discuss how nonlinear effects may be incorporated without sacrificing the speed of computation. First we classify the nonlinearities as

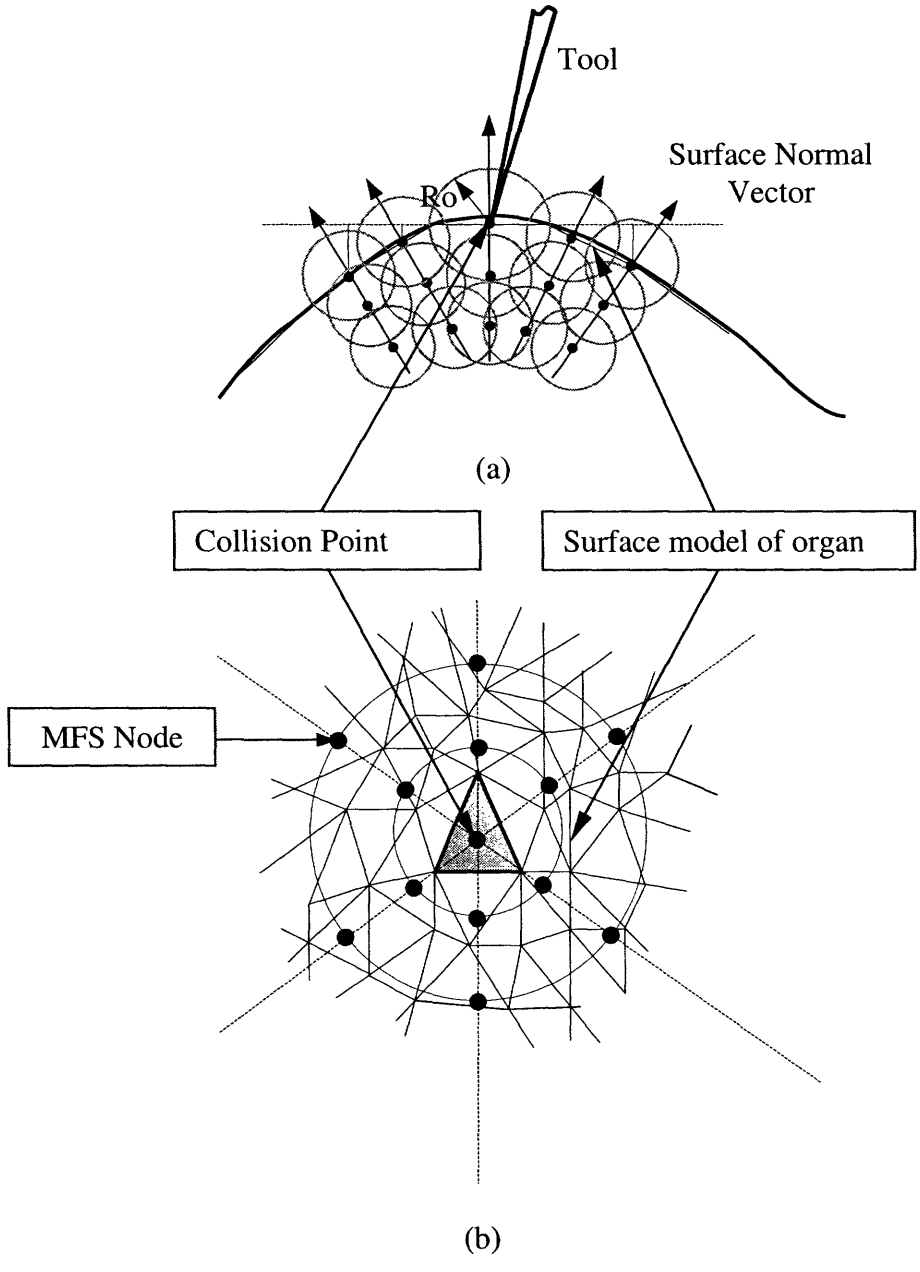


Figure 8-6: Placement of MFS nodes once a collision point has been detected. In (a) we show a section through the polygonal surface model of an organ. (b) shows a part of the top view.

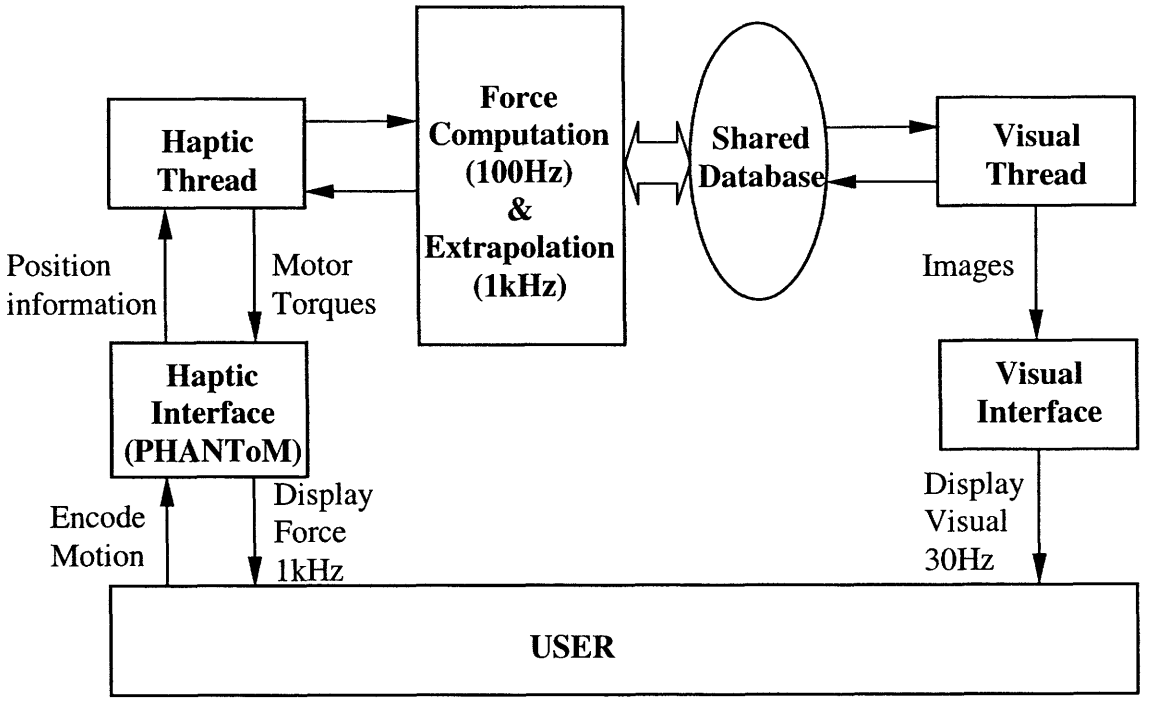


Figure 8-7: A schematic of the inter process communication in a typical multi modal surgical simulation software.

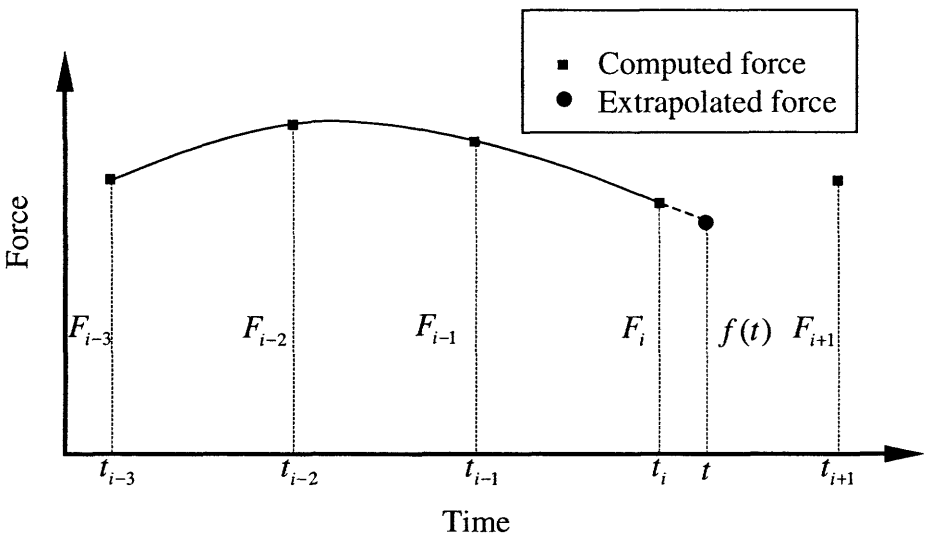


Figure 8-8: Force extrapolation to obtain real time haptic rendering. The forces F_{i-1}, F_i etc. are computed using the method of finite spheres at 100 Hz. The force at the tool tip ($f(t)$) at time t, in between times t_i and t_{i+1} is obtained by extrapolating from the forces F_i, F_{i-1}, F_{i-2} and F_{i-3} corresponding to times t_i, t_{i-1}, t_{i-2} and t_{i-3}, respectively.

1. *Geometric nonlinearities:* Nonlinearities arising from large deformations. The small strain theory where strains are related to the displacements through the relationship $\epsilon_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$ for $i, j \in \{1, 2, 3\}$ may no longer be used. Several large deformation strain measures are available in literature (see [1]).

2. *Material nonlinearities:* The stress-strain relationship may not be linear as in equation (8.10). A wide variety of nonlinear stress-strain behaviors is possible for soft tissues. An important observation is that usually the stiffness increases with increasing deformation. In this respect soft tissues behave as hardening springs.

The mathematically correct way of handling nonlinear behavior is to use an incremental step-by-step approach (see [1] for a detailed discussion). We assume that the solution at a discrete time step t is known ("time" is just a parameter in static time-independent analysis). The solution at time $t + \Delta t$ is required (where Δt is a suitably chosen time increment). At time $t + \Delta t$ we have the basic force balance equation

$$^{t+\Delta t}\mathbf{F} = {}^{t+\Delta t}\mathbf{R} \tag{8.31}$$

where $^{t+\Delta t}\mathbf{F}$ lists the nodal point forces due to element stresses and $^{t+\Delta t}\mathbf{R}$ lists the external nodal point loads at time $t + \Delta t$. We linearize about the solution at time t and write

$$^{t+\Delta t}\mathbf{F} \doteq {}^{t}\mathbf{F} + {}^{t}\mathbf{K}\mathbf{U} \tag{8.32}$$

where $^{t}\mathbf{K}$ is the tangent stiffness matrix which corresponds to the geometric and material conditions at time t and $\mathbf{U}$ is the vector of incremental nodal coefficients. From (8.31) and (8.32) we obtain

$$^{t}\mathbf{K}\mathbf{U} = {}^{t+\Delta t}\mathbf{R} - {}^{t}\mathbf{F} \tag{8.33}$$

which is solved for $\mathbf{U}$. The nodal coefficient vector at time $t + \Delta t$ is obtained as

$$^{t+\Delta t}\mathbf{U} = {}^{t}\mathbf{U} + \mathbf{U} \tag{8.34}$$

Newton iterations are usually performed for every time increment to obtain an accurate solution[1]. However, computation of the tangent stiffness matrix may be very time consuming if full Newton Raphson iterations are employed. Moreover, for the purpose of our

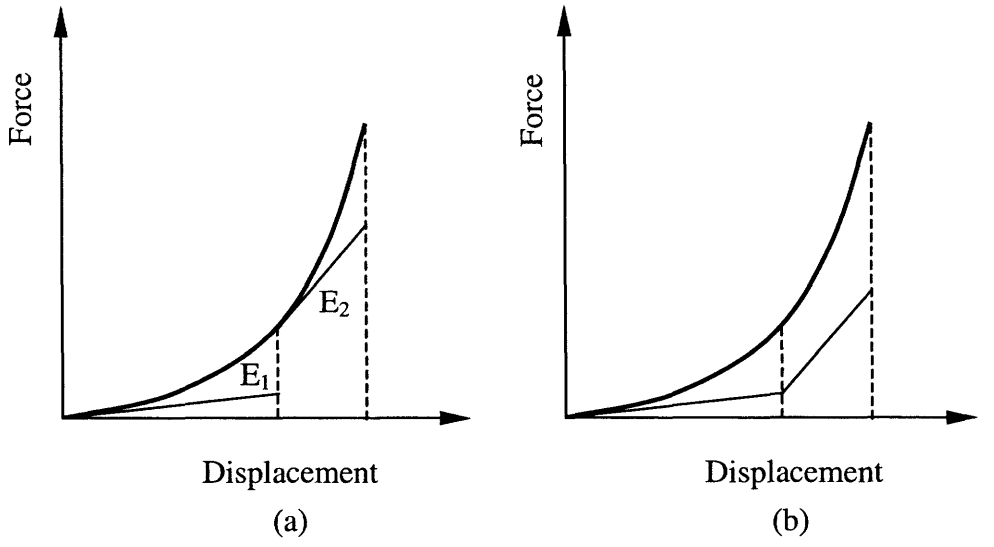


Figure 8-9: Treatment of nonlinear tissue behavior (shown for a simple one-dimensional case). In (a) the materially nonlinear only (MNO) scheme is shown. E_1 and E_2 are the Young's moduli at two different depths of indentation. In (b) the geometrically nonlinear only (GNO) scheme is shown. The thick smooth curves represent the true nonlinear force-displacement response of the tissue. The piecewise straight lines represent the approximate solution schemes.

application, such high accuracies may not be required. Hence, we propose to use a one-step Newton process, which amounts to computing the tangent stiffness matrix once every time step.

Simulation of nonlinear tissue behavior may be performed along one of the three lines:

1. *Materially nonlinear only (MNO) analysis:* In this approach we neglect large deformation effects and assume that the nonlinearity is primarily due to a nonlinear constitutive equation governing tissue behavior. To keep the discussion simple we consider a material model of the form

$$\tau(\mathbf{u}) = \mathbf{C}(\mathbf{u})\epsilon(\mathbf{u}). \tag{8.35}$$

   Specifically, we assume that the Young's modulus is a function of displacement, i.e. $E = E(\mathbf{u})$. Corresponding to different depths of indentation, we set a different value of the Young's modulus (see figure 8-9 (a)). These moduli may be obtained from actual *in vivo* experiments performed on the organ.

2. *Geometrically nonlinear only (GNO) analysis:* In this approach we assume linear elastic tissue behavior (see equation (8.10)). However, large deformation effects are considered. A rigorous geometrically nonlinear analysis would be desirable but is very expensive. We therefore propose a piece-wise linear analysis where the tangent stiffness matrix ${}^t\mathbf{K}$ is computed at each time step corresponding to the deformed configuration at time t (see figure 8-9 (b)). The solution may, however,diverge if large steps are used.

3. *Fully nonlinear analysis*: This approach takes into account both material and geometric nonlinearities.

8.4 Simulation demonstrations

We have implemented the point collocation based method of finite spheres for real time simulation and display of deformation and tool tip reaction force for certain simple 3D geometries; a hemisphere (see figure 8-10) and a liver model (see figure 8-12). In both cases, linear elastic tissue behavior has been assumed. These example problems demonstrate the applicability of the new scheme proposed here to the field of surgery simulation. However, whether the scheme is robust enough to handle the diverse problems faced by an industrial-strength surgical simulator is yet to be established.

In figure 8-10 we compare the displacement solution results obtained for the hemisphere problem using the method of finite spheres (using 34 nodes) and the finite element technique (using 27 noded volumetric brick elements). The displacement profile computed using the method of finite spheres is observed to be quite accurate in the vicinity of the tool tip. However, considerable error is observed away from the tool tip. For the purpose of surgical simulation, this displacement profile appears to be admissible. In figure 8-11 we compare the reaction at the tool tip obtained using the two methods. Since only part of the domain is modeled using the method of finite spheres, the computed force is less than that obtained using the finite element method for the same displacement of the tool tip. However, with increase in the number of spheres used for discretization (from 24 to 34) there is significant improvement in the computed reaction force.

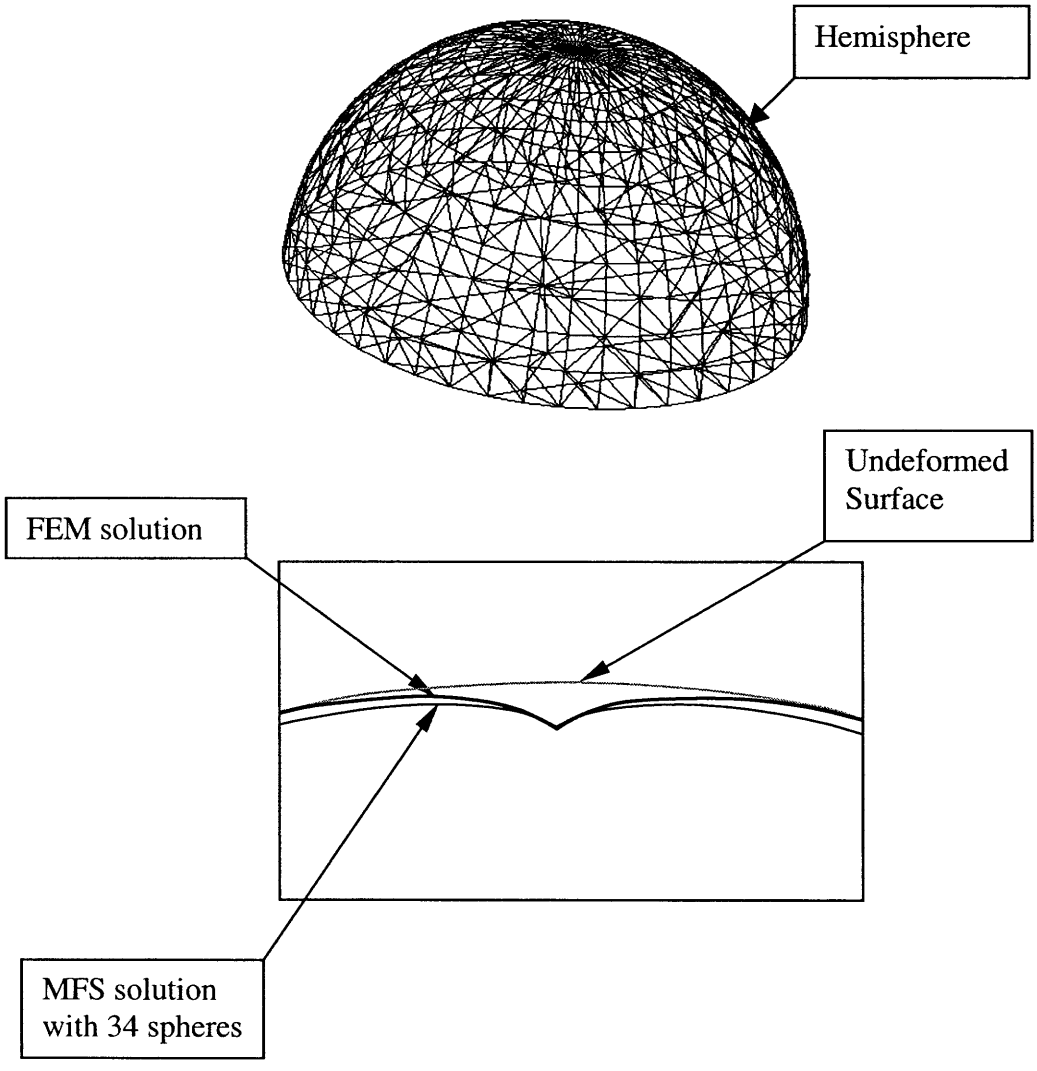


Figure 8-10: The hemisphere problem with a displacement applied at the top. The finite element solution using 27 noded isoparametric elements is compared with the solution obtained using 34 spheres around the tool tip. The two solutions are quite similar near the tool tip. However, the MFS solution differs considerably from the FEM solution away from the tool tip. This demonstrates that the MFS technique is quite satisfactory for localized solutions.

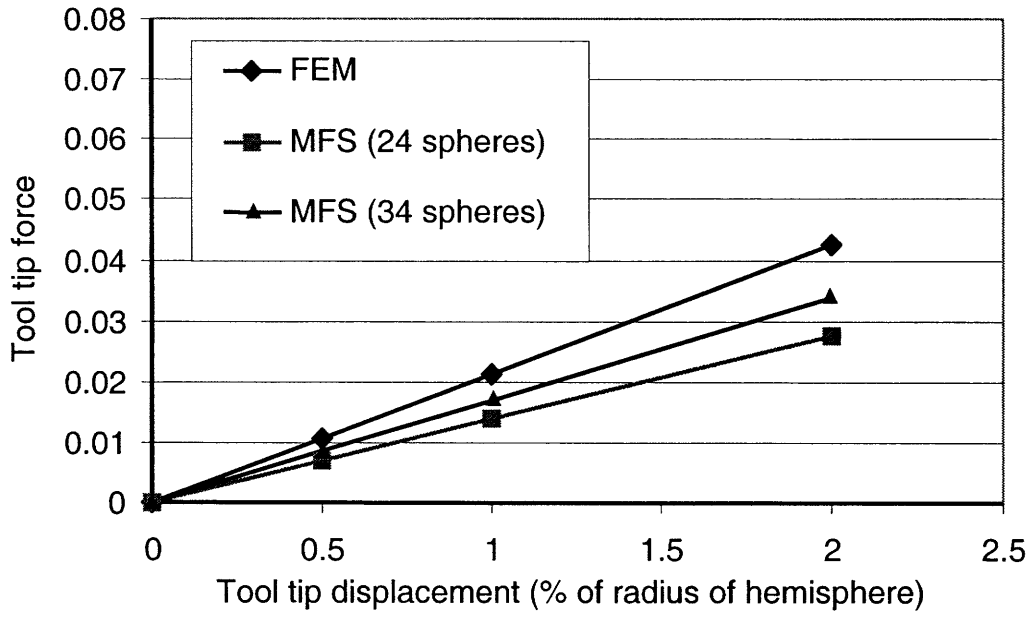


Figure 8-11: The force response at the tool tip in the hemisphere problem. The MFS solution is compared with the solution obtained using finite element modeling. The force predicted by MFS is less than that predicted by FEM since only part of the domain is discretized. However, the force response improves with increase in the number of spheres used.

8.5 Efficiency Issues

We perform a comparison of computational time along the lines of chapter 7. The total solution time is assumed to be composed of the time to generate the stiffness matrix and time to solve the resulting system of algebraic equations. From chapter 7, for the finite element technique, time for computation of the stiffness matrix may be modeled as (see chapter 7 for an explanation of the notation used)

$$T_K^{FEM} = \mathcal{O}(d^2 N_g^{FEM} N^{FEM} M^{FEM} T_h^{FEM}). \tag{8.36}$$

and the time for solution of the algebraic equations is

$$T_s^{FEM} = \mathcal{O}(dN^{FEM} \left(m_K^{FEM}\right)^2). \tag{8.37}$$

For the point collocation based method of finite spheres, we may model the cost of compu-

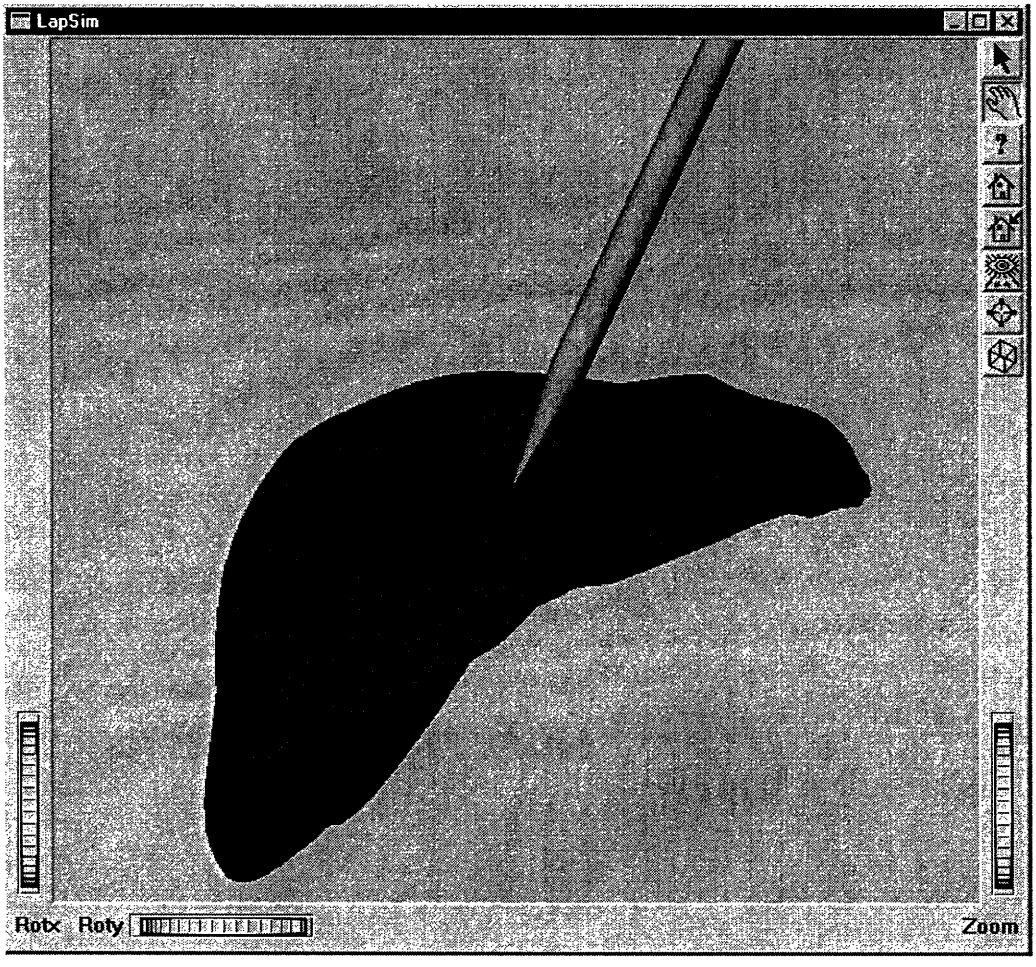


Figure 8-12: A snapshot of the laparoscopic surgical simulator LapSim showing a liver model. The deformation field, as well as tool tip reaction force are computed in real time using the point collocation based MFS.

tation of the global stiffness matrix as

$$T_K^{MFS} = \mathcal{O}(d^2 n^2 N^{MFS} (M^{MFS})^2 T_W). \tag{8.38}$$

Notice the absence of the term N_g^{MFS} since no numerical integration is involved. The cost of solution of the resulting set of linear algebraic equations may be modeled as

$$T_s^{MLS} = \mathcal{O}(dN^{MFS} \left(m_K^{MFS}\right)^2) \tag{8.39}$$

Let us consider an example of the hemisphere shown in figure 8-10. The following data was obtained

$$\begin{aligned} &N^{FEM} = 4045;\ N^{MFS} = 34; \\ &N_g^{FEM} = 27; \\ &M^{FEM} \sim 125;\ M^{MFS} \sim 4; \\ &n = 4 \text{ (for first order consistency).} \end{aligned} \tag{8.40}$$

We may therefore compute from estimates (8.36) and (8.38)

$$\frac{T_K^{MFS}}{T_K^{FEM}} \sim 6.4 \times 10^{-4} \tag{8.41}$$

assuming that T_h^{FEM} and T_W are of the same order of magnitude.

The finite element model of the hemisphere problem was solved using the commercial finite element software package ADINA. 34 nodal points were used in the method of finite spheres solution of the problem. Actual CPU times recorded when the problem was run on a Pentium III (500 MHz) workstation are presented in table 8.1. The estimate in (8.41) predicts $T_K^{MFS} \sim 4.1\ ms$ which is very close to the actually observed computation time of $6.359\ ms$ ($= 7.876\ ms - 1.517\ ms$).

With

$$\frac{T_K^{FEM}}{T_s^{FEM}} \sim 0.5 \tag{8.42}$$

Table 8.1: Computation times for FEM and MFS

	Total CPU time	**Time for solution of equations**
FEM	18.32s	11.88s
MFS	7.876ms	1.517ms

the ratio of total solution times is predicted as

$$\frac{T_{total}^{MFS}}{T_{total}^{FEM}} \sim \frac{1}{3}\left(T_K^{MFS}/T_K^{FEM}\right) + \frac{2}{3}\left(T_s^{MFS}/T_s^{FEM}\right). \tag{8.43}$$

With $T_s^{MFS} = 1.517\ ms$ and $T_s^{FEM} = 11.88\ s$, we may compute $T_s^{MFS}/T_s^{FEM} \sim 1.3\times10^{-4}$. Therefore, using (8.41) in (8.43)

$$\frac{T_{total}^{MFS}}{T_{total}^{FEM}} \sim 3.88 \times 10^{-4}$$

is a reasonable estimate of the ratio of the total solution times for the two techniques for elastostatic problems in three dimensions (the observed ratio is $\frac{7.876\ ms}{18.32\ s} = 4.3 \times 10^{-4}$). Of course, the point collocation based method of finite spheres is much less accurate than the finite element technique and only part of the domain is discretized.

Chapter 9

Summary and concluding remarks

In this work a computational scheme, the method of finite spheres, has been presented for the solution of boundary value problems on complex domains without the use of a mesh. In this technique, the domain to be analyzed is discretized using a set of nodal points. The approximation functions are compactly supported on general d-dimensional spheres centered at these nodes. A weighted residual scheme is used to obtain the discretized set of algebraic equations. Since no mesh is required for interpolation and numerical integration, the method of finite spheres is a truly meshless technique.

The elimination of a mesh implies that the preprocessing time is considerably reduced. Moreover, the expense associated with remeshing, a necessary operation when large deformations or changes in topology are encountered, can also be substantially reduced. However, for a meshless technique to be widely applicable, it must be reasonably **efficient** in comparison with the existing and well established finite element/finite volume schemes, and, of course, it must be **reliable**.

In our work we concentrate on these issues in the context of the method of finite spheres. The basic idea behind meshless techniques is rather straightforward and a few such schemes are currently available in literature. While a majority of these techniques are not truly meshless as they require a background mesh for numerical integration purposes, a few of them do not require a mesh for either interpolation or numerical integration. But all these methods are computationally quite inefficient compared to the traditional finite element/finite volume

techniques.

Computational efficiency is achieved by the appropriate choice of effective and low-cost approximation functions, simple integration sub-domains, efficient techniques for the incorporation of boundary conditions and effective numerical integration schemes. In this work we have presented original contributions made in each of these areas.

Spheres appear to be the natural choice of computational subdomains due to their high regularity. The support of a finite element shape function is a general d-dimensional polytope. The natural generalization is, of course, a sphere. Moreover, the intersection of two spheres is completely determined from a knowledge of their radii and coordinates of their centers.

The partition of unity paradigm is adopted for generating finite dimensional approximation spaces. The use of Shepard functions, with carefully chosen weighting/window functions, to provide the partition of unity on the analysis domain appears to be much less expensive than any of the other techniques found in literature. The major advantage is that the necessary and sufficient conditions for obtaining well conditioned system matrices (for a well-posed physical problem) is that the analysis domain is a subset of the union of the spheres. Moving least squares based meshless techniques require a much more stringent condition of overlap. This results in very high computational costs.

Another important issue is, of course, the imposition of Dirichlet boundary conditions in the absence of the Kronecker delta property. We have presented an efficient technique for that purpose by appropriately modifying the variational formulation. This technique does not introduce additional unknowns in the discretized set of equations, nor does it result in ill-conditioned system matrices. The Dirichlet boundary conditions are, however, incorporated in a weak sense.

The Galerkin weak forms involve integrals of rational functions on domains that are considerably more complex than those in the finite element techniques. Therefore specialized numerical integration rules have been developed to provide arbitrary polynomial accuracy on disks (in R^2). Specialized integration rules have also been presented for integration on

"boundary disks", disks that have nonzero intercepts on the boundary of the domain, and on the "lens" shaped region of overlap of two disks.

The displacement-based method of finite spheres is observed to exhibit volumetric locking when it is applied to the solution of incompressible (or almost incompressible) media. Such problems arise when we are concerned with the analysis of rubber-like solids, incompressible fluid flows and in inelastic response calculations (due to plasticity and creep). We have presented a mixed formulation using displacement and pressure interpolations to alleviate the problem of locking.

Such a formulation results in a saddle point problem which is indefinite and stability of the resulting system cannot be guaranteed unless the displacement and pressure approximation spaces are appropriately chosen. We have considered several displacement/pressure mixed discretization schemes and tested the stability of the schemes by performing numerical inf-sup tests.

We have applied the method of finite spheres to the solution of several representative problems in heat conduction, linear elastostatics and convection-diffusion in one and two dimensions. An interesting application is the use of spherical shells, rather than spheres, for directly modeling spherical cavities inside domains. The advantage of being able to incorporate the *a-priori* knowledge of the solution of the governing differential equations in the generation of the finite dimensional approximation spaces has also been demonstrated.

We have developed a point collocation based version of the method of finite spheres for an application of a completely different nature. The problem under consideration is real time multi modal (using visual and haptic modalities) medical simulations which requires very high update rates. To achieve high computational speeds, drastic approximations and simplifications are necessary. We have developed a localized discretization scheme using the method of finite spheres approach and point collocation. In spite of the approximations introduced, the results are quite encouraging and this technique shows considerable promise in the development of surgical simulators of the future.

We have performed computational cost analysis for the method of finite spheres as well as performance comparisons with the traditional finite element techniques and another meshless technique based on the moving least squares functions. When compared with the techniques based on the moving least squares interpolants, there is a clear advantage in computational cost since no matrix inversion is required. However, for comparable accuracy, the method of finite spheres, in its current state of implementation is about five times slower than the finite element techniques for the solution of problems in two-dimensional elastostatics.

The method of finite spheres exhibits a higher convergence rate than the finite element techniques since approximations using higher order continuity are used. Therefore considerably fewer nodes are required for comparable accuracy. Hence the size and half-bandwidth of the system matrices generated using the method of finite spheres are smaller than those generated using the finite element techniques. Therefore, the computational cost associated with the solution of the algebraic set of equations is considerably less in the method of finite spheres.

However, the numerical integration of the terms in the stiffness matrix is the major source of computational cost in the method of finite spheres. A large number of integration points is required for each disk (typically 144 integration points per disk were used in this work). This is primarily due to the rational nature of the integrands.

The current computational speed of the method of finite spheres is, however, not discouraging since the preprocessing time is considerably reduced when compared with the finite element techniques. However, further advances in computational efficiency may be achieved through the development of more effective shape functions and efficient numerical integration schemes. The key to a much faster computational scheme really lies in developing radically new integration schemes which are tailor-made for the functions encountered in the method of finite spheres.

A natural next step is to extend the technique to problems in three dimensions. New integration rules are required. Techniques for the automatic computation of intersections of

spheres with general boundary surfaces also need to be developed.

In this work static problems were considered. The performance of the method of finite spheres for problems in dynamics needs to be evaluated. We have also not considered non-linear problems. Problems involving large deformations and inelastic material response are frequently encountered in engineering analysis and it needs to be seen how the method of finite spheres performs in the solution of such problems. Another problem area that needs considerable attention is the modeling of layered media.

Some of the most challenging problems in engineering analysis are the modeling of contact with thermomechanical coupling, dynamic crack propagation, high Reynolds/Peclet number flows in multiple dimensions, compressible fluid flow problems involving shocks and fully coupled fluid/structure interaction problems. Extensive research has already been conducted (and is currently under way) in applying finite element/finite volume techniques to the solution of such problems. Whether the method of finite spheres can perform as well as or outperform the traditional mesh based techniques is yet to be seen.

While the method of finite spheres is still not as efficient as the traditional finite element/finite volume techniques, it has certain advantages since no mesh is required. However, further improvement in efficiency is possible along the directions indicated in this work and the method of finite spheres has excellent potential for application to problems in fluid and solid mechanics.

Appendix A

Inf-sup test for the 1-D convection-diffusion equation

In this appendix we consider the stability and optimality issues in the solution of the convection-diffusion equation in R^1. It is well known that the Galerkin finite element techniques perform quite well in the solution of the convection-diffusion equation when the Peclet number of the flow is small. However, unless a very fine discretization is used, the solution exhibits artificial oscillations when the Peclet number is high [1]. This indicates a loss of stability.

When the method of finite spheres is used to solve the convection-diffusion problem, we observe similar loss of stability when a *linear* local basis is used. However, in chapter 3 we showed that when an *exponential* local basis was used, no oscillations were observed. Therefore, using an exponential basis in the framework of the Galerkin formulation has the effect of stabilizing the method of finite spheres. This is an interesting observation that we explain in this appendix in the light of the inf-sup condition. As explained in chapter 6, it is quite difficult to prove analytically that a given discretization scheme satisfies the inf-sup condition. Therefore, a numerical inf-sup test is developed (see references [45, 77] for further details).

Let us consider the nondimensionalized convection-diffusion problem in chapter 3. The weak form of the problem is

Find $\theta_h \in V_h \subset H_E^1(\Omega)$ such that

$$a(\theta_h, \psi_h) = b(\psi_h) \qquad \forall \psi_h \in W_h \subset H_0^1(\Omega) \tag{A.1}$$

where $\Omega = [0,1]$ is the domain under consideration, $H_E^1(\Omega)$ contains functions that are in the first order Hilbert space and satisfy the nonhomogeneous boundary conditions of the problem and $H_0^1(\Omega)$ contains functions that are in the first order Hilbert space but satisfy homogeneous boundary conditions. V_h and W_h are finite dimensional subspaces of $H_E^1(\Omega)$ and $H_0^1(\Omega)$ respectively. In (A.1), the bilinear form

$$a(\theta_h, \psi_h) = \int_0^1 \left(\frac{d\,\psi_h}{d\,x} \frac{1}{Pe} \frac{d\,\theta_h}{d\,x} + \psi_h \frac{d\,\theta_h}{d\,x} \right) d\,x \tag{A.2}$$

and $b(\psi_h) = 0$.

PROPOSITION A.1 *The bilinear form $a(\cdot,\cdot)$ is* **continuous**, *i.e.*

$$\exists k_m > 0 \text{ such that } a(\varphi, \psi) \leq k_m \ \|\varphi\|_S \ \|\psi\|_T \qquad \forall \varphi, \psi \in H_0^1(\Omega) \tag{A.3}$$

where $\|\cdot\|_S$ and $\|\cdot\|_T$ are norms introduced to measure the trial and test functions [77].

Proof:

$$\begin{aligned}
\int_0^1 \left(\tfrac{d\psi}{dx} \tfrac{1}{Pe} \tfrac{d\varphi}{dx} + \psi \tfrac{d\varphi}{dx} \right) d\,x &\leq \int_0^1 \sqrt{2 \left(\left(\tfrac{d\psi}{dx} \tfrac{1}{Pe} \tfrac{d\varphi}{dx} \right)^2 + \left(\psi \tfrac{d\varphi}{dx} \right)^2 \right)} d\,x \\
&= \int_0^1 \sqrt{\tfrac{2}{Pe} \left(\tfrac{d\varphi}{dx} \right)^2} \sqrt{\tfrac{1}{Pe} \left(\tfrac{d\psi}{dx} \right)^2 + Pe\psi^2} d\,x \\
&\leq \sqrt{\int_0^1 \tfrac{2}{Pe} \left(\tfrac{d\varphi}{dx} \right)^2 d\,x} \sqrt{\int_0^1 \left(\tfrac{1}{Pe} \left(\tfrac{d\psi}{dx} \right)^2 + Pe\psi^2 \right) dx}.
\end{aligned}$$

The last inequality follows from the Cauchy-Schwarz inequality. Hence, if we choose

$$\|\varphi\|_S = \sqrt{\int_0^1 \frac{2}{Pe} \left(\frac{d\,\varphi}{d\,x} \right)^2 d\,x} \tag{A.4}$$

and

$$\|\psi\|_T = \sqrt{\int_0^1 \left(\frac{1}{Pe} \left(\frac{d\psi}{dx} \right)^2 + Pe\psi^2 \right) dx} \tag{A.5}$$

we obtain (A.3) with $k_m = 1$.

The condition for the stability and optimality of the finite dimensional problem (A.1) is given by the inf-sup condition on the bilinear form $a(\cdot,\cdot)$:

$$\inf_{\varphi_h \in V_h} \sup_{\psi_h \in W_h} \frac{a(\varphi_h, \psi_h)}{\|\varphi_h\|_S \|\psi_h\|_T} = \gamma_h \geq \gamma > 0 \tag{A.6}$$

PROPOSITION A.2 *When the inf-sup condition (A.6) is satisfied, the following* **optimal error bound** *may be established*

$$\|\theta - \theta_h\|_S \leq \left(1 + \frac{k_m}{\gamma}\right) \inf_{\varphi_h \in V_h} \|\theta - \varphi_h\|_S \tag{A.7}$$

where θ and θ_h are the exact solution and the numerical solution to the convection-diffusion problem, respectively [45].

Hence, if the constant γ is bounded away from zero, the term $\left(1 + \frac{k_m}{\gamma}\right)$ is well-behaved. In the numerical inf-sup test, we either compute the inf-sup parameter γ_h for a sequence of increasingly coarse discretizations (the Peclet number remaining fixed) or for a sequence of increasing Peclet number of the flow (for a given discretization). If the inf-sup parameter decreases to zero in either case, the numerical inf-sup test is not passed.

In order to obtain the inf-sup parameter, γ_h, numerically for a given discretization, we express the relationship (A.6) in matrix form

$$\inf_{\boldsymbol{\Phi}_h} \sup_{\boldsymbol{\Psi}_h} \frac{\boldsymbol{\Psi}_h^T \mathbf{A}_h \boldsymbol{\Phi}_h}{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h} \sqrt{\boldsymbol{\Psi}_h^T \mathbf{T}_h \boldsymbol{\Psi}_h}} = \gamma_h \geq \gamma > 0 \tag{A.8}$$

where

$$\|\varphi_h\|_S^2 = \boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h, \tag{A.9}$$

$$\|\psi_h\|_T^2 = \boldsymbol{\Psi}_h^T \mathbf{T}_h \boldsymbol{\Psi}_h \tag{A.10}$$

and

$$a(\varphi_h, \psi_h) = \boldsymbol{\Psi}_h^T \mathbf{A}_h \boldsymbol{\Phi}_h.$$

Here $\boldsymbol{\Phi}_h$ and $\boldsymbol{\Psi}_h$ are nodal parameters corresponding to φ_h and ψ_h, respectively. If

$$\varphi_h = \mathbf{H}\boldsymbol{\Phi}_h \text{ and } \psi_h = \mathbf{H}\boldsymbol{\Psi}_h,$$

where $\mathbf{H}$ is the matrix of shape functions, the explicit forms of the matrices $\mathbf{A}_h$, $\mathbf{S}_h$ and $\mathbf{T}_h$ are

$$\mathbf{A}_h = \int_0^1 \left(\frac{1}{Pe} \mathbf{H}_{,x}^T \mathbf{H}_{,x} + \mathbf{H}^T \mathbf{H}_{,x} \right) dx$$

$$\mathbf{S}_h = \int_0^1 \left(\frac{2}{Pe} \mathbf{H}_{,x}^T \mathbf{H}_{,x} \right) dx$$

and

$$\mathbf{T}_h = \int_0^1 \left(\frac{1}{Pe} \mathbf{H}_{,x}^T \mathbf{H}_{,x} + Pe\, \mathbf{H}^T \mathbf{H} \right) dx$$

where $\mathbf{H}_{,x} = d\mathbf{H}/dx$.

PROPOSITION A.3 *The inf-sup parameter, $\gamma_h = \sqrt{\lambda_p}$, where λ_p is the minimum nonzero eigenvalue of the generalized eigenvalue problem [77]*

$$\left(\mathbf{A}_h^T \mathbf{T}_h^{-1} \mathbf{A}_h \right) \boldsymbol{\Phi} = \lambda \mathbf{S}_h \boldsymbol{\Phi} \tag{A.11}$$

Proof: Let us define

$$f(\boldsymbol{\Phi}_h, \boldsymbol{\Psi}_h) = \frac{\boldsymbol{\Psi}_h^T \mathbf{A}_h \boldsymbol{\Phi}_h}{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h} \sqrt{\boldsymbol{\Psi}_h^T \mathbf{T}_h \boldsymbol{\Psi}_h}}$$

and let

$$\mathbf{T}_h = \mathbf{L}_h^T \mathbf{L}_h$$

be the Cholesky decomposition of $\mathbf{T}_h$. We also define $\mathbf{X}_h = \mathbf{L}_h \boldsymbol{\Psi}_h$. Hence

$$f(\boldsymbol{\Phi}_h, \boldsymbol{\Psi}_h) = \frac{\mathbf{X}_h^T \mathbf{L}_h^{-T} \mathbf{A}_h \boldsymbol{\Phi}_h}{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h} \sqrt{\mathbf{X}_h^T \mathbf{X}_h}}.$$

We now use the Cauchy-Schwarz inequality

$$\left| \mathbf{X}_h^T \mathbf{L}_h^{-T} \mathbf{A}_h \boldsymbol{\Phi}_h \right| \leq \|\mathbf{X}_h\| \left\| \mathbf{L}_h^{-T} \mathbf{A}_h \boldsymbol{\Phi}_h \right\|$$

where the Euclidean norm $\|\mathbf{X}_h\| = \sqrt{\mathbf{X}_h^T \mathbf{X}_h}$. Therefore

$$
\begin{aligned}
f(\boldsymbol{\Phi}_h, \boldsymbol{\Psi}_h) &\leq \frac{\|\mathbf{L}_h^{-T} \mathbf{A}_h \boldsymbol{\Phi}_h\|}{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h}} \\
&= \frac{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{A}_h^T \mathbf{L}_h^{-1} \mathbf{L}_h^{-T} \mathbf{A}_h \boldsymbol{\Phi}_h}}{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h}} \\
&= \frac{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{A}_h^T \mathbf{T}_h^{-1} \mathbf{A}_h \boldsymbol{\Phi}_h}}{\sqrt{\boldsymbol{\Phi}_h^T \mathbf{S}_h \boldsymbol{\Phi}_h}}
\end{aligned}
$$

If λ_p is the minimum nonzero eigenvalue of (A.11) then

$$
\inf_{\boldsymbol{\Phi}_h} \sup_{\boldsymbol{\Psi}_h} f(\boldsymbol{\Phi}_h, \boldsymbol{\Psi}_h) = \sqrt{\lambda_p}.
$$

In figure A-1 we plot the inf-sup parameter computed as a function of the Peclet number of the flow for a given discretization. Two approximation schemes are considered. In one approximation scheme linear local basis is used, whereas in the other an exponential local basis is used as described in chapter 6. Clearly, the discretization scheme with linear local basis fails the inf-sup test whereas the one with exponential basis passes the test.

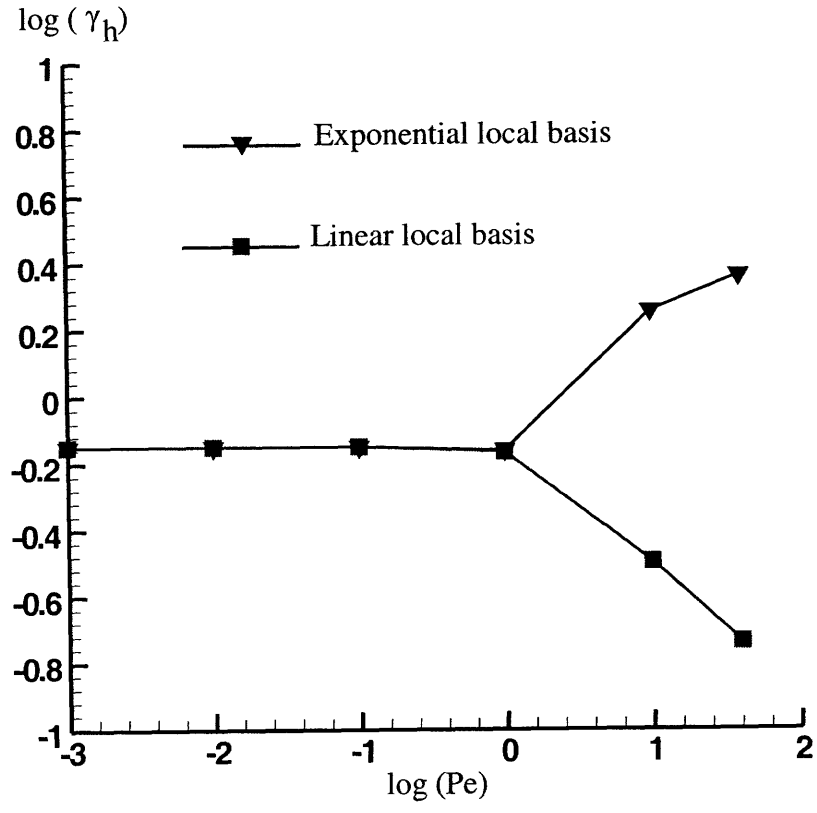


Figure A-1: Inf-sup test results for the convection diffusion problem in 1D.

Appendix B

Cubature rule for an annular sector in R^2

In this appendix we first state and then prove a product cubature rule for the integral

$$I = \iint_{\Omega} f(x,y)dxdy \doteq \sum_i \sum_j D_{ij} f(x_i, y_j) \tag{B.1}$$

with an accuracy of k. The region under consideration is the annular sector with inner radius R_i and outer radius R_o and angular span of $[-\theta_o, \theta_o]$ (see figure B-1). The dot over the equality signifies that the relationship is a strict equality if the function $f(x,y)$ is a polynomial of order at most k in x and y, otherwise it is an approximation.

The current development is based on the work of Stroud [40] and Peirce [41].

For the sake of completeness we state here a theorem of quadrature due to Gauss:

Theorem B.1 *Consider the following formula for the one-dimensional integration of a function $f(x)$ with weight $w(x)$ on the closed interval $[a,b]$:*

$$\int_a^b w(x)f(x)dx \doteq a_1 f(x_1) + a_2 f(x_2) + \cdots + a_n f(x_n). \tag{B.2}$$

The necessary and sufficient condition that a unique set of coefficients a_1, a_2, $\cdots$, a_n exist such that the above formula will have a (maximum) accuracy of $2n-1$ is that the n evaluation

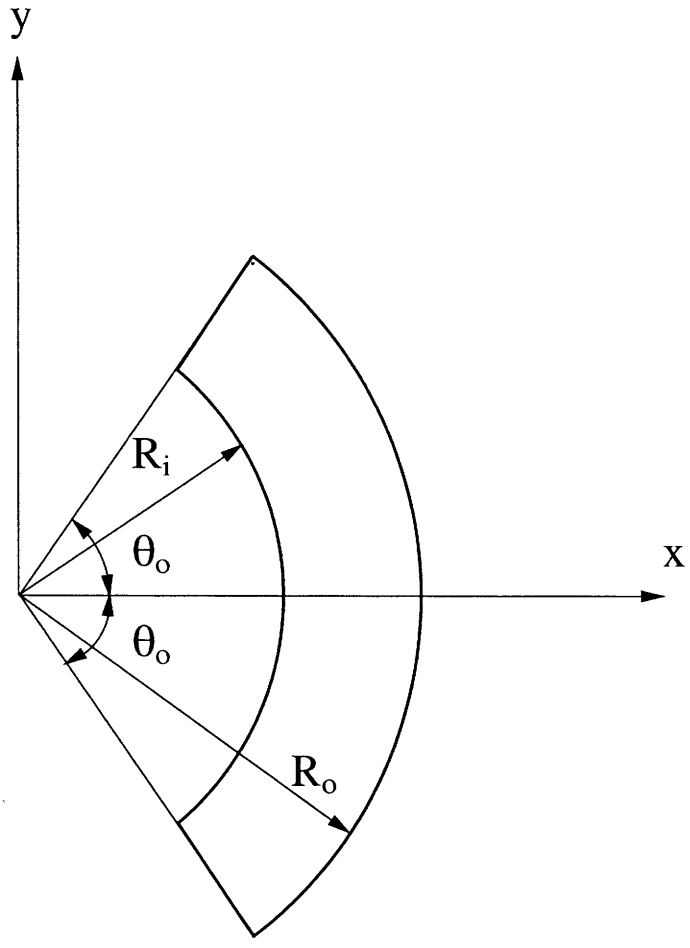


Figure B-1: An annular sector.

points x_1, x_2,$\cdots$,x_n are the n zeros of the polynomial $\pi_n(x)$ of degree n which is orthogonal to all polynomials of inferior degree over the interval $[a,b]$, relative to the weight $w(x)$. The coefficients can be uniquely determined as

$$a_i = \int_a^b w(x)[l_i(x)]^2 dx \tag{B.3}$$

where

$$l_i(x) = \frac{\pi_n(x)}{\pi_n'(x_i)(x - x_i)}. \tag{B.4}$$

This integration rule involves integration points which are real, distinct and in $[a,b]$ and the integration order of $2n-1$ is the maximum obtainable for a set of n-points. We now prove the following theorem for the integral (B.1).

Theorem B.2 : *If it is required that the rule in equation (B.1) have accuracy k=4m+3, m=0,1,2,..., in $x = r\cos\theta$ and $y = r\sin\theta$, and if it is required to have a minimum number of evaluation points which are taken at the intersection of concentric arcs (radius r_j) with rays emanating from the origin (angle θ_i), then it is both necessary and sufficient for the existence of a unique set of weights $D_{ij} \in R$ that the following two conditions be satisfied:*

(1) Let z_α be the $(k+1)/2$ zeros of the polynomial $\pi_p(z)$ of degree $p = (k+1)/2$ which is orthogonal to all polynomials of inferior degree on $[-1,1]$ relative to the weight

$$w(z) = (1+z)^{-\frac{1}{2}} \left(\frac{3 + \cos\theta_o}{1 - \cos\theta_o} - z \right)^{-\frac{1}{2}}.$$

Let angular positions φ_α be defined by the following relationship

$$\cos\varphi_\alpha = \left(\frac{1+\cos\theta_o}{2} \right) - \left(\frac{1-\cos\theta_o}{2} \right) z_\alpha \qquad \alpha = 1,2,\ldots,(k+1)/2.$$

Then the $(k+1)$ angular positions θ_i of the evaluation points are given by

$$\theta_i = \begin{cases} \varphi_i & i = 1,2,\ldots,(k+1)/2 \\ -\varphi_{i-(k+1)/2} & i = (k+3)/2,\ldots,(k+1) \end{cases}$$

(2) The radial positions of the evaluation points r_j are the positive square roots of the $m+1$ zeros of $P_{m+1}(r^2)$, the Legendre polynomial in r^2 of degree $m+1$, orthogonalized on

$[R_i^2, R_o^2]$.

The (unique) weights D_{ij} are equal to $A_i B_j$, where

$$A_i = \begin{cases} \int_{-1}^{1} w(\xi)[l_i(\xi)]^2 d\xi & i = 1, 2, \ldots, (k+1)/2 \\ \int_{-1}^{1} w(\xi)[l_{i-(k+1)/2}(\xi)]^2 d\xi & i = (k+3)/2, ..., (k+1) \end{cases}$$

and

$$B_j = \frac{1}{2P'_{m+1}(r_j^2)} \int_{R_i^2}^{R_o^2} \frac{P_{m+1}(r^2)}{r^2 - r_j^2} dr^2 \qquad j = 1, 2, ..., m+1.$$

The functions $l_i(z)$ being given as

$$l_i(z) = \frac{\pi_p(z)}{(z - z_i)\pi'_p(z_i)}.$$

Proof: We transform to polar coordinates using

$$\begin{aligned} x &= r\cos\theta \\ y &= r\sin\theta. \end{aligned} \tag{B.5}$$

The integral in (B.1) can be written as:

$$I = \int_{r=R_i}^{R_o} r g(r) dr \tag{B.6}$$

where

$$g(r) = \int_{\theta=-\theta_o}^{\theta_o} F(r, \theta) d\theta \tag{B.7}$$

and $F(r, \theta) = f(r\cos\theta, r\sin\theta)$.

We now develop two formulas, namely

$$g(r) = \int_{\theta=-\theta_o}^{\theta_o} F(r, \theta) d\theta \doteq \sum_{i=1}^{p} A_i F(r, \theta_i) \tag{B.8}$$

and

$$I = \int_{r=R_i}^{R_o} r g(r) dr \doteq \sum_{j=1}^{q} B_j g(r_j) \tag{B.9}$$

and combine them together (i.e. substitute equation (B.8) into equation (B.9)) and obtain the formula in equation (B.1) with $D_{ij} = A_i B_j$.

Lets first look at formula (B.8). The integer p is to be determined from the condition that this formula is exact whenever $F(r,\theta)$ is a polynomial of degree at most k in $r\cos\theta$ and $r\sin\theta$. We split the integral in equation (B.8) as

$$g(r) = g^+(r) + g^-(r) \tag{B.10}$$

where

$$g^+(r) = \int_{\theta=0}^{\theta_o} F(r,\theta)d\theta \tag{B.11}$$

and

$$g^-(r) = \int_{\theta=-\theta_o}^{0} F(r,\theta)d\theta \tag{B.12}$$

and develop the following rules

$$g^+(r) = \int_{\theta=0}^{\theta_o} F(r,\theta)d\theta \doteq \sum_{i=1}^{p^+} A_i^+ F(r,\theta_i^+) \tag{B.13}$$

and

$$g^-(r) = \int_{\theta=-\theta_o}^{0} F(r,\theta)d\theta \doteq \sum_{i=1}^{p^-} A_i^- F(r,\theta_i^-). \tag{B.14}$$

It is actually necessary to develop only one of (B.13) and (B.14). To see this, we make the substitution $\varphi = -\theta$ in (B.14) and obtain using (B.13)

$$g^-(r) = \int_{\varphi=0}^{\varphi_o} F(r,-\varphi)d\theta \doteq \sum_{i=1}^{p^+} A_i^+ F(r,-\varphi_i). \tag{B.15}$$

Comparing the right hand sides of (B.14) and (B.15) we observe that

$$\left.\begin{aligned} p &= p^+ = p^- \\ A_i^+ &= A_i^- \\ \theta_i^- &= -\theta_i^+ \end{aligned}\right\} \tag{B.16}$$

Let us now develop rule (B.13). We make the following transformation

$$z = \frac{2}{1-\cos\theta_o}\left[\frac{1+\cos\theta_o}{2} - \cos\theta\right]$$

and write the integral in (B.13) as

$$g^+(r) = \int_{z=-1}^{1} w(z)G(r,z)dz \tag{B.17}$$

where

$$w(z) = (1+z)^{-\frac{1}{2}} \left(\frac{3+\cos\theta_o}{1-\cos\theta_o} - z \right)^{-\frac{1}{2}} \tag{B.18}$$

and $G(r,z) = F(r,\theta(z))$. Now we use Theorem B.1 and identify

$$p^+ = \frac{(k+1)}{2}. \tag{B.19}$$

The integration points are given by the angular positions θ_i^+ which are computed from the following relationship

$$\cos\theta_i^+ = \frac{1+\cos\theta_o}{2} - \frac{1-\cos\theta_o}{2} z_i$$

where the z_i s are the $(k+1)/2$ zeros of the polynomial $\pi_p(z)$ which is orthogonal to all polynomials of inferior degree with respect to the weight function $w(z)$, i.e.

$$\pi_p(z_i) = 0.$$

The integration weights are (uniquely) given by

$$A_i^+ = \int_{\xi=-1}^{1} w(\xi)[l_i(\xi)]^2 d\xi$$

where

$$l_i(\xi) = \frac{\pi_n(\xi)}{\pi_n'(\xi_i)(\xi - \xi_i)}.$$

This completes the development of rule (B.8).

Now we develop rule (B.9)

$$I = \int_{r=R_i}^{R_o} rg(r)dr = \frac{1}{2} \int_{r=R_i^2}^{R_o^2} g(r^2)dr^2 \doteq \sum_{j=1}^{q} B_j g(r_j).$$

The integer q is determined from the fact that this formula has to be accurate whenever $g(r)$ is a polynomial of order at most $(k-1)$ in r, or a polynomial of order at most $(k-1)/2$

in r^2. Using Theorem B.1, we obtain $2q-1=(k-1)/2$, i.e. $q=(k+1)/4$. The integration points, r_j, are the positive square roots of the q zeros of the Legendre polynomial, $P_q(r^2)$ of degree q, normalized on $[R_i^2, R_o^2]$, i.e.

$$P_q(r_j^2)=0.$$

The integration weights are given by

$$B_j=\frac{1}{2P_q'(r_j^2)}\int_{R_i^2}^{R_o^2}\frac{P_q(r^2)}{r^2-r_j^2}dr^2$$

For q to be an integer, $(k+1)$ must be divisible by 4. Hence we choose $m=(k+1)/4$, where m is an integer. This completes the proof of Theorem B.2. We now make the following remarks.

Remark B.1 An important corollary of this theorem is the case when $\theta_o=\pi$, i.e. when we have internal disks or annuli. The complicated polynomial $\pi_p(z)$ in the above theorem reduces to the well known Chebyshev polynomial of the first kind and the quantities θ_i and A_i are obtained in closed form. This case has been already considered in chapter 4 and was previously studied by Peirce [41].

Remark B.2 There are $4(m+1)^2$ integration points irrespective of θ_o.

Remark B.3 All the integration points are real, distinct and lie within the annular sector.

Bibliography

[1] K. J. Bathe, *Finite Element Procedures.* Prentice Hall, Englewood Cliffs, N. J. (1996).

[2] 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).

[3] T. Liszka, An Interpolation Method for an Irregular Net of Nodes, *International Journal for Numerical Methods in Engineering,* **20**, 1599-1612 (1984).

[4] W. Tworzydlo, Analysis of Large Deformations of Membrane Shells by the Generalized Finite Difference Method, *Computers and Structures*, **27**(1), 39-59 (1987).

[5] W. Tworzydlo, The FDM in Arbitrary Curvilinear Coordinates – Formulation, Numerical Approach and Applications, *International Journal for Numerical Methods in Engineering,* **28**, 261-277 (1989).

[6] L. B. Lucy, A Numerical Approach to the Testing of the Fission Hypothesis, *Astronomical Journal*, **82**, 1013-1024 (1977).

[7] R. A. Gingold and J. J. Monaghan, Kernel Estimates as a Basis for General Particle Methods in Hydrodynamics, *Journal of Computational Physics*, **46**, 429-453 (1982).

[8] J. J. Monaghan, Why Particle Methods Work, *SIAM J. Sci. Stat. Comput.*, **3**(4), 422-433 (1982).

[9] J. J. Monaghan, An Introduction to SPH, *Computer Physics Comm*unicationss, **48**, 89-96 (1988).

[10] B. Nayroles, G. Touzot and P. Villon, Generalizing the FEM: Diffuse Approximation and Diffuse Elements, *Computational Mechanics*, **10**, 307-318 (1992).

[11] T. Belytschko, Y. Y. Lu and L. Gu, Element Free Galerkin Methods, *International Journal for Numerical Methods in Engineering*, **37**, 229-256 (1994).

[12] Y. Y. Lu, T. Belytschko and L. Gu, A New Implementation of the Element Free Galerkin Method, *Computer Methods in Applied Mechanics and Engineering*, **113**, 397-414 (1994).

[13] W. K. Liu, J. Adee and S. Jun, Reproducing Kernel Particle Methods for Elastic and Plastic Problems, *Advanced Computational Methods for Material Modeling.* Eds. D. J. Benson and R. A. Asaro, AMD 180 and PVP 33, ASME, 175-190 (1993).

[14] W. K. Liu, S. Li and T. Belytschko, Moving Least-Square Reproducing Kernel Methods, Part I: Methodology and Convergence, *Computer Methods in Applied Mechanics and Engineering*, **143**, 113-154 (1997).

[15] S. Li and W. K. Liu, Moving Least-Square Reproducing Kernel Method, Part II: Fourier Analysis, *Computer Methods in Applied Mechanics and Engineering*, **139**, 159-193 (1996).

[16] J. M. Melenk, Finite Element Methods with Harmonic Shape Functions for Solving Laplace's Equation, *Master of Arts Dissertation Thesis, University of Maryland* (1992).

[17] J. M. Melenk and I. Babuška, The Partition of Unity Finite Element Method: Basic Theory and Applications, *Computer Methods in Applied Mechanics and Engineering,* **139**, 289-314 (1996).

[18] J. M. Melenk and I. Babuška, Approximation with Harmonic and Generalized Harmonic Polynomials in the Partition of Unity Method, *Computer Assisted Mechanics and Engineering Sciences*, **4**, 607-632 (1997).

[19] I. Babuška and J. M. Melenk, The Partition of Unity Method, *International Journal for Numerical Methods in Engineering,* **40**, 727-758 (1997).

[20] 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).

[21] C. A. Duarte and J. T. Oden, An *h-p* Adaptive Method Using Clouds, *Computer Methods in Applied Mechanics and Engineering*, **139**, 237-262 (1996).

[22] S. Li and W. K. Liu, Reproducing Kernel Hierarchical Partition of Unity, Part I—Formulation and Theory, *International Journal for Numerical Methods in Engineering*, **45**, 251-288 (1999).

[23] S. Li and W. K. Liu, Reproducing Kernel Hierarchical Partition of Unity, Part II—Applications, *International Journal for Numerical Methods in Engineering*, **45**, 289-317 (1999).

[24] E. Oñate, S. 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*, Vol. 39, 3839-3866 (1996).

[25] T. Zhu, J. D. Zhang, S. N. Atluri, A Local Boundary Integral Equation (LBIE) Method in Computational Mechanics and a Meshless Discretization Approach, *Computational Mechanics*, **21**, 223-235 (1998).

[26] S. N. Atluri, H. Kim and J. Y. Cho, A Critical Assessment of the Truly Meshless Local Petrov-Galerkin (MLPG), and Local Boundary Integral Equation (LBIE) Methods, *Computational Mechanics*, **24**, 348-372 (1999).

[27] S. N. Atluri, J. Y. Cho and H. Kim, Analysis of Thin Beams, using the Meshless Local Petrov-Galerkin Method, with Generalized Moving Least Squares Interpolations, *Computational Mechanics*, **24**, 334-347 (1999).

[28] S. N. Atluri and T. Zhu, A New Meshless Local Petrov-Galerkin (MLPG) Approach in Computational Mechanics, *Comp. Mechanics*, **22**, 117-127 (1998).

[29] S. N. Atluri and T. L. Zhu, A New Meshless Local Petrov-Galerkin (MLPG) Approach to Nonlinear Problems in Computer Modeling and Simulation, *Comput. Mod. and Simulation in Engrg.*, Vol. 3, No. 3, 187-196 (1998).

[30] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming and P. Krysl, Meshless Methods: An Overview and Recent Developments. *Computer Methods in Applied Mechanics and Engineering*, **139**, 3-47 (1996).

[31] C. A. Duarte, A Review of Some Meshless Methods to Solve Partial Differential Equations, *TICAM Report 95-06, TICAM, The University of Texas at Austin* (1995).

[32] C. A. Duarte, I. Babuška and J. T. Oden, Generalized Finite Element Methods for Three Dimensional Structural Mechanics Problems, Submitted to *Computers and Structures*, (1999).

[33] J. T. Oden, C. A. Duarte and O. C. Zienkiewicz, A New Cloud-Based *hp* Finite Element Method, *Computer Methods in Applied Mechanics and Engineering*, **153**, 117-126 (1998).

[34] S. De and K. J. Bathe, The Method of Finite Spheres, *Computational Mechanics*, **25**, 329-345 (2000).

[35] D. Shepard, A Two-Dimensional Interpolation Function for Irregularly Spaced Data, *Proc. 23rd Nat. Conf. ACM*, 517-524 (1968).

[36] K. Yosida, *Functional Analysis.* 5th Edn. Springer-Verlag, Berlin Heidelberg (1978).

[37] T. Zhu and S. N. Atluri, A Modified Collocation Method and a Penalty Formulation for Enforcing the Essential Boundary Conditions in the Element Free Galerkin Method, *Computational Mechanics*, **21**, 211-222 (1998).

[38] G. J. Wagner and W. K. Liu, Application of Essential Boundary Conditions in Mesh-Free Methods: a Corrected Collocation Method, Submitted to *Intl. J. Numer. Methds. Engrg.* (1999).

[39] Y. Krongauz and T. Belytschko, Enforcement of Essential Boundary Conditions in Meshless Approximations Using Finite Elements, *Computer Methods in Applied Mechanics and Engineering*, **131**, 133-145 (1996).

[40] A. H. Stroud, *Approximate Calculation of Multiple Integrals.* Prentice-Hall, Inc., Englewood Cliffs, N.J. (1971).

[41] W. H. Peirce, Numerical Integration Over the Planar Annulus, *J. of the Soc. for Ind. and App. Math.*, Vol. 5, Issue 2, 66-73 (1957).

[42] W. H. Peirce, Numerical Integration Over the Spherical Shell, *Math. Tables and Other Aids to Computation*, Vol. 11, Issue 60, 244-249 (1957).

[43] F. G. Lether, A Generalized Product Rule for the Circle, *SIAM J. on Num. Anal.*, Vol. 8, Issue 2, 249-253 (1971).

[44] S. Stoyanova, Cubature Formulae of the Seventh Degree for the Hypersphere, *J. Comput. Appl. Math.* **84**, 15-21 (1997).

[45] K. J. Bathe, The Inf-sup Condition and its Evaluation for Mixed Finite Element Methods, *Computers & Structures*, in press.

[46] P. Krysl and T. Belytschko, Analysis of Thin Shells by the Element-Free Galerkin Method, *International Journal of Solids and Structures*, **33**, 3057-3080 (1996).

[47] T. Zhu and S. N. Atluri, A Modified Collocation Method and a Penalty Formulation for Enforcing the Essential Boundary Conditions in the Element Free Galerkin Method, *Computational Mechanics*, **21**, 211-222 (1998).

[48] H. Askes, R. de Borst and O. Heeres, Conditions for Locking-Free Elasto-Plastic Analyses in the Element-Free Galerkin Method, *Computer Methods in Applied Mechanics and Engineering*, **173**, 99-109 (1999).

[49] J. S. Chen, C. Pan and C. T. Wu, Large Deformation Analysis of Rubber Based on a Reproducing Kernel Particle Method, *Computational Mechanics*, **19**, 211-227 (1997).

[50] S. Li and W. K. Liu, Numerical Simulations of Strain Localization in Inelastic Solids using Meshfree Methods, Submitted to *International Journal for Numerical Methods in Engineering.*

[51] J. Dolbow and T. Belytschko, Volumetric Locking in the Element Free Galerkin Method, *International Journal for Numerical Methods in Engineering*, **46**, 925-942 (1999).

[52] J. S. Chen, S. Yoon, H. P. Wang and W. K. Liu, An Improved Reproducing Kernel Particle Method for Nearly Incompressible Finite Elasticity Solids, *Computer Methods in Applied Mechanics and Engineering*, **181**, 117-145 (2000).

[53] J. S. Chen, H. P. Wang, S. Yoon and Y. You, Some Recent Improvements in Meshfree Methods for Incompressible Finite Elasticity Boundary Value Problems with Contact, *Computational Mechanics,* **25**, 137-156 (2000).

[54] S. De and K. J. Bathe, Displacement/Pressure Mixed Interpolation in the Method of Finite Spheres, *International Journal for Numerical Methods in Engineering* (in press).

[55] O. C. Zienkiewicz, S. Qu, R. L. Taylor and S. Nakazawa, The Patch Test for Mixed Formulations, *International Journal for Numerical Methods in Engineering,* **23**, 1873-1883 (1986).

[56] A. Iosilevich, K. J. Bathe and F. Brezzi, On Evaluating the Inf-sup Condition for Plate Bending Elements, *International Journal for Numerical Methods in Engineering,* **40**, 3639-3663 (1997)

[57] P. Breitkopf, A. Rassineux, G. Touzot and P. Villon, Explicit Form and Efficient Computation of MLS Shape Functions and their Derivatives, *International Journal for Numerical Methods in Engineering*, **48**, 451-466 (2000).

[58] S. De and K. J. Bathe, Towards an Efficient Meshless Computational Technique: the Method of Finite Spheres, to appear in *Engineering Computations.*

[59] S. De and M. A. Srinivasan, Thin Walled Models for Haptic and Graphical Rendering of Soft Tissues in Surgical Simulation, *Proceeding of MMVR 7 Conference*, 94-99 (1999).

[60] T. Massie and J. K. Salisbury, The PHANTOM Haptic Interface: A Device of Probing Virtual Objects, *Proceeding of the ASME Winter Annual Meeting,* 295-301,(1994).

[61] J. K. Salisbury and M.A. Srinivasan, Phantom-Based Haptic Interaction with Virtual Objects, *IEEE Computer Graphics and Applications*, 1997. 17(5): p. 6-10.

[62] D. Terzopoulos, J. Platt, A. Barr and K. Fleischer, Elastically Deformable Models, *Computer Graphics,*. **21**(4), 205-214 (1987)

[63] D. Terzopoulos and K. Fleischer, Deformable Models, *The Visual Computer*, 4(6), 306-331 (1988).

[64] T. Sederberg and S. Parry, Free-Form Deformation of Solid Geometric Models, *Proceeding of ACM SIGGRAPH '86*, 150-160 (1986).

[65] W. Hsu, J. Hughes, and H. Kaufman, Direct Manipulation of Free-Form Deformation, *Proceeding of ACM SIGGRAPH '92*, **6**(2),147-159 (1992).

[66] C. Basdogan, C. Ho, and M. A. Srinivasan, Force Interaction in Laparoscopic Simulation: Haptic Rendering of Soft Tissue, *Proceeding of MMVR 6 Conference,* 585-391 (1998).

[67] S. Cover, N. Ezauerra and J. O'Brien, Interactively Deformable Models for Surgery Simulation, *IEEE Computer Graphics Application Magazine*, 68-75 (1992).

[68] D. Terzopoulos and K. Waters, Physically Based Facial Modeling, Analysis and Animation, *Journal of Visualization and Computer Animation*, **1**, 73-80 (1990).

[69] S. F. F. Gibson and B. Mirtich, A Survey of Deformable Modeling in Computer Graphics, MERL Report, 1-31 (1999).

[70] S. Pieper, J. Rosen, and D. Zeltzer, Interactive Graphics for Plastic Surgery : A Task Level Analysis and Implementation. Proceeding of Computer Graphics, 127-134 (1992).

[71] M. Bro-Nielsen and S. Cotin, Real Time Volumetric Deformable Models for Surgery Simulation Using Finite Elements and Condensation. *Proceeding of Computer Graphics Forum, Eurographics 96*, 57-66 (1996).

[72] S. Cotin, H. Delingette, J. M. Clement, V. Tassetti, J. Marescaux and N. Ayache, Geometric and Physical Representations for a Simulator of Hepatic Surgery, *Proceedings of MMVR 4 Conference,* 139-151, (1996).

[73] S. Cotin, H. Delingette, and N. Ayache, Real-Time Elastic Deformations of Soft Tissue for Surgery Simulation, *IEEE Transactions On Visualization and Computer Graphics*, **5**(1), 62-73 (1999).

[74] M. Bro-Nielsen, Finite Element Modeling in Surgery Simulation, *Proceeding of IEEE*, **86**(3), 490-503 (1998).

[75] J. Berkely, S. Weghorst, H. Gladstone, G. Raugi, D. Berg and M. Ganter, Fast Finite Element Modeling for Surgical Simulation, *Proceeding of MMVR 7 Conference* (1999).

[76] C. Ho, C. Basdogan and M. A. Srinivasan, Efficient Point-Based Rendering Techniques for Haptic Display of Virtual Objects, *Presence*, **8** (5),477-491 (1999).

[77] K. J. Bathe, D. Hendriana, F. Brezzi and G. Sangalli, Inf-sup Testing of Upwind Methods, *International Journal for Numerical Methods in Engineering,* **48**, 745-760 (2000).