Titre: Title:	Numerical simulation for 2-D viscous sintering
Auteur: Author:	Guan-Qing Lin
Date:	2007
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Lin, G.-Q. (2007). Numerical simulation for 2-D viscous sintering [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7934/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/7934/
Directeurs de recherche: Advisors:	Suzanne Lacroix, & Xavier Daxhelet
Programme: Program:	Unspecified

UNIVERSITÉ DE MONTRÉAL

Numerical Simulation for 2-D Viscous Sintering

GUAN-QING LIN
DÉPARTEMENT DE GÉNIE PHYSIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE PHYSIQUE)
AVRIL 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-29241-9
Our file Notre référence
ISBN: 978-0-494-29241-9

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

Numerical Simulation for 2-D Viscous Sintering

présenté par: LIN Guan-Qing

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté par le jury d'examen constitué de:

M. TEYSSEDOU Alberto, Ph.D., président

Mme LACROIX Suzanne, D.Sc., membre et directrice de recherche

M. DAXHELET Xavier, Ph.D., membre et codirecteur de recherche

M. PONE Elio, Ph.D., membre

Acknowledgments

I would like to express my gratitude to a couple of people who have contributed directly or indirectly to the work presented in this thesis. Firstly, I would like to thank my directors – Professor Suzanne LACROIX and Dr. Xavier DAXHELET, for giving me a chance to study at this beautiful campus, and their financial support helped me to spend my most difficult time in Montreal. On the other hand, I also thank them for their help in directing my research. Fiber optics was a new field of research for me and I have learned tremendously. Thanks to Elio PONE for his pleasant and fruitful discussions.

I feel eternal gratitude to my nice colleagues – Bryan BURGOYNE and Alexandre DUPUIS. They always nicely helped me to solve the problems that I met in mathematics, writing code, using software and writing thesis. It would have been impossible to finish my program so fast without their help.

Futhermore, I would like to express my gratitude to Simon LAFLAMME for French help.

Finally, I also want to thank my parents and my wife for supporting and encouraging me to finish my master program when I met difficulties in this period.

Résumé

Dans ce mémoire de maîtrise, une méthode numérique fiable est développée pour prévoir comment la tension superficielle déforme une région de liquide visqueux, newtonien, incompressible. Une méthode intégrale est développée pour résoudre l'équation de Stokes à deux dimensions avec des conditions aux frontières de type Neumann pour des domaines à connexions multiples dans lesquels les trous intérieurs peuvent se rétrécir et même disparaître. Autrement dit, nous transformons le problème de Stokes en équations intégrales équivalentes en employant les formules de Green correspondantes.

La méthode des frontières élémentaires (MFE) est ensuite utilisée pour résoudre les équations intégrales de Stokes décrivant l'écoulement, en prenant bien soin lors de l'implémentation qu'il y ait une solution unique. La description du rétreint des trous à l'intérieur du domaine liquide nécessite, par ailleurs, une modification de la méthode.

Deux représentations décrivant le mouvement des frontières ont été utilisées : la représentation d'Euler et la représentation de Lagrange. Dans la représentation lagrangienne, une méthode de différences finies arrières à pas variable est employée pour obtenir la solution numérique des équations différentielles.

En conclusion, nous donnons quelques résultats numériques de frittage visqueux en deux dimensions et comparons l'un d'entre eux à une solution analytique exacte. Cette comparaison montre une excellente concordance. Avec la MFE, la variation relative de la surface totale du domaine est infèrieure à $0,002\%$.

Abstract

In this master thesis, a reliable numerical method is developed to predict the deformation of an incompressible Newtonian viscous fluid region driven by surface tension. Firstly, an integral method is developed to solve the two-dimensional Stokes problem with Neumann boundary conditions for multiply connected domains in which the inside holes can shrink and disappear, i.e., we transform the Stokes problem into equivalent integral equations by using *Green's formula* corresponding to the Stokes problem.

Subsequently, the Boundary Element Method (BEM) is applied to solve the integral equations of Stokes problem flow involved. During the process of the implementation of BEM, special care has to be taken for making unique solution of Stokes problem with Neumann boundary conditions. Moreover, the formulation has to be modified to describe the shrinkage holes inside the fluid domain.

Furthermore, we use two ways to describe the boundary motion, i.e., Eulerian representation and Lagrangian representation. In the Lagrangian way, a variable step, variable order Backward Difference Formula (BDF) method is applied to obtain the numerical solution of ODEs.

Finally, we give out some numerical results of 2-D viscous sintering, and compare one of them to its exact analytical solution, and show an excellent match. With the BDF method, the relative change in total area is less than 0.002%.

Condensé

Le processus de frittage visqueux se produit dans un certain nombre d'applications industrielles. La motivation principale pour écrire ce mémoire est la nécessité d'obtenir une meilleure compréhension de ce phénomène. Le modèle mathématique de ce processus physique est étudié numériquement et la méthode numérique pour résoudre le problème d'écoulement de Stokes est le sujet principal de ce mémoire. Dans ce projet, nous nous limitons au frittage associé à un écoulement visqueux. Nous débutons avec le modèle mathématique décrivant le processus du frittage visqueux. Puis, nous présentons la méthode numérique des éléments de frontière (MEF) permettant de résoudre les équations de Stokes décrivant le frittage. Nous développons également une méthode d'intégration dans le temps afin d'obtenir l'évolution de la frontière. Finalement, nous donnons quelques exemples de simulations pour des formes bidimensionnelles.

1. Frittage visqueux et ses applications

Le frittage, d'une façon générale, est le processus qui consiste à chauffer un amas granulaire constitué de nombreuses particules (par exemple de métal, de cristal ionique ou de verre) à une température assez élevée pour qu'il y ait suffisamment de mobilité pour libérer l'énergie libre excédante de la surface de l'amas, ce qui permet aux particules de se joindre. En conséquence, la cohésion de l'amas augmente avec le temps de chauffage. Le processus de frittage peut être divisé selon les trois types suivants : sans phase liquide (frittage à semi-conducteurs), avec une faible phase liquide ($< 10\%$) (frittage en phase liquide) et avec uniquement une phase liquide (frittage visqueux). Lors du frittage visqueux, le transport de matière est causé par la tension de surface.

De nos jours, les fibres optiques microstructurées (FOM) et les fibres à cristaux photoniques (FCP) sont très étudiées pour leurs propriétés extraordinaires. Parmi elles, citons l'opération unimodale de l'ultraviolet à l'infrarouge, la propagation de modes à grande aire modale avec des diamètres de cœurs supérieurs à $20\mu m$, l'augmentation des effets non-linéaires avec une dispersion optimisée et des ouvertures numériques allant de valeurs arbitrairement faibles jusqu'à $0,9$. Lorsque des dispositifs sont fabriqués à partir de ces fibres (tels des fibres effilées, des coupleurs, et d'autres), il est important de savoir comment ces fibres se déforment. C'est pourquoi il y a un besoin de modèles théoriques ainsi que d'outils de simulation numérique pour prévoir l'évolution de la section transversale d'une fibre optique microstructurée lorsqu'elles est chauffée.

2. Description mathématique du frittage visqueux en deux dimensions

Les problèmes de frittage visqueux peuvent être fondamentalement modélisés à l'aide de l'équation de Navier-Stokes qui décrit le mouvement d'un fluide d'une façon générale. Nous supposons ici que le fluide est newtonien et incompressible. Après avoir normalisé et évalué les quantités physiques impliquées dans le calcul, nous obtenons les équations d'écoulement de Stokes. D'après la thermodynamique, la tension de surface sur le contour du fluide, orientée normalement à la surface et vers l'extérieur, doit être proportionnelle à la courbure du contour. En appliquant ces conditions, l'écoulement est déterminé pour un domaine donné à un mouvement de translation-rotation du domaine près.

Puisque la dépendence temporelle n'apparaît pas explicitement dans les équations de Stokes, il faut définir une contrainte cinématique décrivant le mouvement de la région liquide. Cette équation cinématique, régissant le mouvement de frontière, est donnée soit par la représentation d'Euler, soit par la représentation de Lagrange. Il est possible de démontrer qu'une forme de contour 2D quelconque évolue par frittage vers un cercle puisque le cercle est le contour qui minimise le périmètre pour une surface donnée. Puisque nous étudions l'évolution d'un domaine fermé, il n'est nécessaire de connaître le champ d'écoulement que sur le contour du domaine. Un tel problème est aisément résolu numériquement par la MEF, car cette méthode permet d'obtenir le champ de vitesses sur le contour. Pour ce faire, il faut transformer les équations d'écoulement avec des conditions aux frontières de Neumann en un ensemble équivalent d'équations intégrales.

Nous commençons d'abord par la déterminer ces équations intégrales pour un domaine simplement connecté. Le résultat est une équation intégrale de Fredholm de deuxième espèce, exprimée en termes des potentiels à simple et double couches. Afin d'obtenir une solution unique aux équations de Stokes, nous réduisons l'opérateur intégral du potentiel à double couche à l'aide des équations du mouvement d'un corps rigide puisqu'elles sont l'espace nul de l'opérateur intégral. Outre les équations décrivant un domaine simplement connecté, nous avons également besoin de formules décrivant la déformation de domaines connectés de manière multiple. Pour y arriver, nous procédons à une réduction supplémentaire par rapport à la normale extérieure qui est une fonction propre de l'adjoint du potentiel à double couche. Cette réduction est utilisée pour décrire l'effrondrement ou l'expansion de trous à l'intérieur du domaine liquide.

3. Résolution numérique des équations de Stokes

Afin d'obtenir une solution numérique aux équations de Stokes, il faut discrétiser le contour du domaine. Dans ce projet, nous utilisons des éléments quadratiques pour toutes les équations intégrales, c'est-à-dire que la position d'un point quelconque du contour peut être trouvée à l'aide d'un polynôme d'interpolation lagrangien de second ordre. Chaque équation intégrale peut être approximée par une quadrature de Gauss. Quand le point de collocation coïncide avec un point nodal de l'élément de discrétisation du contour, les intégrants des potentiels à simple et double couches ont une singularité dans le domaine d'intégration. Dans ce cas, il faut utiliser la quadrature logarithmique de Gauss afin d'évaluer correctement les intégrales.

Pour calculer numériquement la tension de surface, la courbure et le vecteur normal extérieur, on passe un polynôme d'interpolation lagrangien du quatrième ordre par cinq points nodaux successifs. Un problème se produit lorsque la courbure du contour devient grande. Dans ce cas d'évolution du contour, les points nodaux décrivant ce contour se rapprochent, causant l'annulation numérique du numérateur et du dénominateur dans le calcul de la courbure. Il en résulte que le calcul de la tension de surface peut dévier considérablement de la réalité. Cette déviation peut mener à des instabilités numériques, de faux résultats et même une défaillance complète de la simulation numérique. Dans notre projet, nous donnons l'exemple du rétreint d'un trou elliptique occasionnant ce problème. Afin de le régler, les points nodaux doivent être redistribués lors de l'évolution du contour. Nous proposons un algorithme simple mais efficace pour la redistribution des points nodaux. Cet algorithme est basé sur le principe d'equi-distribution des points le long de l'arc de chaque contour.

4. Méthode d'intégration temporelle

Après avoir calculé le champ de vitesse sur le contour pendant un certain intervalle de temps, un pas temporel doit être fait afin d'obtenir le nouveau contour. D'une façon générale, il y a deux manières de procéder pour décrire le mouvement du contour, soit en utilisant un schéma d'Euler, soit en utilisant un schéma de Lagrange. Dans le schéma d'Euler, les points de collocation se déplacent dans la direction normale au contour; on utilise donc les composantes normales au contour du champ de vitesse. L'avantage de cette approche est qu'elle est simple et rapide pour obtenir les nouveaux contours à chaque pas de temps. L'inconvénient est la difficulté d'implémenter un schéma d'intégration de temporelle d'ordre supérieur. De plus, les pas temporels doivent être suffisamment petits afin de réduire les erreurs de propagation. Par conséquent, seulement une méthode vers l'avant simple d'Euler avec de petites pas de temps est utilisée pour obtenir les nouveaux contours.

Dans la description lagrangienne, les noeuds de discrétisation sont considérés comme des points matériels et se déplaçant selon la vitesse réelle du contour. La trajectoire des points nodaux est décrite par des équations différentielles non-linéaires qui sont résolues à l'aide d'un intégrateur multipas. L'avantage principal de cette approche est qu'un grand pas de temps peut être fait sans augmenter l'erreur. Cependant, l'inconvénient de cette méthode est sa complexité et et sa lenteur de calculs, surtout pour un grand nombre de points nodaux.

Dans ce projet, nous employons un schéma arrière d'intégration par différences finies d'ordre variable à pas variable pour résoudre les équations différentielles *abruptes* et *non-abruptes*. L'algorithme fait d'abord une prédiction avec une représentation de Nordsieck qui est ensuite corrigée par une méthode itérative de Newton modifiée. Mentionnons qu'une matrice jacobienne approximative est utilisée dans la méthode de Newton, ce qui sauve beaucoup de temps de calcul.

Pour un certain ordre de la méthode q, après $q+1$ itérations avec le même pas h, nous mettons à jour la taille du pas h et l'ordre q de la méthode. Dans la mise à jour, les erreurs locales de troncature sont évaluées pour les ordre $q-1$, q, et $q+1$. La méthode utilise ensuite le pas de taille maximal dans la suite de l'algorithme.

En utilisant la méthode décrite ci-dessus, nous donnons quelques résultats de simulation pour plusieurs contours en deux dimensions. L'un d'entre eux est comparé à la solution analytique. Nous constatons que les solutions concordent très bien, tant au niveau de l'évolution dans le temps qu'à celui du changement des contours.

5. Conclusion

Dans ce mémoire, nous avons développé un modèle mathématique et son implémentation numérique pour simuler les problèmes de frittage visqueux en deux dimensions. Puisque nous nous intéressons seulement au mouvement du contour du domaine liquide, nous avons choisi la méthode des frontières élémentaires pour résoudre ce problème de frittage. Pendant le processus de frittage, les points nodaux sur le contour se rapprochent les uns des autres. Une grande erreur dans le calcul de la courbure du contour est prévue pour les régions de grande courbure si les noeuds sont trop près les uns des autres. Nous avons ainsi choisi de redistribuer ces nœuds également le long de l'arc après un certain temps de calcul. Il s'avère que cette méthode est très efficace.

Il y a deux manières de décrire le mouvement du contour, soit par le schéma d'Euler, soit par celui de Lagrange. Dans ce mémoire nous avons présenté les deux schémas pour obtenir le nouveau contour. En particulier, nous avons présenté de manière détaillée la méthode d'intégration temporelle (en utilisant le schéma de Lagrange) pour obtenir le nouveau contour. Bien qu'elle soit plus précise qu'une méthode à pas simple, le temps de calcul augmente significativement avec le nombre de points de discrétisation.

Nous avons développé un logiciel basé sur cette méthode avec un interface utilisateur conviviale en langage C++. Il vaut la peine de mentionner que nous utilisons la méthode d'itération du gradient conjugué stabilisé (Bi-CGStab) plutôt que de résoudre les équations algébriques linéaires de manière exacte. La méthode Bi-CGStab est presque 100 fois plus rapide que la méthode de décomposition triangulaire inférieure et supérieure de la matrice (LU) ou que la méthode orthogonal-triangulaire de décomposition (QR) pour résoudre un système d'équations linéaires de 200×200

équations. D'autre part, la méthode de Bi-CGStab est plus précise que les méthodes de décomposition LU ou QR pour les systèmes d'équations algébriques linéaires mal conditionnés.

Nous avons simulé l'évolution les contours de domaines connectés de façon multiple dans plusieurs cas. Quand les résultats peuvent être comparés à des solutions analytiques, nous avons trouvés avons un bon accord entre les deux. Le changement relatif de l'aire totale du domaine liquide est inférieur à 8×10^{-4} (pour une méthode à pas simple avec $h = 0,002$) ou $1,5 \times 10^{-5}$ (pour une méthode d'intégration dans le temps).

En employant le logiciel de simulation que nous avons développé, nous avons pu simuler l'évolution des contours de différents domaines connectés de façon multiple à partir des contours initiaux. La limitation principale vient des ressources informatiques. Notons, cependant, que dans ces simulations, nous supposons que tous les contours initialement séparés le demeurent pendant le processus de frittage; nous ne traitons pas du cas où les contours se fusionnent lors de l'évolution. Une amélioration possible du programme serait d'ajouter des algorithmes pour détecter et manipuler les contours se fusionnant ainsi que pour tenir compte de l'effet de la pression positive ou negative dans les trous.

Table of Contents

List of Tables

List of Figures

List of Symbols and Abbreviations

$\mathbf{b}$ = vector of surface tension [−]

h = normalized time step size [−]

l = characteristic length of fluid [m]

$\mathbf{n}$ = vector of outer normal of boundary [−]

p_c = characteristic pressure of fluid $[\frac{\mathrm{N}}{\mathrm{m}^2}]$

q = BDF method order [−]

t_c = characteristic time of fluid [s]

$\mathbf{v}$ = vector of boundary velocity [−]

v_c = characteristic velocity of fluid $[\frac{\mathrm{m}}{\mathrm{s}}]$

$\mathbf{y}$ = vector of boundary coordinates [−]

$\mathcal{G}$ = single layer potential operator [−]

$\mathcal{H}$ = double layer potential operator [−]

J = Jacobian factor [−]

$\mathscr{J}$ = Jacobian matrix in Newton iteration method [−]

T_{ij} = stress tensor of fluid $[\frac{\mathrm{N}}{\mathrm{m}^2}]$

Greek symbols

γ = surface tension of fluid $[\frac{\mathrm{N}}{\mathrm{m}}]$

$\delta(\mathbf{x})$ = Dirac delta function [−]

δ_{ij} = Kronecker delta symbol [−]

ε_{ij} = rate of deformation tensor of fluid $[\frac{1}{\mathrm{s}}]$

η = dynamic viscosity of fluid $[\frac{\mathrm{N\cdot s}}{\mathrm{m}^2}]$

κ = curvature of boundary $[\frac{1}{\mathrm{m}}]$

ν = kinematic viscosity of fluid $[\frac{\mathrm{m}^2}{\mathrm{s}}]$

ρ = density of fluid $[\frac{\mathrm{kg}}{\mathrm{m}^3}]$

$\hat{\boldsymbol{\varphi}}^{\boldsymbol{k}}$ = orthonormal vector of rigid-body motion for k^{th} direction $[-]$

ϕ_i = Lagrangian interpolation functions $[-]$

Γ = fluid boundary $[-]$

Δ = Laplace operator $[-]$

$\mathbf{\Phi}$ = matrix of the Lagrangian finite type polynomial $[-]$

Ω = fluid domain $[-]$

Abbreviations

BDF = Backward Difference Formula

BEM = Boundary Element Method

Bi-CGStab = BiConjugate Gradient Stabilized Method

CGS = Conjugate Gradient Squared Method

FEM = Finite Element Method

Fr = Froude Number

ODEs = Ordinary Differential Equations

PBG = Photonic Band Gap Fiber

PCF = Photonic Crystal Fiber

Re = Reynolds Number

SDLP = Single- and Double-Layer Potential

List of Appendices

Introduction

The process of *viscous sintering* occurs in a number of industrial applications. The main motivation to write this thesis is the need to obtain a better theoretical understanding of this phenomenon. The mathematical model of this physical process is investigated both analytically and numerically, and the numerical solution is the main subject of this thesis.

In this introduction, we start with a brief outline describing the sintering process and physical principles that are involved. We restrict ourselves to sintering associated to a viscous flow. Moreover, we describe the motion of fluid boundary. Finally, we outline the structure of this thesis and discuss other possible application of the methods developed.

1. Viscous Sintering and Its Applications

Sintering, in general, is the process in which a granular compact of metals, ionic crystals or glasses, consisting of many particles, is heated to such a high temperature that sufficient mobility is present to release the excess free energy of the surface of compact, thereby joining the particles together. As a result, the cohesion of the compact increase with time. The sintering process may be divided into the following three types: without liquid phase (solid-state sintering), with limited ($< 10\%$) liquid phase (liquid-phase sintering) or as a liquid phase only (e.g., viscous sintering). Moreover, one can apply pressure (hot-pressing), carry out chemical reactions (reactive sintering) or both together during sintering.

An example of sintering phenomenon occurring in nature is the formation of the rock strata from sandy sediments under the influence of high pressure exerted by later deposits. One of the oldest applications of sintering by humans is the production of earthenware from clay. Nowadays, besides the more traditional products (e.g. bricks, tiles, porcelain), microstructure optical fibers or photonic crystal fiber (PCF) and photonic band gap (PBG) fiber have been studied widely, and their significance is still increasing because of their extraordinary performance (see Figure 1). Among them we can mention properties such as: single-mode operation from the UV to IR spectral regions, large mode areas with core diameters larger than $20\mu m$, highly nonlinear performance with optimized dispersion properties, and numerical aperture (NA) values ranging from arbitrarily low value up to 0.9. However, when making some devices (such as tapers, couplers, etc.) with this optical fiber, we need to know its deformation. Therefore there is a need for theoretical models and numerical simulation for predicting the shrinkage of inner holes.

In viscous sintering the material transport is caused by a volume flow driven by surface tension. In this type of sintering, there are no grain boundaries between two coalescing particles because they are made of fluids. When *viscous sintering* term is used, we refer to a viscous sintering process where the volume flow is taken to be Newtonian and incompressible. Sintering processes of this type occur in the sintering of amorphous material such as glass. At sintering temperature glasses appear to behave like incompressible Newtonian fluid reasonably well [Wu 1982].

Most studies on viscous sintering aim at better insights in densification kinetics of such a gel (compact). In particular, one is interested in the shrinkage rate as a function of important variables involved such as viscosity and particle size, thus, reflecting how time, temperature and microstructure influence the development of the densification process. In particular, one wants to know what kind of structural configuration leads to a higher free energy of gel (compact) and hence to a higher

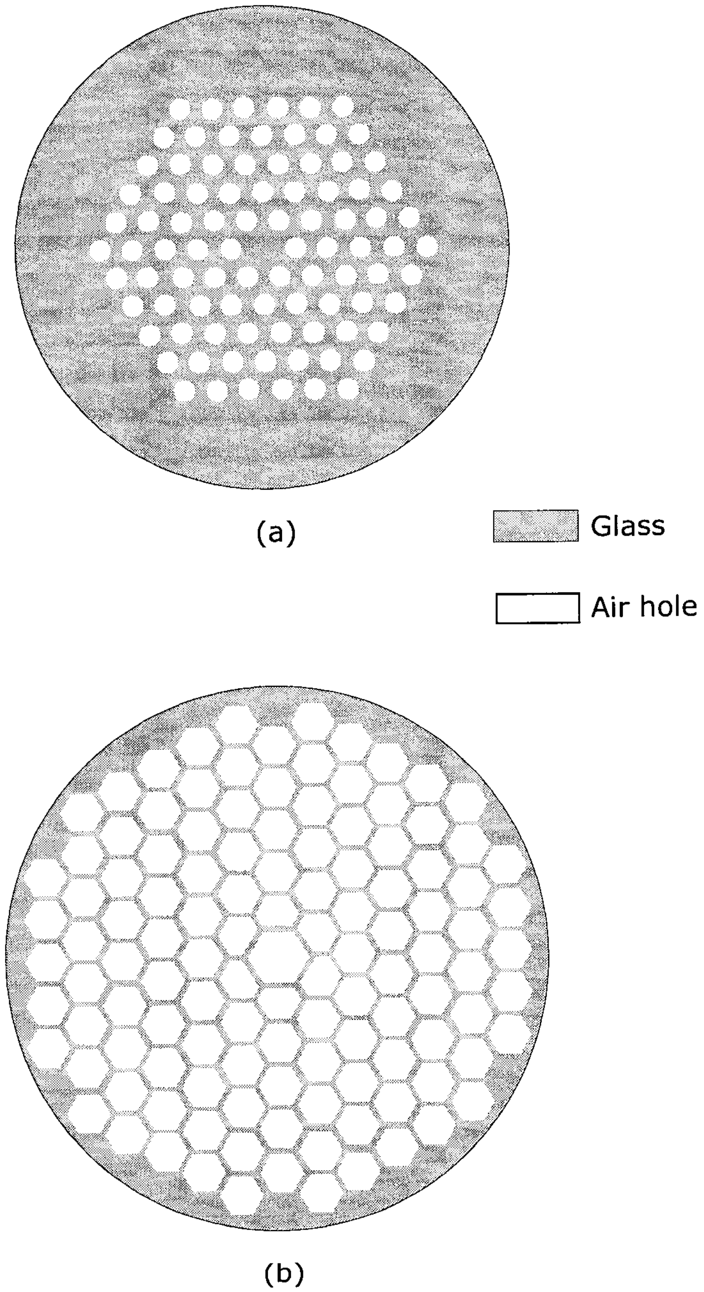


Figure 1 : Examples of photonic crystal fibers (PCFs)
(a) Endlessly single mode PCF with large mode area,
(b) air core photonic bandgap fiber (PBG) for high-power delivery and sensing.

densification rate. Moreover, one is interested to find the controlling factors that also influence the processing time and temperature.

The ultimate goal of the present research work consists of developing numerical algorithms that are able to simulate the problem of Stokes flow.

2. Analytical Approaches to Viscous Sintering

Most mathematical models of viscous sintering to date are based on an empirical rule first introduced by Frenkel [Frenkel 1945]. This rule states that the work done by surface tension in decreasing total surface area is equal to the total energy produced by dissipation of the flow. So in applying this energy balance, one has to know both the shape (e.g., deformation) and the description of the governing flow field during sintering. Generally, the deformation and flow field variables are unknown and approximate descriptions are used which only allow rough estimates of the sintering kinetics to be predicted. Frenkel's method for analyzing viscous sintering has been used for example by Mackenzie and Shuttleworth [Mackenzie & Shuttleworth 1949] and Scherer [Scherer 1984] subsequently. However, these works only describe the early stage of coalescence.

On the other hand, the Mackenzie and Shuttleworth model, i.e., MS-model, is generally accepted for describing the late stage of viscous sintering phenomena [Mackenzie & Shuttleworth 1949]. In this model, the densification results from the shrinkage of uniform spherical pores distributed throughout the compact. Hence, the MS-model is also referred to as the *closed pores model.* After applying the Frenkel's empirical rule, the MS-model leads to an equation for the sintering time necessary to reach a particular density of the compact.

Scherer developed the so-called *open pore model* that assumes the gel is formed by a regular three dimension array of interconnected liquid cylinders that shrink [Scherer 1977]. This model was used by Scherer to analyze early and intermediate stage of the sintering of gels. After applying Freknel's energy balance, he obtained an analytical relationship of the relative density variation as a function of time. However, this model breaks down in the late sintering stage.

Starting from the surface curvature driven by Stokes' flow equations, without any assumption on the deformation and flow field, Hopper solved two unit viscous sintering problems analytically [Hopper 1984] - [Hopper 1992], i.e., the coalescence of two equal cylinders and the coalescence of a cylinder on to a half-space. He applied the conformal mapping technique to solve the Stokes equations in which the evolution of the shape was described in terms of an equation of motion involving a mapping function. However, this method depends on the guess of the mapping function which, initially, has a built-in behavior that describes the evolution of the shape. The difficulty of finding such a mapping is discussed in [Hopper 1991]. Some mathematical and physical details of Hopper's method are discussed in [Graaf 1992]. Richardson was able to solve the problem of two unequal cylinders by applying Hopper's method [Richardson 1992]. An experimental verification of Hopper's solution for the coalescence of two equal cylinders has been performed by Korwin et al. They obtained a reasonably good agreement between the measured values and the analytical values [Korwin, Lange, Eaton, Joseph *et al.* 1992].

3. Numerical Solutions of Viscous Sintering

Application of computational methods by which the sintering process is simulated is relatively recent. Generally, these numerical simulations are carried out by successively solving the Stokes problem for a given geometry, and employing a time step (usually Forward Euler) to predict the new geometry.

The first attempt to simulate a unit problem of viscous sintering numerically was performed by Cosgrove et al [Cosgrove, Strozier & Seigle 1976]. They considered the sintering of an infinite line of cylinders by employing a partly analytical and numerical technique based on Fourier series. However, this method is restricted to planar surfaces.

The first genuine numerical simulation of viscous sintering problem was carried out by Ross et al [Ross, Miller & Weatherly 1981]. They considered the sintering of an infinite line of equal cylinders using the Finite Element Method (FEM) to solve the Stokes problem. Unfortunately, they did not consider the shrinkage. Jagota and Dawson were the first who could simulate some realistic problems [Jagota & Dawson 1990]. By employing the FEM, they calculated the coalescence of two equal spheres [Jagota & Dawson 1988].

The difficulty in applying the FEM lies the motion of the mesh since the error of the approximate solution depends on both the regularity of the triangulation and the angles of every single triangle. These triangles may not be reduced enough in order to fit the cusp parts of the boundary. Also, it is clear that a distortion of triangulation occurs at each time step. These difficulties regarding a moving mesh also occur in the study of Helmes who applied the FEM for the sintering of arbitrary two-dimensional fluid domains. Jagota and Dawson solved this problem by dividing the FEM mesh in a number of sub-regions after each time step [Jagota & Dawson 1990].

Kuiken simulated viscous sintering problems for domains with moderately curved shapes [Kuiken 1992]. He used an integral representation in terms of the stream and vorticity function. A BEM was employed to solve the resulting equations. In summary, this method consists of discretizing the boundary into a sequence of nodal points and both the curve and the unknowns are approximated by element-wise truncated polynomial expansions (basis function). After the work of Ingham and Kelmanson [Ingham & Kelmanson 1984], Kuiken applied this approach to solve a number of potential and biharmonic problems in which a free boundary was involved. Since the only interest is the motion of the total fluid domain, the BEM is the best way of solving sintering problems. Another advantage of the BEM is that the dimensionality of the problem is reduced by one, i.e., the two-dimensional mesh is replaced by a discretized boundary curve. However, Kuiken's approach has serious numerical problems for simulating non-smooth boundaries.

Another integral formulation of the Stokes problem is based on the distribution of hydrodynamical single- and double-layer potentials (SDLP) where the velocity and surface tension are the primary variables. In this document, this representation will be referred to as the *SDLP-formulation.* Van de Vorst and Mattheij used this method to solve the sintering problem with non-smooth boundaries, and introduced a way to distribute (redistribute) the nodal points on the boundaries [van de Vorst & Mattheij 1992]. However, the distribution technique is quite complicated and time consuming when the number of boundaries is large.

As has already been mentioned, the sintering problem is solved through the determination of the boundary velocity field. Hence, the SDLP-formulation appears to be the best choice to determine the unknown velocity field. In this thesis, the SDLP-formulation will be solved numerically by applying the BEM and a fast redistribution of nodal points on boundaries.

4. Integral Method and Numerical Solution for Stokes Flows with Moving Boundaries

In this section, we will review the application of SDLP-formulation based on hydrodynamical single- and double-layer potentials to Stokes flows with moving boundaries. The movement of boundaries is described by a kinematic constraint that relates the change of the boundary curve to the velocity at the curve, i.e., a *quasi-static* approach, is employed. The SDLP-formulation appears to be a Fredholm integral equation of the second kind [Kress 1989], [Arfken 1985]. It can be deduced by applying a fundamental singular solution of the Stokes differential equations for free space into the so-called Green's integral identity [Baker 1977]. More precisely, a fundamental solution is used to represent the velocity field in the free space induced by a force at a fixed point. This solution is also referred to as the flow due to a *Stokeslet*. A proof of the existence of the solution is given by Ladyzhenskaya [Ladyzhenskaya 1963]. The fundamental solution for geometries other than in free space, like a rigid plane wall, can be found in [Pozrikidis 1992].

In the middle of the seventies, many researchers, such as Youngren and Acrivon [Youngern & Acrivos 1975], Rallison and Acrivon [Rallison & Acrivos 1978], Lee and Leal [Lee & Leal 1982] simulated the viscous flow by using SDLP-formulation. Moreover, a generalization of SDLP-formulation developed by Power and Miranda [Power & Miranda 1987] is used to solve microhydrodynamical problems that occur in filtration, floatation, aggregation, and deposition processes [Kim & Karrila 1991].

In early numerical approaches, this integral formulation for viscous flow problems was solved by using the BEM techniques in which the velocity and tension are taken constant over each boundary element referred to as the so-called constant element. Later approaches use a high order local piecewise polynomial interpolation or global interpolations involving all nodal points by applying spline function ([Cabral, Wrobel & Brebbia 1990] - [Cabral, Wrobel & Brebbia 1991]). Thereafter, the discretized integral equation is enforced on all collocation points to produce a square full rank system of linear algebraic equations for the velocity field. This matrix equation may be solved by a direct method such as the Gaussian elimination method. However, when dealing with a large system of equations, it is necessary to use an iterative method to get the solution. Monographs that outline the numerical implementation of the BEM are, among others, Jawson and Symm [Jawaon & Symm 1991], Brebbia et al [Brebbia & Walker 1980] - [Brebbia & Dominguez 1989], and Becker [Becker 1992].

Having calculated the boundary velocity field for a given time, a time step has to be performed in order to obtain the new boundaries. The difficulties that can be encountered during the evolution of such a fluid domain are the following: (1) the boundary conditions involve the approximate computation of the curvature in order to describe the surface tension instead of the accurate values on boundaries. Furthermore, the boundaries may undergo large distortion. This provokes: (2) an uneven distribution of the collocation points. That may lead to numerical inaccuracies on both the boundary conditions as well as the determination of the unknowns. Moreover, (3) non-analytical cusp-like curves may evolve.

As mentioned before, the boundary movement is modeled by a kinematical constraint. Generally, there are three kinds of representations, namely, Eulerian, Lagrangian, and mixed Eulerian-Lagrangian constraints.

In the Eulerian viewpoint, the collocation points on the boundary move in the direction normal to the boundary, and so the normal component of the boundary velocity field is used. An advantage of this approach is that for each time step, it is quite simple and fast to obtain the position of the new boundaries. The disadvantage of this approach consists of the difficulty of implementing a higher order time integration scheme for the governing kinematic constraint. Moreover, time steps must be small enough in order to reduce propagation error. Therefore, only a simple Forward Euler method with small time steps is employed to obtain the new boundaries.

In the Lagrangian viewpoint, the nodes are considered to be material points and move according to the actual boundary velocity, i.e., the characteristic curves are followed. Actually, one obtains a system of nonlinear ordinary differential equations (ODEs) which can be solved by using a multistep integrator. The major advantage of this approach is that a large time step can be used without increasing the error. However, the disadvantage of the Lagrangian representation is its complexity and computation performance.

In the mixed Eulerian-Lagrangian method, the trajectory of a fluid particle is followed by using higher order Taylor expansion, by using the material time derivative. This explicit time step method has not been applied in the Stokes flows. However, it has been used successfully in simulation of water waves when the velocity can be modeled by Laplace's equations [Grilli, Skourup & Svendsen 1989] - [Cooker, Peregrine, Vidal & Dold 1990].

In this thesis, the SDLP-formulation is solved by applying quadratic boundary element. The kinematic constraint uses Eulerian or Lagrangian viewpoints. In order to ensure that the distribution of the nodal points remain reasonable, an efficient node redistribution scheme is developed.

5. The Content of This Thesis

In this section, we briefly outline the content of each chapter of this thesis. In Chapter 1, we describe the derivation of the mathematical formulation chosen to model viscous sintering. Basically, this physical process can be modeled by the full-scale Navier-Stokes equations which describe the motion of a fluid in general. Assuming the fluid to be Newtonian and incompressible, making the Navier-Stokes equations dimensionless and assessing the parameters of the involved physical quantities, we obtain the Stokes (creeping) flow equations. From the thermodynamic consideration, it follows that, on the fluid boundary, the surface tension in the outer normal direction has to be proportional to the average surface curvature. With this boundary condition, the flow equation can be solved for a fixed domain up to an arbitrary rigid-body motion of the domain. Since the time-dependence does not appear in the governing Stokes equations explicitly, we have to define a kinematic constraint that describes the movement of the fluid region. For this kinematic equation, we use an Eulerian or a Lagrangian description of boundary velocity field. Furthermore, we show that an arbitrary sintering domain is evolving to a sphere (or circle for 2D) as time is going, since the sphere minimizes the surface of the shape that occupies a certain amount of matter. As we discussed in the previous section, the viscous sintering problem is ideally suited to be solved numerically by the BEM. Therefore, the creeping equations are transformed into an equivalent system of integral equations, namely, the SDLP-formulation. Firstly, we consider the derivation of this formulation for a simple connected domain. It appears that the rigid-body motion function spans the null space of the involved integral operator: the double layer potential. In order to make a unique solution, this double layer is "deflated" with respect to the rigid-body motion. Secondly, we extend SDLP-formulation to a multiply connected domain. Thus, we have to apply another deflation operation with respect to the outer normal which appears to be an eigenfunction of the adjoint of the double layer potential. The latter deflation has to be performed in order to obtain the formulation that can describe the shrinkage and expansion of the holes inside the fluid domain.

In Chapter 2, we consider in more details the numerical solution of SDLP-formulation by applying BEM. This result in a full rank system of linear algebraic equations for an unknown boundary velocity field at a fixed time. Furthermore, we pay special attention to the problems that occur from the numerically approximated curvature, especially at certain cusp-like points. The curvature at such a cusp point may become very large. As a consequence, the approximation error may be large due to the numerical cancellation since the collocation points are very closed to each other in such a region. In order to handle this problem, we propose an algorithm for the node redistribution based on equidistributing arc length of the boundary.

Chapter 3 deals with the numerical solution of the motion of the fluid domain. Substituting the Lagrangian representation for the velocity of the collocation points into the system of linear algebraic equations derived from the BEM, we obtain a system of nonlinear ODEs. In this chapter, we consider the numerical integration of this ODEs. It turns out that for most of geometry shapes under consideration, the system of ODEs is stiff. Because of this, the time step will be carried out by a more sophisticated time integrator: a variable step, variable order Backward Differential Formula scheme. Thus, we have to use the implicit multistep method. The numerical implementation of this integration method is outlined. In particular we highlight some important features such as the approximation of the Jacobian matrix and the continuation of integration after mesh redistribution.

In Chapter 4, the numerical results obtained by simulating a number of two-dimensional sintering geometries are presented in order to demonstrate both the typical evolution properties and the developed code. In particular, few numerical results are compared to the corresponding analytical solutions to show the validity of the algorithm.

Chapter 1

The Mathematical Model

Viscous sintering problems can be basically modeled by a full-scale Navier-Stokes equation which describes the motion of a fluid in general. Here, we assume that the fluid is Newtonian and incompressible. After non-dimensioning these equations and assessing the parameters of the involved physical quantities, we obtain Stokes (creeping) flow equation. From the thermodynamics theory, on the fluid boundary the surface tension in the outer normal direction has to be proportional to the curvature of boundary. Applying these conditions, the flow is determined for a fixed domain up to an arbitrary rigid-body motion of the domain.

Since the time-dependence does not appear in the governing Stokes equations explicitly, we have to define a kinematic constraint that describes the movement of the fluid region. For this kinematic equation, we use the Eulerian or Lagrangian description of the boundary velocity field. Furthermore, we show that an arbitrary sintering domain evolves to a sphere (or a circle for 2D) with time, since the sphere minimizes the surface of the shape that occupies a certain amount of matter.

From the outline above, it is clear that we are only interested in the evolution of a particular, initially prescribed, fluid domain. Therefore, we require the boundary flow field of the evolving shape only. Such a problem is ideally suited to be solved numerically by the Boundary Element Method (BEM), since this method allows the boundary velocity field to be obtained. To do this, we have to reformulate the problem as an integral equation over the boundary. Thus, we transform the creeping flow equations with Neumann boundary conditions into an equivalent set of integral equations.

Firstly, we start with the derivation of this integral equation for a simple connected domain. The obtained formulation is a Fredholm integral equation of the second kind, which is expressed in terms of the single and double layers potentials (SDLP) formulation. We summarize some general theorems concerning these type of Fredholm integral equations. These properties are applied to make the integral equation uniquely determined by "deflating" the double layer potential with respect to the rigid-body motion since the motion functions represent the null space of the integral operator. Finally, we give the expression of the integral equation for multiply connected regions. It appears that, besides the deflation of the rigid-body motion, we have to apply an extra deflation with respect to the outer normal, which appears to be an eigenfuction of the adjoint of double-layer potential. This deflation is used to describe the shrinkage and expansion of holes inside the fluid domain.

1.1 Stokes Flow Equations

We consider a two-dimensional incompressible fluid flow characterized by the physical parameters: η – the dynamic viscosity; γ – the surface tension; and l – a characteristic length (the magnitude of the body, characterized by its cross-section). The dimensions of these parameters are $\mathbf{N \cdot s \cdot m^{-2}}$, $\mathbf{N \cdot m^{-1}}$, and $\mathbf{m}$ respectively. The velocity of the fluid is denoted by $\mathbf{v}$ and the pressure by p. A simply connected region of fluid is defined by closed curve Γ, and the interior area is denoted by Ω. In the case of a multiply connected domain, the domain is bounded externally by Γ_0, and internally by Γ_1, Γ_2, $\ldots$, Γ_M. The complete boundary of a multiply connected domain will be denoted Γ. The orientation of the boundary is taken in such a way that when progressing on contour Γ, the fluid region Ω lies on the *left hand side.*

The motion of the fluid is governed by the following two equations (see, among others, [Wu 1982], [Batchelor 1967]). First, we require the conservation of mass, i.e., the *continuity equation*, that, for an incompressible fluid, is reduced to:

$$\mathbf{div}\ \mathbf{v} = 0. \tag{1.1}$$

A vector field $\mathbf{v}$ that satisfies Equation (1.1) is also called *solenoidal.* Secondly, the Navier-Stokes equation (the equation of motion) holds for the flow,

$$\rho \left[\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \mathbf{grad})\mathbf{v} \right] = -\mathbf{grad}\ p + \eta \Delta \mathbf{v} + \rho g, \tag{1.2}$$

where ρ is the density of the fluid, and ρg is the gravity force (here we consider only a gravitational force). The latter equation expresses the *conservation of momentum.* Since we assume the fluid to be Newtonian, this yields the following equation for stress tensor T, which has the components given by:

$$T_{ij} = -p\delta_{ij} + 2\eta\varepsilon_{ij}, \tag{1.3}$$

where δ_{ij} is the Kronecker delta symbol, and the components of rate of deformation tensor ε_{ij} are equal to

$$\varepsilon_{ij} = \frac{1}{2} \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right). \tag{1.4}$$

Here, we can define a characteristic velocity v_c, a characteristic pressure p_c, and a characteristic time t_c, based on the parameters η, γ, and l, as

$$v_c = \frac{\gamma}{\eta}; \qquad p_c = \frac{\gamma}{l}; \qquad t_c = \frac{l\eta}{\gamma} = \frac{l}{v_c}. \tag{1.5}$$

Using these characteristic parameters, we obtain the following dimensionless form of the Navier-Stokes equation (1.2):

$$Re\left[\frac{\partial \mathbf{v}'}{\partial t'} + (\mathbf{v}' \cdot \mathbf{grad}')\mathbf{v}'\right] = -\mathbf{grad}'\, p' + \Delta\mathbf{v}' + \frac{Re}{Fr}g', \tag{1.6}$$

where $\mathbf{v}'$, p', g', and t' are the dimensionless velocity, pressure, gravity acceleration, and time respectively. The differential operator grad$'$ and Laplacian Δ' denote these operators in the dimensionless space. Here *Re* is the *Reynolds number* defined by:

$$Re = \frac{\rho l v_c}{\eta} = \frac{\rho l \gamma}{\eta^2},$$

and *Fr* is the *Froude number* which is equal to

$$Fr = \frac{v_c^2}{gl} = \frac{\gamma^2}{gl\eta^2},$$

where g is the acceleration due to gravity .

For typical sintering system of a glassy aerogel, the magnitudes of the *Reynolds number* (of the order of 10^{-21}) and *Froude number* (of the order of 10^{-13}) are very small [van de Vorst 1994], and some terms in the equation can be neglected. Hence, both the convective and gravitation forces can be neglected in comparison to the viscous forces, so that Equation (1.6) reduces to so-called *Stokes creeping flow* equations. With the dimensionless of continuity equation (1.1), we can obtain:

$$\begin{aligned} \Delta\mathbf{v}' - \mathbf{grad}'\, p' &\cong 0, \\ \mathbf{div}'\, \mathbf{v}' &= 0, \end{aligned} \tag{1.7}$$

which have to be solved together with the dimensionless form of the constitutive relation (1.3). Finally we omit $'$, and call $\mathbf{v}$, p the dimensionless velocity and pressure respectively.

As already mentioned in the previous chapter, the driving force of sintering is the excess of free energy of the surface. Physically, a molecule inside the fluid is completely surrounded by other molecules so that these molecules have equal attraction in all directions. On the other hand, a molecule at the surface is subjected to a net inwards attraction. Therefore, The fluid surface will always tend to contract to a shape with the smallest possible surface area. The work required to decrease the area is called the *free surface energy.* The energy change can be stated by the following thermodynamic relation:

$$\gamma = \left(\frac{dG}{dA}\right)_{p,T,N_i} , \tag{1.8}$$

where G is the Gibbs free energy [Waldram 1985], and A is the surface area. This equation expresses that the pressure (p), the temperature (T), and the number of species (N_i) in the system remain constant during the change in the surface area ([Reed 1988]). Here, we assume a homogeneous surface tension. If a volume element dV is removed from or added to a surface with principal curvatures κ_1, and κ_2, the energy change involved can be written as:

$$\frac{dG}{dV} = \gamma \frac{dA}{dV} = \gamma(\kappa_1 + \kappa_2). \tag{1.9}$$

The term dG/dV corresponds to a stress tensor. From Equation (1.9) the boundary condition in the normal direction for a free surface of a fluid particle can be written as ([Batchelor 1967]):

$$T_{ij}n_j = -\gamma(\kappa_1 + \kappa_2)n_i \ , \tag{1.10}$$

where a repeated index in an expression denotes a summation i.e., Einstein summation convention. Moreover, the vector $\mathbf{n}$ denotes the outward normal vector. The calculation of the principal curvatures κ_i of a surface in three-dimensional plane can be found in [Aris 1962].

Note that the tangential component of the surface tension vector must be equal to zero, because there is no fluid outside Γ that can provoke a shearing force, i.e.,

$$T_{ij} n_j \tau_i = 0, \tag{1.11}$$

where τ is the unit tangential vector at the boundary.

For a two-dimensional case, the dimensionless form of Equation (1.10) can be written as:

$$T_{ij} n_j = -\kappa n_i = b_i \ , \tag{1.12}$$

where the vector $\mathbf{b}$ will be referred to the *surface tension* in the normal direction. When the curve of the boundary is written in a parametric form, say $\boldsymbol{x} = \boldsymbol{x}(s)$, we obtain the following expression

$$\kappa(s) = \frac{x_1' x_2'' - x_2' x_1''}{\left[(x_1')^2 + (x_2')^2 \right]^{3/2}} \ , \tag{1.13}$$

where $'$ and $''$ denote the first and second derivative with respect to the parameter s, and the indices 1 and 2 denote the two directions of Cartesian coordinate system.

In the case of multiply connected regions, we also assume that the normal component of the stress vector is proportional to the local curvature of the inner hole boundaries. Here, we exclude movement of the holes as a consequence of buoyancy, i.e., there is not an extra stress component due to a gas inside those holes. This is a reasonable simplification, since we have observed that the force due to gravity can be neglected for very low Reynolds number. Furthermore, we assume that the gas properties inside the holes do not change significantly during the evolution process. Thus, its effect can be neglected.

1.2 The Uniqueness of the Solution

The uniqueness of the fixed domain solution of the above outlined Stokes problem can be formulated by the following statement.

Theorem 1.1 *The solution of the Stokes problem (1.7), subjected to the boundary surface tension in Equation (1.10) is uniquely determined by the arbitrary rigid-body motion of the domain.*

For a real flow velocity field, let Ω be an arbitrary multiply-connected domain that is bounded by a smooth boundary Γ. Using the divergence theorem, the continuity equation, and the Stokes equations, we compute the dimensionless form of the integral $\oint_\Gamma v_i b_j d\Gamma$:

$$\begin{aligned}
\oint_\Gamma v_i b_j \, \mathrm{d}\Gamma &= \oint_\Gamma v_i(-p\delta_{ij} + 2\varepsilon_{ij})n_j \, \mathrm{d}\Gamma \\
&= \oint_\Omega \left[-v_j \frac{\partial p}{\partial x_j} + 2\frac{\partial}{\partial x_j}(v_i \varepsilon_{ij}) \right] \mathrm{d}\Omega \\
&= \oint_\Omega \left[2\frac{\partial v_i}{\partial x_j}\varepsilon_{ij} + 2v_i \frac{\partial \varepsilon_{ij}}{\partial x_j} - v_j \frac{\partial^2 v_j}{\partial x_i^2} \right] \mathrm{d}\Omega \\
&= 2\oint_\Omega \varepsilon_{ij}\varepsilon_{ij} \, \mathrm{d}\Omega, \qquad (1.14)
\end{aligned}$$

where the unit normal vector $\mathbf{n}$ points outward the control domain Ω, ε_{ij} is the rate of deformation tensor defined in (1.4). Thus, we have:

$$\oint_\Gamma v_i b_i \, \mathrm{d}\Gamma = 2\oint_\Omega \varepsilon_{ij}\varepsilon_{ij} \, \mathrm{d}\Omega. \qquad (1.15)$$

The right-hand side of Equation (1.15) represents the rate of dissipation of mechanical energy within the domain Ω, and the left-hand side is the rate of work carried out by the surface tension over Γ. Therefore, Equation (1.15) represents the conservation of energy. Because the fluid does not transport momentum, i.e., the inertial and acceleration terms have been neglected, the rate of accumulation of kinetic energy into the domain is equal to zero.

Now let's assume that, over the boundary Γ, either the velocity $\mathbf{v}$ vanishes, or the surface tension $\mathbf{b}$ vanishes, or the velocity is normal to the surface tension $\mathbf{v} \cdot \mathbf{b} = 0$. Under these circumstances, the left-hand side of Equation (1.15) will be equal to zero, requiring that the rate of deformation tensor vanish ($\varepsilon_{ij} = 0$) throughout Ω and hence, therefore $\mathbf{v}$ expresses a rigid-body motion, i.e., a translation and a rotation. Assume for a moment that there are two solutions corresponding to a set of boundary conditions on Γ, either for velocity or for surface tension. The flow expressed by the difference between these solutions must express a rigid-body motion. If the boundary conditions specify the velocity over a portion of Γ, then the rigid-body motion is not permitted, any difference of the flow velocity must vanish, which implies that the solution is unique. If the boundary conditions specify the surface tension exclusively, any solution may be augmented by the addition of an arbitrary rigid-body motion.

1.3 A Quasi-static Property of the Stokes Problem

If we were to solve the Stokes equations (1.7) for a fixed domain, we would find a non-zero flow velocity field on the boundary, which would mean an inflow through one part of the boundary and an outflow elsewhere. This is unphysical since we have a material boundary: therefore the fluid domain is moving. The time-dependence does not appear explicitly in the Stokes equations (1.7). Thus, we define a *kinematic constraint* that describes the movement of the fluid domain. This treatment is called a *quasi-static* approach, which requires that the inertial force is much smaller than the viscous force, i.e., from Equation (1.2), results in:

$$\frac{\partial \mathbf{v}}{\partial t} \ll \cdots \nu \Delta \mathbf{v}, \tag{1.16}$$

or by applying a dimensional analysis

$$\frac{l^2}{\nu} \ll t_c, \tag{1.17}$$

where ν is the kinematic viscosity that is related to the dynamical viscosity η by $\nu = \eta/\rho$. The time l^2/ν denotes the time scale of the diffusion of the vorticity over a length l. In the case of sintering of glassy gel we observe that l^2/ν is order of 10^{-18} – 10^{-21}s, and the characteristic time t_c is of the order of 10^3s [Kuiken 1990], hence the usage of the quasi static approach is clearly reasonable.

As we already mentioned in the previous chapter, one of two kinematic constraints is used for describing the motion of a boundary of the viscous fluid. These constraints are derived for an Eulerian or a Lagrangian fluid representation. We note that, in fluid mechanics, a particle description is referred to as the *Lagrangian representation*, while a field description is known as the *Eulerian representation.*

The first constraint is derived based on an Eulerian representation as follows. We assume that the material boundary can be represented by a function $f(\mathbf{x}, t) = 0$. Differentiating this function with respect to the time t, yields [Wu 1982]:

$$\frac{\partial f}{\partial t} = \mathbf{grad}\, f \cdot \mathbf{v}\ . \tag{1.18}$$

Because there is no straightforward way to solve this equation, a simple forward Euler method is often applied that requires small time steps in order to describe correctly the boundary evolution. Therefore, this constraint is generally used for smooth boundary.

The second constraint is derived based on the Lagrangian representation by taking into account the material velocity given by:

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}), \qquad (\mathbf{x} \in \Gamma). \tag{1.19}$$

This equation represents the trajectories of material particles. From Equation (1.19), a system of nonlinear ODEs can be obtained, which describe the trajectories of the nodal points. In Chapter 3, we discuss the solution of this system of nonlinear differential equations.

Now, we consider the asymptotic behavior of the quasi-static Stokes flow for advancing time steps. As it is shown before, the sintering driving force arises from an excess of free surface energy. Therefore, a viscous domain transforms itself into a sphere (a circle for a 2D space), because this shape minimizes the outer surface (a perimeter in a 2D space) that occupies the domain. This can also be demonstrated from the balance of energy dissipation given by Equation (1.15), i.e,

$$\int_{\Gamma} v_i b_i \,\mathrm{d}\Gamma = -\int_{\Gamma} \kappa v_i n_i \,\mathrm{d}\Gamma = 2\int_{\Omega} \varepsilon_{ij}\varepsilon_{ij}\,\mathrm{d}\Omega \geq 0. \tag{1.20}$$

In the case of a spherical domain Ω , the integral on the left-hand side vanish, which follows from the integral of the continuity equation together with the fact that κ is uniform. Hence $\varepsilon_{ij} = 0$, and if we do not allow any rigid-body motions, this implies that $\mathbf{v} = 0$. Thus, this property can be summarized as follows:

Property 1.1 *An arbitrary sintering geometry deforms to a steady state with increasing time.*

Now, we show the limiting property for $t \longrightarrow \infty$ for the solution of a quasi-static problem. This property is derived under the assumption that the solution does exist for any time $t \geq 0$ for a certain starting shape. Here, we just briefly sketch the proof. Let $\Gamma_t = \Gamma(t)$, and let's denote the perimeter of a 2-D viscous geometry shape by $A(t)$, i.e.,

$$A(t) = \int_{\Gamma_t} \mathrm{d}\Gamma_t\,.$$

Differentiating the above equation with respect to time t, Prokert [Prokert 1993] showed that

$$\frac{d}{dt}\Big[A(t)\Big] = \int_{\Gamma_t} \kappa v_i n_i \,\mathrm{d}\Gamma_t\,.$$

Substituting the left side into Equation (1.20), we derive that the perimeter of viscous geometry shape $A(t)$ has to be *monotonously decreasing.* Since the fluid is incompressible, this means that $A(t)$ has to be bounded from below. Hence:

$$\frac{d}{dt}\Big[A(t)\Big] \rightarrow 0, \qquad\qquad (t \rightarrow \infty),$$

and from the uniqueness of $\mathbf{v}$, we obtain $\mathbf{v} \rightarrow 0$ which implies that Ω has to be a sphere as shown above.

1.4 Integral Formulation for a Simply Connected Domain

From the previous section it is clear that we are only interested in the evolution of a particular initially prescribed fluid domain. Hence, we only require the boundary flow field variables of the evolving shape. Here, we reformulate the problem as an integral equation over the boundary, i.e., we transform the two-dimensional creeping flow equations with boundary conditions into an equivalent set of integral equations. In this section, we deduce the formulation for a simply connected and multiply connected domain.

Consider Ω to be a *simply connected* domain surrounded by a closed curve Γ in the space $\mathbb{R}^2$. Here we assume that the normal vector is continuously varying, and we closely follow the derivation of the integral equation as can be found in [Ladyzhenskaya 1963].

We introduce the fundamental singular solution of the Stokes system, or more exactly, the tensor formed by the solutions corresponding to concentrated forces directed along the various coordinate axes. Thus we consider that the vector field $\mathbf{u}^k(\mathbf{x}, \mathbf{y})$

and the scalar function $q^k(\mathbf{x},\mathbf{y})$ satisfy the following equations

$$\begin{aligned} \Delta\mathbf{u}^k(\mathbf{x},\mathbf{y}) - \mathbf{grad}\ q^k(\mathbf{x},\mathbf{y}) &= \delta(\mathbf{x}-\mathbf{y})\mathbf{e}^k, \\ \text{div}\ \mathbf{u}^k &= 0, \end{aligned} \tag{1.21}$$

where $k = 1, 2,\ \mathbf{e}^k = (\delta_{1k}, \delta_{2k})$ is a unit vector directed along the k^{th} coordinate axis, and $\delta(\mathbf{x}-\mathbf{y})$ is the Dirac delta function. All the differentiations are carried out with respect to the variable $\mathbf{x}$, and $\mathbf{y}$ plays the role of a parameter (the force is applied at $\mathbf{y}$). The system is supplemented by the requirement that $\mathbf{u}^k$ and q^k approach zero as $|x| \to \infty$. Physically, these equations may be interpreted as follows: the velocity at point $\mathbf{x}$ is induced by a two-dimensional unit point force in the $\mathbf{e}^k$-direction at point $\mathbf{y}$. This solution is also referred to as a flow due to a *Stokeslet.* Applying a Fourier transformation [Ladyzhenskaya 1963], we obtain

$$\begin{aligned} u_j^k(\mathbf{x},\mathbf{y}) &= -\frac{1}{4\pi}\left[\delta_{jk}\ln\frac{1}{|\mathbf{x}-\mathbf{y}|} + \frac{(x_j-y_j)(x_k-y_k)}{|\mathbf{x}-\mathbf{y}|^2}\right], \\ q^k(\mathbf{x},\mathbf{y}) &= -\frac{x_k-y_k}{2\pi|\mathbf{x}-\mathbf{y}|^2}. \end{aligned} \tag{1.22}$$

The functions (1.22) are also the solutions to the *adjoint system*

$$\begin{aligned} \Delta_y\mathbf{u}^k(\mathbf{x},\mathbf{y}) - \text{grad}_y\ q^k(\mathbf{x},\mathbf{y}) &= \delta(\mathbf{x}-\mathbf{y})\mathbf{e}^k, \\ \text{div}_y\ \mathbf{u}^k &= 0. \end{aligned} \tag{1.23}$$

By $(\ )_y$ we mean that the differentiation is carried out with respect to $\mathbf{y}$. Next, we derive the so-called *Green's formula* to the Stokes problem.

Theorem 1.2 (Green's Formula) *Let* $\mathbf{u}$ *and* $\mathbf{v}$ *be two arbitrary solenoidal vector fields, and let the scalar function* p, q *be "sufficiently" smooth. Then the following integral identity holds for those function*

$$\int_{\Omega}\left[\left(\Delta v_i - \frac{\partial p}{\partial x_i}\right) u_i - \left(\Delta u_i + \frac{\partial q}{\partial x_i}\right) v_i\right] \mathrm{d}\Omega = \int_{\Gamma} [T_{ij}(p,\ \mathbf{v}) u_i n_j - T_{ij}(-q,\ \mathbf{u}) v_i n_j]\ \mathrm{d}\Gamma, \tag{1.24}$$

where the stress tensor $T_{ij}(q,\ \mathbf{u})$ for this Newtonian fluid is equal to

$$T_{ij}(q,\ \mathbf{u}) = -q\delta_{ij} + \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{1.25}$$

The proof for the above identity is given as follows. Firstly, we take the divergence of $T_{ik}(q,\ \mathbf{u})v_i$ given by:

$$\begin{aligned} \frac{\partial}{\partial x_k}[T_{ik}(q,\ \mathbf{u})v_i] &= \frac{\partial}{\partial x_k}\left\{\left[-q\delta_{ik} + \left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right)\right] v_i\right\} \\ &= \frac{1}{2}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i}\right) + \left(\Delta u_i - \frac{\partial q}{\partial x_i}\right) v_i\,. \end{aligned} \tag{1.26}$$

Secondly, we integrate this equation over Ω, and we use the divergence, which yields:

$$\begin{aligned} \int_{\Omega}\left(\Delta u_i - \frac{\partial q}{\partial x_i}\right) v_i\, d\Omega &= -\frac{1}{2}\int_{\Omega}\left(\frac{\partial u_i}{\partial x_k} + \frac{\partial u_k}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_k} + \frac{\partial v_k}{\partial x_i}\right) d\Omega \\ &+ \int_{\Gamma} T_{ik}(q,\ \mathbf{u}) v_i n_k\, d\Gamma. \end{aligned} \tag{1.27}$$

By interchanging u_i and v_i in Equation (1.27) and introducing the arbitrary smooth function p together with q, from Equation (1.27) we can obtain the Green's formula corresponding to the Stokes problem, i.e., Equation (1.24).

We replace $\mathbf{u}$ and q in the Green's formula with the fundamental singular solutions $\mathbf{u}^k(\mathbf{x},\ \mathbf{y})$, $q^k(\mathbf{x},\ \mathbf{y})$ of (1.22) and we consider the function of $\mathbf{y}$ to be a solution of the adjoint system (1.23). Furthermore, we require that $\mathbf{v}$ and p satisfy the Stokes problem described in the previous section. So, we obtain the following Fredholm integral equation of the second kind [Kress 1989], that is written as:

$$v_k(\mathbf{x}) = \int_\Gamma T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \, \mathrm{d}\Gamma_y - \int_\Gamma T_{ij}(p,\ \mathbf{v}) u_i^k n_j \, \mathrm{d}\Gamma_y, \tag{1.28}$$

for *any* $\mathbf{x}\ \in\ \Omega$. Substituting the fundamental solution (1.22) into Equation (1.25), yields:

$$T_{ij}(-q^k,\ \mathbf{u}^k)_y = -\frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)}{\pi|\mathbf{x} - \mathbf{y}|^4}. \tag{1.29}$$

Note that $T_{ij}(p,\ \mathbf{v})n_j$ is given by the boundary condition (1.12), i.e., $T(p,\ \mathbf{v})\mathbf{n} = -\kappa(\mathbf{y})\mathbf{n} = \mathbf{b}$.

In the following section we use "$\mathbf{y}$" to denote the points that belong to the boundary Γ. Following [Ladyzhenskaya 1963], we define a single layer potential with surface tension $\mathbf{b}(\mathbf{y})$ as:

$$V_i(\mathbf{x},\ \psi) = -\int_\Gamma u_i^k(\mathbf{x},\ \mathbf{y}) b_k(\mathbf{y}) \, \mathrm{d}\Gamma_y. \tag{1.30}$$

The double layer potential with velocity $\mathbf{v}(\mathbf{y})$ is defined as:

$$\begin{aligned} W_k(\mathbf{x},\ \varphi) &= \int_\Gamma T_{ij}(-q^k,\ \mathbf{u}^k)_y\ v_i(\mathbf{y}) n_j(\mathbf{y}) \, \mathrm{d}\Gamma_y \\ &= \int_\Gamma K_{kj}(\mathbf{x},\ \mathbf{y}) v_j(\mathbf{y}) \, \mathrm{d}\Gamma_y, \end{aligned} \tag{1.31}$$

where

$$K_{ij}(\mathbf{x},\ \mathbf{y}) = -\frac{(x_i - y_i)(x_j - y_j)(x_k - y_k)}{\pi|\mathbf{x} - \mathbf{y}|^4} n_k(\mathbf{y}). \tag{1.32}$$

From (1.28), we obtain for $\mathbf{x}\ \in\ \Omega$:

$$v_k(\mathbf{x}) = W_k(\mathbf{x},\ \mathbf{v}) + V_k(\mathbf{x},\ \mathbf{b}). \tag{1.33}$$

Thus, the boundary integrals in (1.28) are called hydrodynamical potentials of single- and double-layer functions.

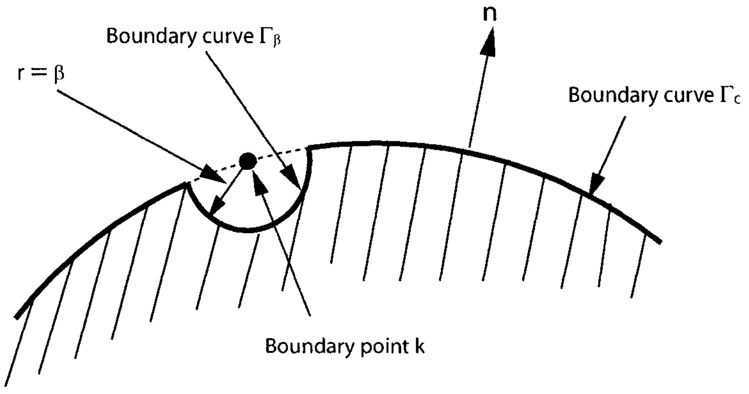


Figure 1.1 : Boundary point for two-dimensional case, augmented by a small semicircle.

When we let point $\mathbf{x}$ in Equation (1.28) approach the boundary, and assuming that the boundary is "smooth", the velocity for point $\mathbf{x}$ is given by:

$$\frac{1}{2}v_k(\mathbf{x}) = W_k(\mathbf{x},\ \mathbf{v}) + V_k(\mathbf{x},\ \mathbf{b}). \tag{1.34}$$

The factor 1/2 is caused by the "jump" condition of the double layer potential $W_k(\mathbf{x},\ \mathbf{v})$. Now, we give the proof of Equation (1.34). When $\mathbf{x}$ is a boundary point, we use a small circle of radius β to surround this point situated in Ω, as shown in Figure 1.1. Γ_β denotes the intersection curve of Γ with the small piece of circle, and Γ_c the rest of the boundary. Consider the integral Equation (1.28) in this case, we have

$$\begin{aligned} v_k(\mathbf{x}) &= \int_{\Gamma_c+\Gamma_\beta} T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \,\mathrm{d}\Gamma_y - \int_{\Gamma_c+\Gamma_\beta} T_{ij}(p,\ \mathbf{v}) u_i^k n_j \,\mathrm{d}\Gamma_y \\ &= \int_{\Gamma_c} T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \,\mathrm{d}\Gamma_y - \int_{\Gamma_c} T_{ij}(p,\ \mathbf{v}) u_i^k n_j \,\mathrm{d}\Gamma_y \\ &+ \int_{\Gamma_\beta} T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \,\mathrm{d}\Gamma_y - \int_{\Gamma_\beta} T_{ij}(p,\ \mathbf{v}) u_i^k n_j \,\mathrm{d}\Gamma_y, \end{aligned} \tag{1.35}$$

and take the limit for $\beta \to 0$. It is clear, from Figure (1.1), that

$$\lim_{\beta\to 0}\int_{\Gamma_c} T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \,\mathrm{d}\Gamma_y = \int_{\Gamma} T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \,\mathrm{d}\Gamma_y, \tag{1.36}$$

and

$$\lim_{\beta\to 0}\int_{\Gamma_c} T_{ij}(p,\ \mathbf{v}) u_i^k n_j \,\mathrm{d}\Gamma_y = \int_{\Gamma} T_{ij}(p,\ \mathbf{v}) u_i^k n_j \,\mathrm{d}\Gamma_y. \tag{1.37}$$

For calculating the limit for boundary Γ_β, we use the following formulation:

$$\begin{aligned}
\lim_{\beta\to 0}\int_{\Gamma_\beta} T_{ij}(-q^k,\ \mathbf{u}^k)_y v_i n_j \,\mathrm{d}\Gamma_y &= \lim_{\beta\to 0}\int_{\Gamma_\beta} -\frac{(x_i-y_i)(x_j-y_j)(x_k-y_k)}{\pi|\mathbf{x}-\mathbf{y}|^4} v_i n_j \,\mathrm{d}\Gamma_y \\
&= -v_k(\mathbf{x})\frac{1}{\pi}\lim_{\beta\to 0}\int_{\Gamma_\beta}\left(-\frac{\cos^2\theta}{\beta}\right)\mathrm{d}\Gamma_y \\
&= v_k(\mathbf{x})\frac{1}{\pi}\int_0^{\pi}\cos^2\theta \,\mathrm{d}\theta \\
&= \frac{1}{2}v_k(\mathbf{x}),
\end{aligned} \tag{1.38}$$

and

$$\begin{aligned}
\lim_{\beta\to 0}\int_{\Gamma_\beta} T_{ij}(p,\ \mathbf{v}) u_i^k n_j \,\mathrm{d}\Gamma_y &= \lim_{\beta\to 0}\int_{\Gamma_\beta} b_i u_i^k \,\mathrm{d}\Gamma_y \\
&= \lim_{\beta\to 0}\int_{\Gamma_\beta} -\frac{1}{4\pi}\left[\delta_{jk}\ln\frac{1}{|\mathbf{x}-\mathbf{y}|} + \frac{(x_i-y_i)(x_j-y_j)}{|\mathbf{x}-\mathbf{y}|^2}\right] b_i \,\mathrm{d}\Gamma_y \\
&= C_1 \lim_{\beta\to 0}(\beta\ln\beta) + C_2\lim_{\beta\to 0}\beta \\
&= 0,
\end{aligned} \tag{1.39}$$

where C_1 and C_2 are arbitrary constants. Thus, combining Equations (1.34) , (1.35), (1.36), (1.37), (1.38), we can obtain the expression of the velocity for $\mathbf{x}$ over the boundary Γ, i.e., Equation (1.34).

In general, by substituting Equation (1.22), (1.29), and (1.12) into Equation (1.33) or (1.34), we have

$$c_{ij}v_j(\mathbf{x}) + \int_\Gamma q_{ij}(\mathbf{x},\ \mathbf{y})v_j(\mathbf{y})\,\mathrm{d}\Gamma_y = \int_\Gamma u_{ij}(\mathbf{x},\ \mathbf{y})b_j(\mathbf{y})\,\mathrm{d}\Gamma, \tag{1.40}$$

where c_{ij}, q_{ij} and u_{ij} are respectively equal to:

$$\begin{aligned} c_{ij} &= \begin{cases} \delta_{ij} & \mathbf{x} \in \Omega, \\ \frac{1}{2}\delta_{ij} & \mathbf{x} \in \Gamma, \\ 0 & \mathbf{x} \in \Omega', \end{cases} \\ q_{ij}(\mathbf{x},\ \mathbf{y}) &= \frac{r_i r_j}{\pi R^4} r_k n_k, \\ u_{ij}(\mathbf{x},\ \mathbf{y}) &= \frac{1}{4\pi}\left[-\delta_{ij}\ln R + \frac{r_i r_j}{R^2}\right], \end{aligned} \tag{1.41}$$

where $r_i = x_i - y_i$, $R = \sqrt{r_1^2 + r_2^2} = |\mathbf{x} - \mathbf{y}|$, and the Ω' is the complementary region of $\Omega \bigcup \Gamma$. Right-hand side integral of Equation (1.40) is the so-called *single-layer hydrodynamic potential*, and the second integral is called the *double-layer hydrodynamic potential.* Therefore Equation (1.40) is usually referred to as the integral equation of single- and double-layer potentials, and will be denoted as the *SDLP-formulation.*

Here, we denote the single layer potential operator with a continuous varying density $\varphi(\mathbf{y})$ by:

$$(\mathcal{G}\varphi)_i = \int_\Gamma u_{ij}(\mathbf{x},\ \mathbf{y})\varphi_j(\mathbf{y})\,\mathrm{d}\Gamma_y, \tag{1.42}$$

and the double layer potential operator with density $\psi(\mathbf{y})$ defined as:

$$(\mathcal{H}\psi)_i = \int_\Gamma q_{ij}(\mathbf{x},\ \mathbf{y})\psi_j(\mathbf{y})\,\mathrm{d}\Gamma_y. \tag{1.43}$$

The kernel of the single layer potential (1.42) is weakly singular so that $\mathcal{G}$ is a compact operator [Kress 1989]. It can be shown that the kernel of the double layer potential (1.43) is continuous and bounded, thus the operator $\mathcal{H}$ is compact too. This implies that the Fredholm-Riesz-Schauder theory for compact operators can be applied to these two potentials. The essence of this theory is that the Fredholm equation of the second kind shares several properties with ordinary square matrices. Hence, the functional operators might be viewed as the matrix operations used in linear algebra. Thus, the SDLP-formulation (1.40) can be represented in terms of these potentials as follows:

$$\big(\lambda\mathcal{J} \;+\; \mathcal{H}\big)\mathbf{v} = \mathcal{G}\mathbf{b}, \tag{1.44}$$

where $\mathcal{J}$ is the identity operator, and $\lambda = c_{11} = c_{22}$. From now on, we assume that point $\mathbf{x}$ is on the boundary Γ, i.e., $\lambda = \frac{1}{2}$, hence we have to solve the Fredholm integral equation of the second kind. In the next section, we give some of mathematical definitions and theorems which will be applied further on.

Firstly, we give two definitions, the functional *inner product* that is defined in the usual way, that is:

$$\langle \phi, \psi \rangle =: \int_\Gamma \phi_i \psi_i \,\mathrm{d}\Gamma, \tag{1.45}$$

and the adjoint $\mathcal{A}^*$ of an operator $\mathcal{A}$ defined as:

$$\langle \phi, \mathcal{A}\psi \rangle =: \langle \mathcal{A}^*\phi, \psi \rangle, \tag{1.46}$$

here the adjoint operator of (1.45) and (1.46) can be found by transposing the indices j and k and swapping the argument $\mathbf{x}$ and $\mathbf{y}$ of the coefficients in (1.41). In particular, it follows that

$$\Big(\mathcal{G}^*\phi\Big)_i = \Big(\mathcal{G}\phi\Big)_i \,; \qquad \Big(\mathcal{H}^*\psi\Big)_i = -n_k(\mathbf{x}) \int_\Gamma \frac{r_i r_j r_k}{\pi R^4} \psi_j \,\mathrm{d}\Gamma_y, \tag{1.47}$$

where $\mathcal{G}$ is a *self-adjoint* (symmetric) linear operator.

Theorem 1.3 (First Fredholm Theorem) *Let $\mathcal{A}$ be a compact operator, then the null spaces of the operator $\mathcal{J} - \mathcal{A}$ and $\mathcal{J} - \mathcal{A}^*$ have the same finite dimension.*

The proof of this theorem can be found in [Kress 1989].

Theorem 1.4 (Second Fredholm Theorem) *Let $\mathcal{A}$ be a compact operator, the nonhomogeneous equation*

$$\phi - \mathcal{A}\phi = \mathbf{f},$$

has one solution if and only if the condition $\langle \mathbf{f}, \psi \rangle = 0$ is satisfied for all the solutions of the homogeneous adjoint equation $\psi - \mathcal{A}^*\psi = 0$

Again the proof of this theorem is outlined in [Kress 1989], and is also called Fredholm alternative. Note that the above theorem also holds when $\mathcal{A}$ and $\mathcal{A}^*$ are interchanged.

Let the three linearly independent rigid-body motions be denoted by $\boldsymbol{\varphi}^k$, where

$$\boldsymbol{\varphi}^k(\boldsymbol{x}) = (\delta_{k1},\ \delta_{k2}), \qquad (k = 1,\ 2), \qquad \text{and} \qquad \boldsymbol{\varphi}^3(\boldsymbol{x}) = (x_2,\ -x_1). \qquad (1.48)$$

For the solution to the homogeneous equation (1.44) the following theorem holds.

Theorem 1.5 *The functions $\boldsymbol{\varphi}^{\boldsymbol{k}}(\boldsymbol{x})$ span the null spaces of the operator $(\frac{1}{2}\mathcal{J} + \mathcal{H})$*

From Theorem 1.1 we observed that the functions $\boldsymbol{\varphi}^{\boldsymbol{k}}$ were the solutions of the Stokes equations with zero tensions on the boundary, i.e., the body force $\mathbf{b} = \mathbf{0}$. By substituting this equality into Equation (1.44), it is clear that these functions satisfy the homogeneous part of this equation. In order to prove that these functions are the basis for the null space, it is sufficient to show the validity for the adjoint $(\frac{1}{2}\mathcal{J} + \mathcal{H}^*)$, since the First Fredholm theorem 1.3 states that the null spaces of both operators have the same dimension. Therefore, we firstly suppose that $\boldsymbol{\phi}$ satisfies

$$(\frac{1}{2}\mathcal{J} + \mathcal{H}^*)\boldsymbol{\phi} = 0. \qquad (1.49)$$

Then, we define the single-layer potential velocity field $\hat{\mathbf{v}}$ associated with ϕ by

$$\hat{\mathbf{v}}(\mathbf{x}) = \mathcal{G}\phi, \tag{1.50}$$

and the single-layer potential that is associated with the pressure $\hat{p}$ of this flow as

$$\hat{p}(\mathbf{x}) = -\int_\Gamma q^k(\mathbf{x},\mathbf{y})\phi_k(\mathbf{y})\,\mathrm{d}\Gamma_y, \tag{1.51}$$

where $\mathbf{x} \in \Omega$. By using these definitions, we obtain the following stress tensor associated with the pressure and velocity [Pozrikidis 1992]

$$T_{ij}(\hat{p},\hat{\mathbf{v}}) = -\int_\Gamma \frac{r_i r_j r_k \phi_k}{\pi R^4}\,\mathrm{d}\Gamma_y. \tag{1.52}$$

Taking the limit from the inside of fluid towards a point $\mathbf{x}$ of boundary curve (this limit can be found in a similar way as the coefficient c_{ij} deduced previously), we obtain the tension in the normal direction, given by:

$$T_{ij}(\hat{p},\hat{\mathbf{v}})n_j(\mathbf{x}) = (\frac{1}{2}\mathcal{J} + \mathcal{H}^*)\phi = 0. \tag{1.53}$$

Hence, this zero boundary tension implies that ϕ must follow a rigid-body motion. It is clear that a rigid-body force will not change the actual flow field of the fluid domain.

Now, we also can prove that any solution of Equation (1.44) satisfies the incompressibility condition.

Theorem 1.6 *Let* $\mathbf{v}$ *be a solution of Equation (1.44), then this function satisfies*

$$\langle \mathbf{v}, \mathbf{n} \rangle = 0.$$

In order to prove this theorem, we will use the following identities [Yang & Zhao 2002] that apply to the potentials along the outer normal direction,

$$\Big(\mathcal{H}^*\mathbf{n}\Big)_i = \Big(-n_j(\mathbf{x})\mathcal{H}\mathbf{e}\Big)_i = \begin{cases} n_i & \mathbf{x} \textit{ inside } \Gamma, \\ \frac{1}{2}n_i & \mathbf{x} \textit{ on } \Gamma, \\ 0 & \mathbf{x} \textit{ outside } \Gamma. \end{cases} \tag{1.54}$$

where $\mathbf{e} = (1,\,1,\,\cdots,\,1)^T$. Hence, the outer normal vector $\mathbf{n}$ is an eigenfunction of the adjoint operator $\mathcal{H}^*$. Furthermore, for the single-layer potential, we have:

$$\Big(\mathcal{G}\mathbf{n}\Big)_i = \int_\Gamma u_{ij}n_j\,\mathrm{d}\Gamma = \int_\Omega \mathrm{div}\mathbf{u}^i\,\mathrm{d}\Omega = 0, \tag{1.55}$$

which is derived from the application of the divergence theorem and by noting that $\mathbf{u}^i$ corresponds to the expression given by Equation (1.41). Let $\mathbf{x} \in \Gamma$, taking the inner product of Equation (1.44) with respect to outer normal, we obtain

$$\begin{aligned} \big\langle \mathbf{n}, (\frac{1}{2}\mathcal{J} + \mathcal{H})\mathbf{v}\big\rangle &= \big\langle (\frac{1}{2}\mathcal{J} + \mathcal{H}^*)\mathbf{n}, \mathbf{v}\big\rangle \\ &= \big\langle \mathbf{n}, \mathbf{v}\big\rangle \\ &= \big\langle \mathbf{n}, \mathcal{G}\mathbf{b}\big\rangle = \big\langle \mathcal{G}^*\mathbf{n}, \mathbf{b}\big\rangle = 0, \end{aligned} \tag{1.56}$$

where we used the self-adjointness of $\mathcal{G}$ and the above mentioned identities. For $\mathbf{x} \in \Omega$, the proof is straightforward.

There are two main approaches to make (1.40) uniquely solvable. The first method is to add three additional variables that describe those rigid-body movements in order to achieve a full rank system, employed by Hsiao et al [Hsiao, Kopp & Wendland 1984] and van de Vorst et al [van de Vorst & Mattheij 1992]. The translation freedom is prescribed by considering the flow stationary at a reference point inside the fluid. This point is normally taken equal to the center of mass. With regard to this point, the boundary flow field is computed. Substitution of the reference point as the $\mathbf{x}$-argument into the integral equation yields two integral constraints. The third integral constraint is derived by taking the inner-product of the velocity field and the tangential vector along the boundary curve equal to zero.

The other method consists of "removing" the eigenvalue of the double layer integral operator that causes this null space, i.e. $-1/2$, and constructing a "deflated" operator. The advantage of this deflation is that the number of unknowns does not change, and we do not have to seek extra integral constraints. It is a purely mathematical approach. This approach can be achieved by using the *Wieland's deflation* technique [Wilkinson 1965], which is extensively discussed by Kim and Karrila [Kim & Karrila 1991] and Pozrikidis [Pozrikidis 1992] respectively. In short, this approach "replaces" the deflated eigenvalue by zero without effecting the other eigenvalues, and achieve a full rank system so that the integral equation can be solved uniquely. In this project, we adopted this "deflated" method. Here, we just use this approach to construct a deflated operator, and the detail about this method can be found in above mentioned reference books.

Before presenting the "deflated" equation, we require the rigid-body motion φ^k to be orthonormal, this can be done by applying the Gram-Schmidt method, that is:

$$\begin{aligned} \hat{\varphi}^k &= \frac{\varphi^k}{\sqrt{L}}, \quad (k = 1,\ 2), \\ \hat{\varphi}^3 &= \frac{(x_2 - x_2')\varphi^1 - (x_1 - x_1')\varphi^2}{\sqrt{I}}, \end{aligned} \tag{1.57}$$

where

$$\begin{aligned} L &= \int_\Gamma \mathrm{d}\Gamma, \\ x_i' &= \frac{1}{L}\int_\Gamma x_i \,\mathrm{d}\Gamma, \\ I &= \int_\Gamma \left[(x_1 - x_1')^2 + (x_2 - x_2')^2\right] \mathrm{d}\Gamma. \end{aligned} \tag{1.58}$$

Here I is the moment of inertia about the center of torque x' of the fluid domain.

Using the function (1.57), we define three projection operators $\boldsymbol{p}^k$ as follows:

$$\boldsymbol{p}^k = \hat{\varphi}^k \langle \,\cdot, \hat{\varphi}^k \rangle. \tag{1.59}$$

By using the "deflated" method [Wilkinson 1965], we can replace the operator $\mathcal{H}$ in Equation (1.44) by

$$\mathcal{H} + \boldsymbol{p} \; := \; \mathcal{H} + \boldsymbol{p}^1 + \boldsymbol{p}^2 + \boldsymbol{p}^3,$$

and we obtain

$$\Big(\frac{1}{2}\mathcal{J} + \mathcal{H} + \boldsymbol{p}\Big)\mathbf{v} = \mathcal{G}\mathbf{b}. \tag{1.60}$$

Now, there exists a unique solution $\mathbf{v}$ of the above equation since the operator$(\frac{1}{2}\mathcal{J} + \mathcal{H} + \boldsymbol{p})$ is bijective [Kress 1989]. Moreover, this solution also satisfies the original Equation (1.44). This means that $\boldsymbol{p}\mathbf{v} = 0$, which is fulfilled due to

$$\begin{aligned}
\langle \mathbf{v}, \hat{\varphi}^k \rangle &= \langle \mathbf{v}, (\frac{1}{2}\mathcal{J} + \mathcal{H}^*)\hat{\varphi}^k \rangle + \langle \mathbf{v}, \hat{\varphi}^k \rangle \\
&= \langle (\frac{1}{2}\mathcal{J} + \mathcal{H})\mathbf{v}, \hat{\varphi}^k \rangle + \langle \mathbf{v}, \hat{\varphi}^i \rangle \langle \hat{\varphi}^i, \hat{\varphi}^k \rangle \\
&= \langle (\frac{1}{2}\mathcal{J} + \mathcal{H} + \boldsymbol{p})\mathbf{v}, \hat{\varphi}^k \rangle \\
&= \langle \mathcal{G}\mathbf{b}, \hat{\varphi}^k \rangle = 0,
\end{aligned}$$

where the equality follows from the solvability condition of the original equation, i.e., Fredholm's second theorem 1.4. In terms of ordinary integrals, Equation (1.60) can be expressed by

$$c_{ij}v_j(\mathbf{x}) + \int_\Gamma q_{ij}(\mathbf{x},\ \mathbf{y})v_j \,\mathrm{d}\Gamma_y + \hat{\varphi}_i^k(\mathbf{x}) \int_\Gamma \hat{\varphi}_j^k v_j \,\mathrm{d}\Gamma_y = \int_\Gamma u_{ij}(\mathbf{x},\ \mathbf{y})b_j \,\mathrm{d}\Gamma_y. \tag{1.61}$$

The above integral formulation can be solved in the case of a simply connected domain.

1.5 Equations for a Multiply Connected Domain

Let this kind of domain be bounded externally by Γ_0 and internally by $\Gamma_1, \ldots, \Gamma_M$. By Γ, we denote the complete boundary. Moreover, the outer boundary Γ_0 is considered in counter-clockwise direction, and all the inner boundaries Γ_m are passed through in clockwise direction. Hence, the normal vector is always pointing to the outward of the fluid.

If we formally apply Equation (1.44) on to this multiply connected domain, we find

$$\begin{aligned} \Big(\frac{1}{2}\mathcal{J} + \mathcal{H}_0\Big)\mathbf{v}^0 - \sum_{j=1}^{M}\mathcal{H}_j\mathbf{v}^j &= \sum_{j=0}^{M}\mathcal{G}_j\mathbf{b}^j, \qquad \mathbf{x} \in \Gamma_0\,, \\ \mathcal{H}_0\mathbf{v}^0 - \sum_{j=1}^{M}\mathcal{H}_j\mathbf{v}^j + \frac{1}{2}\mathcal{J}\mathbf{v}^m &= \sum_{j=0}^{M}\mathcal{G}_j\mathbf{b}^j, \qquad \mathbf{x} \in \Gamma_m\,, \end{aligned} \tag{1.62}$$

where the subscript j of both potentials $\mathcal{H}_j$, $\mathcal{G}_j$ denote that the integration is subjected to the boundary Γ_j which is passed through in a counter-clockwise direction (this explains the occurrence of the minus sign for the operator $\mathcal{H}_j$ for the inner boundaries). Here $\mathbf{v}^m$, $\mathbf{b}^m$ are the velocity and tension functions respectively on the boundary Γ_m. It is easy to show that the rigid-body movement function (1.57) are a basis for the null space of the homogeneous part of Equation (1.62). By using those functions, the above double-layer potential also can be "deflated" in a similar way to what was done for the simply connected domain.

Let $\varphi^{mk}(\mathbf{v})$ be the k^{th} rigid-body motion of the hole enclosed by Γ_m when $\mathbf{v} \in \Gamma_m$, otherwise this function is taken equal to zero. The projection operator $\boldsymbol{p}_m$ is defined as:

$$\boldsymbol{p}_m := \sum_{k=1}^{3} \hat{\varphi}^{mk}\langle \cdot\ ,\ \hat{\varphi}^{mk}\rangle_m\,, \tag{1.63}$$

where the inner product is performed on boundary Γ_m. Using these projection operators, one obtains the "deflated" form of Equation (1.62) given by:

$$\begin{aligned} \Big(\frac{1}{2}\mathcal{J} + \mathcal{H}_0 + \boldsymbol{p}_0\Big)\mathbf{v}^0 - \sum_{j=1}^{M}\mathcal{H}_j\mathbf{v}^j &= \sum_{j=0}^{M}\mathcal{G}_j\mathbf{b}^j, \qquad \mathbf{x}\in\Gamma_0\,, \\ \mathcal{H}_0\mathbf{v}^0 - \sum_{j=1}^{M}\mathcal{H}_j\mathbf{v}^j + \Big(\frac{1}{2}\mathcal{J} + \boldsymbol{p}_m\Big)\mathbf{v}^m &= \sum_{j=0}^{M}\mathcal{G}_j\mathbf{b}^j\,, \qquad \mathbf{x}\in\Gamma_m\,, \end{aligned} \tag{1.64}$$

or as a similar form of (1.61)

$$\begin{aligned} c_{ij}v_j^m(\mathbf{x}) \;+\; &\sum_{k=0}^{M}\int_{\Gamma_k} q_{ij}(\mathbf{x},\ \mathbf{y})v_j^k\,\mathrm{d}\Gamma_y + \hat{\varphi}_i^{ml}(\mathbf{x})\int_{\Gamma_m}\hat{\varphi}_j^{ml}v_j^m\,\mathrm{d}\Gamma_y \\ =\; &\sum_{k=0}^{M}\int_{\Gamma_k} u_{ij}(\mathbf{x},\ \mathbf{y})b_j^k\,\mathrm{d}\Gamma_y, \qquad \mathbf{x}\in\Gamma_m, \text{and } m\in(0,M). \end{aligned} \tag{1.65}$$

The above integral equation so far cannot be used to describe the shrinkage or expansion of the inside holes in the fluid domain. This can be illustrated by a circular fluid disk with a circular hole centered at the origin. This problem can be solved analytically [van de Vorst 1993]. However, when we put this particular shape into the above equation, the null solution is obtained because the right hand side of the equation is equal to zero. This results from the fact that the curvature is constant and then can be taken to the outside of the operator $\mathcal{G}$, by using Equation (1.55).

The shrinkage or expansion of a hole can be modeled as a *point sink* or a *point source* respectively located inside the interior of such a hole. For the introduction to the usage of such singularity points to model such a flow behavior, we refer to any fluid dynamics book. The reason that Equations (1.64) or (1.65) cannot be used to describe this behavior is due to the fact that the outer normal is an eigenvalue of $\mathcal{H}^*$ (the adjoint double-layer potential) [Pozrikidis 1992]. The method of deflating the boundary normal is described by both Kim and Karrila [Kim & Karrila 1991] and Pozrikidis [Pozrikidis 1992]. However, the approach that can be used for fluid evolution was derived by van de Vorst [van de Vorst 1994]. Here, we just briefly outline the further deflation of Equation (1.65).

In a similar way as the integral formulation (1.40) has been obtained from the fundamental solution which represents a point force in an infinite two dimensional fluid, an integral equation can be obtained that represents a point source at a particular place. Again, we consider a simply connected domain bounded by Γ. The fundamental solution that represents a sources, say $\hat{\mathbf{u}}$ and $\hat{q}$ satisfy the following equations:

$$\begin{aligned} \triangle\hat{\mathbf{u}} - \mathbf{grad}\ \hat{q} &= 0, \\ \mathbf{div}\ \hat{\mathbf{u}} &= -\delta(\mathbf{x}, \mathbf{y}). \end{aligned} \tag{1.66}$$

Physically, Equation (1.66) can be interpreted as the velocity at $\mathbf{x}$ introduced by a point sink with unit force concentrated at point $\mathbf{y}$ (the point $\mathbf{y}$ is called a source point when the above Dirac delta function has a positive sign). This fundamental solution can be deduced after applying a cylindrical coordinate transformation given by:

$$\begin{aligned} \hat{u}_i(\mathbf{x}, \mathbf{y}) &= -\frac{x_i - y_i}{2\pi|\mathbf{x} - \mathbf{y}|^2}, \\ \hat{q}(\mathbf{x}, \mathbf{y}) &= 0. \end{aligned} \tag{1.67}$$

The functions $\mathbf{u}$ and q in Green's formula (1.24) are replaced by the above fundamental solution $\hat{\mathbf{u}}(\mathbf{x}, \mathbf{y}), \hat{q}(\mathbf{x}, \mathbf{y})$ which is also a solution of the adjoint of problem (1.66) when we consider as the function of $\mathbf{y}$. Substituting this fundamental solution into the adjoint stress tensor (1.25) yields

$$T_{ij}(-\hat{q},\ \hat{\mathbf{u}})_y = T_{ij}(0,\ \hat{\mathbf{u}})_y = -\frac{2(x_i - y_i)(x_j - y_j)}{\pi|\mathbf{x} - \mathbf{y}|^4} + \frac{\delta_{ij}}{\pi|\mathbf{x} - \mathbf{y}|^2}. \tag{1.68}$$

Requiring that $\mathbf{v}, p$ satisfy the Stokes problem and considering $\mathbf{x}$ to be inside the fluid, we obtain:

$$\int_{\Gamma} \hat{q}_j(\mathbf{x},\mathbf{y})v_j(\mathbf{y})\,\mathrm{d}\Gamma_y = \int_{\Gamma} \hat{u}_j(\mathbf{x},\mathbf{y})b_j(\mathbf{y})\,\mathrm{d}\Gamma_y\,, \tag{1.69}$$

where

$$\hat{q}_j = \frac{1}{\pi}\left(\frac{2r_j r_i n_i}{R^4} - \frac{\delta_{ji} n_i}{R^2}\right); \qquad \hat{u}_j = -\frac{r_j}{2\pi R^2}. \tag{1.70}$$

Now, we consider the equations for multiply connected domains. Let $\boldsymbol{\alpha}^m$ be a point inside the area surrounded by the Γ_m $(m = 1, \ldots, M)$. Hence we obtain M equations which, after summation, reduce to the following integral formulation that models all sink contribution, thus:

$$\sum_{k=0}^{M} \int_{\Gamma_k} q_j(\mathbf{y})v_j(\mathbf{y})\,\mathrm{d}\Gamma_y = \sum_{k=0}^{M} \int_{\Gamma_k} u_j(\mathbf{y})b_j(\mathbf{y})\,\mathrm{d}\Gamma_y\,, \tag{1.71}$$

where

$$q_j = \sum_{m=1}^{M} \hat{q}_j(\boldsymbol{\alpha}^m,\mathbf{y}); \qquad u_j = \sum_{m=1}^{M} \hat{u}_j(\boldsymbol{\alpha}^m,\mathbf{y}), \tag{1.72}$$

and $r_i^m = \alpha_i^m - y_i$, and $R^m = \sqrt{(r_1^m)^2 + (r_2^m)^2} = |\boldsymbol{\alpha}^m - \mathbf{y}|$ (see Figure 1.2). The above integral equation is applied as a constraint on the solution of Equation (1.65), i.e., we seek a velocity field $\mathbf{v}$ that satisfies both Equation (1.65) and Equation (1.71). We denote Equation (1.71) by using a similar operator notation as we employed for the potentials, i.e.,

$$\mathbf{h}_0\mathbf{v}^0 - \sum_{m=1}^{M} \mathbf{h}_m\mathbf{v}^m = \sum_{m=0}^{M} \mathbf{g}_m\mathbf{b}^m, \tag{1.73}$$

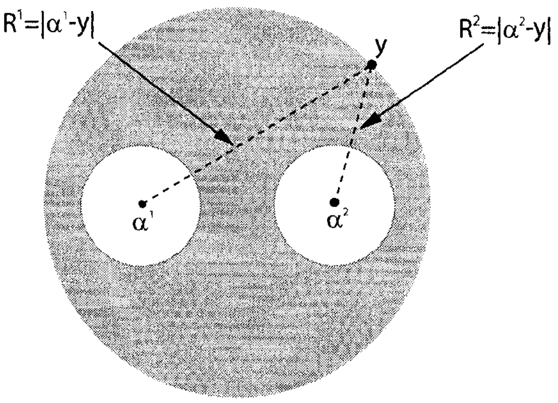


Figure 1.2 : Schema of R^m for expressing q_j and u_j.

where

$$\mathbf{h}_k\boldsymbol{\psi} = \int_{\Gamma_k} q_j\psi_j \,\mathrm{d}\Gamma_y \qquad \text{and} \qquad \mathbf{g}_k\boldsymbol{\phi} = \int_{\Gamma_k} u_j\phi_j \,\mathrm{d}\Gamma_y\,, \tag{1.74}$$

and Γ_k is considered in the counter-clockwise direction.

This equation can be applied in order to deflate Equation (1.64) with respect to the outer normal in the following way:

$$\begin{aligned}\Big(\frac{1}{2}\mathcal{J} + \mathcal{H}_0 + \boldsymbol{p}_0 + \mathbf{n}^0\mathbf{h}_0\Big)\mathbf{v}^0 - \sum_{j=1}^{M}\Big(\mathcal{H}_j \; & + \; \mathbf{n}^0\mathbf{h}_j\Big)\mathbf{v}^j \\ & = \sum_{j=0}^{M}\Big(\mathcal{G}_j + \mathbf{n}^0\mathbf{g}_j\Big)\mathbf{b}^j, \qquad \mathbf{x}\in\Gamma_0\,, \\ \Big(\mathcal{H}_0 + \mathbf{n}^m\mathbf{h}_0\Big)\mathbf{v}^0 - \sum_{j=1}^{M}\Big(\mathcal{H}_j + \mathbf{n}^m\mathbf{h}_j\Big)\mathbf{v}^j \; & + \; \Big(\frac{1}{2}\mathcal{J} + \boldsymbol{p}_m\Big)\mathbf{v}^m \\ & = \sum_{j=0}^{M}\Big(\mathcal{G}_j + \mathbf{n}^m\mathbf{g}_j\Big)\mathbf{b}^j, \qquad \mathbf{x}\in\Gamma_m\,.\end{aligned} \tag{1.75}$$

It is not difficult to prove that the solution of Equation (1.75) satisfies both Equations

(1.64) and (1.73) [van de Vorst 1993].

So far, the deflated integral Equation (1.75) with respect both rigid-body motion and outer normal of $\mathcal{H}^*$ can be expressed in terms of ordinary integral as:

$$\frac{1}{2}v_i^m(\mathbf{x}) + \sum_{k=0}^{M}\left(\int_{\Gamma_k} q_{ij}v_j^k \,\mathrm{d}\Gamma_y + n_i^m(\mathbf{x})\int_{\Gamma_k} q_j v_j^k \,\mathrm{d}\Gamma_y\right) + \hat{\varphi}_i^{ml}(\mathbf{x})\int_{\Gamma_m} \hat{\varphi}_j^{ml} v_j^m \,\mathrm{d}\Gamma_y$$
$$= \sum_{k=0}^{M}\left(\int_{\Gamma_k} u_{ij}b_j^k \,\mathrm{d}\Gamma_y + n_i^m(\mathbf{x})\int_{\Gamma_k} u_j b_j^k \,\mathrm{d}\Gamma_y\right), \quad \mathbf{x} \in \Gamma_m, \; m \in (0, M). \quad (1.76)$$

The above integral equation is applied to simulate a Stokes flow with vanishing holes inside the fluid.

Chapter 2

Numerical Method for Solving the Stokes Problem

In this chapter we discuss the numerical method for the boundary value problem, e.g., the Stokes equations at a fixed time. Since only the velocity field of this boundary is required, this problem can ideally be solved by using the BEM. Therefore, the boundary will be discretized into a sequence of elements and the velocity and surface tension are written in terms of their values at each nodal point. From the discretized form of the governing integral equation, a full rank system of linear algebraic equations is obtained for the unknown velocity field.

Then, we discuss the problem that occurs as a result of the numerical approximation of the curvature, especially when at a certain point of the boundary a cusp arise. In principle, the curvature at this point can be very large due to numerical cancellation. Finally, we propose an algorithm for node distribution (redistribution) based on equidistributing the arc length of the boundary.

2.1 Solution of the Fredholm Integral Equation of the Second Kind

We briefly consider numerical approximation methods, in particular, projection methods, for solving the Fredholm integral equation of the second kind. Consider the following equation

$$\lambda\psi - \mathcal{A}\psi = f, \tag{2.1}$$

where $\mathcal{A}$ is an operator from a Banach space, say $\mathbb{X}$, into the same space. The operator $\mathcal{A}$ is defined by

$$\mathcal{A}\psi(x) = \int_I K(x,y)\psi(y)\,\mathrm{d}y, \qquad (x \in I), \tag{2.2}$$

where the kernel $K(x, y)$ is weakly singular and the interval of integration I is closed. Hence the operator $\mathcal{A}$ is compact. Without loss of the generality, we assume λ in Equation (2.1) to be equal to 1.

The operator $\mathcal{A}$ will be approximated by a sequence of bounded finite rank operators $\mathcal{A}_N$. We achieve this by projecting the solution ψ into finite dimensional subspaces. Here, we just give a brief introduction to the numerical technique used to solve the Fredholm integral equation. The details about this can be found, among others, in the book [Kress 1989] for instance. We start by providing some general properties of such projection operators like their convergence and error estimation; next, we overview two special cases of projection methods that are of interest for us, i.e., the *collocation method* in which the projection will be prominent and *Galerkin's method* where the subspaces come to the front.

Let $\mathbb{X}_N \subset \mathbb{X}$ be a sequence subspaces which, from a practical point of view, are assumed to be finite dimensional. Moreover, let the finite rank operator $\mathcal{P}_n : \mathbb{X} \longrightarrow \mathbb{X}_N$ be a projection from $\mathbb{X}$ onto $\mathbb{X}_N$, i.e., the following properties are satisfied

$$\mathcal{P}_N = \mathcal{P}_N^2, \tag{2.3}$$

$$\mathcal{P}_N \psi_N = \psi_N, \qquad (\psi_N \subset \mathbb{X}_{\mathbb{N}}). \tag{2.4}$$

Next, we define the finite rank operator $\mathcal{A}_N := \mathcal{P}_N \mathcal{A} \mathcal{P}_N$. Replacing $\mathcal{A}$ by $\mathcal{A}_N$ in Equation (2.1), we have:

$$\psi - \mathcal{A}_N \psi = f. \tag{2.5}$$

Then multiplying Equation (2.5) by $\mathcal{P}_N$, and using the properties (2.3), (2.4) yields the following approximate equation in the finite dimensional subspace $\mathbb{X}_N$

$$\psi_N - \mathcal{A}_N \psi_N = f_N, \tag{2.6}$$

where $f_N = \mathcal{P}_N f$. Now, we give the condition that Equation (2.6) is uniquely solvable when $\mathcal{A}_N$ is sufficiently close to $\mathcal{A}$.

Theorem 2.1 *Let $\mathcal{A} : \mathbb{X} \longrightarrow \mathbb{X}$ be a compact operator such that $(\mathcal{J} - \mathcal{A})$ is injective, and let $\mathcal{P}_N$ be a projection from $\mathbb{X}$ onto finite dimensional subspace $\mathbb{X}_N$. Suppose for the finite rank projection $\mathcal{A}_N = \mathcal{P}_N \mathcal{A} \mathcal{P}_N$, there is the estimation*

$$\|(\mathcal{J} - \mathcal{A})^{-1}\| \|\mathcal{A}_N - \mathcal{A}\| < \frac{1}{4}. \tag{2.7}$$

Then for all $f \in \mathbb{X}$, the projection Equation (2.6) is uniquely solvable and if ψ is the solution of Equation (2.1), the following error estimation holds

$$\|\psi_N - \psi\| \leq \|(\mathcal{J} - \mathcal{A})^{-1}\| \Big(M \|\mathcal{A} - \mathcal{A}_N\| \|f\| + \|(\mathcal{J} - \mathcal{P}_N) f\| \Big), \tag{2.8}$$

where M is a positive constant depending on $\mathcal{A}$. This proof can be found in [Kress 1989].

Theorem 2.1 states that for a finite rank operator $\mathcal{A}_N$, sufficiently close to $\mathcal{A}$, the operator $\mathcal{J} - \mathcal{A}_N$ is invertible. The number 1/4 in the estimate given by Equation (2.7) is sufficient to prove this theorem. In practice however, the term $|(\mathcal{J} - \mathcal{A})^{-1}\| \|\mathcal{A}_N - \mathcal{A}\|$ will be much smaller since $\mathcal{A}_N$ is close to $\mathcal{A}$. The error estimate in Equation (2.8) denotes that the error $\|\psi_N - \psi\|$ corresponds to the error that is introduced by the projection. Hence, the error analysis of the approximate Equation (2.6) is reduced to the error estimation for $\mathcal{A} - \mathcal{P}_N\mathcal{A}\mathcal{P}_N$ and $f - \mathcal{P}_N f$.

Now we consider the *collocation method.* This method is usually referred to as the BEM. In this method, we chose a set of nodal points on boundary I, say $x^{i,N}(i = 1, 2, \cdots, N)$, where N is the total number of points. Let $\mathbb{X}_N$ be a sequence subspace, then the values of a function $\psi \in \mathbb{X}$ by $\psi_N \in \mathbb{X}_N$ is accomplished by:

$$\psi_N(x^{i,N}) := \psi(x^{i,N}), \qquad i = 1, 2, \cdots, N. \tag{2.9}$$

The equation defines a projection $\mathcal{P}_N$ from $\mathbb{X}$ onto $\mathbb{X}_N$, that is:

$$\mathcal{P}_N\psi = \psi_N = \sum_{i=1}^{N} \psi(x^{i,N})\phi^{i,N}, \tag{2.10}$$

where $\phi^{i,N}$ is an independent set in the space of continuous function $C(I)$. Hence we can obtain Equation (2.6), which applies to the set of nodal points $x^{i,N}$

$$\psi_N(x^{i,N}) - \mathcal{A}_N\psi_N(x^{i,N}) = f_N(x^{i,N}), \qquad i = 1, 2, \cdots, N, \tag{2.11}$$

where

$$\mathcal{A}_N\psi_N = \sum_{i,j=1}^{N} \psi_N(x^{i,N}) \int_I K(x^{i,N}, y)\phi^{i,N}(y)\,\mathrm{d}y. \tag{2.12}$$

Substituting the definition (2.9) for f_N and $\mathcal{A}_N\psi_N$ in Equation (2.12), we arrive at the collocation equation

$$\psi(x^{i,N}) - \mathcal{A}_N\psi(x^{i,N}) = f(x^{i,N}), \qquad i = 1, 2, \cdots, N. \tag{2.13}$$

Thus, the collocation method considers the integral equation just at discretization points; because of this the points $x^{i,N}$ are called *collocation points.* Note that the space $\mathbb{X}_N$ depends on the choice of $\phi^{i,N}$ and is related to the collocation points.

As stated in the estimate (2.8), the error of the approximate solution is related to the interpolation method. Usually, the truncated piecewise Lagrangian interpolation polynomials are taken for the basis functions $\phi^{i,N}$. Therefore, the discretization error is of order h^{q+1} where q is the polynomial degree and h is the maximum distance between two successive nodes.

In the case of *Galerkin's method* we require a Hilbert space with an inner product $\langle \cdot \, , \cdot \rangle$ (cf. Equation (1.45)) and a sequence of finite dimensional subsequences $\mathbb{X}_N$ in $\mathbb{X}$ with properties $\mathbb{X}_N \subset \mathbb{X}_{N+1}$ and the closure of $(\bigcup \mathbb{X}_N)$ is $\mathbb{X}$ [Yang & Zhao 2002]. Then we use the *orthogonal projection* $\mathcal{P}_N$ as a linear operator that can be used to project $\mathbb{X}$ onto $\mathbb{X}_N$. By using these properties to define the finite rank operator $\mathcal{A}_N = \mathcal{P}_N \mathcal{A} \mathcal{P}_N$, it can be seen that $\|\mathcal{A} - \mathcal{A}_N\| \to 0$ $(N \to \infty)$ [Weidmann 1980]. Hence, there is always a N_0 so that for $N > N_0$ the property (2.7) in Theorem 2.1 is satisfied. Now, we choose an orthonormal basis for the subspace $\mathbb{X}_N$ which is denoted again by $\phi^{i,N}$. The finite rank projection is written as:

$$\mathcal{P}_N \psi = \psi_N = \sum_{i=1}^{N} \langle \psi, \phi^{i,N} \rangle \phi^{i,N}. \tag{2.14}$$

By using the above definition, the projection Equation (2.6) can be written as:

$$\langle \psi_N, \phi^{i,N} \rangle - \langle \mathcal{A}_N \psi_N, \phi^{i,N} \rangle = \langle f_N, \phi^{i,N} \rangle, \qquad i = 1, 2, \cdots, N. \tag{2.15}$$

It can be proved that the above equation is the discrete Galerkin equation [Yang & Zhao 2002]. Hence, this method consists of finding the function $\psi_N \in \mathbb{X}_N$ that satisfies Equation (2.15). The advantage of Galerkin's method is that a symmetric kernel yields a symmetric matrix. However, the above equation also shows that Galerkin's method requires an additional integration for each matrix element as compared to the collocation method. Because of this additional integration, a collocation method is faster to compute and is, therefore, the most recommended for practical application. It appears, however, that the error analysis of Galerkin's method gives more satisfactory results as compared to the collocation method. In general, error estimations for collocation methods are based on the equivalence that can be established with certain Galerkin's methods [Arnold & Wendland 1983].

Finally, we consider the case that the integral Equation (2.1) is defined on a boundary curve Γ. When approximating this integral operator by a projection method, the error estimate in(2.8) still holds. However, an extra error occurs when replacing the boundary curve by a polygonal approximation (the exact analytical description of this boundary is in general not available).

2.2 The Implementation of the BEM for the Stokes Problem

The implementation of the BEM that is applied for discretizing the integral formulation for the Stokes problem presented in Sections 1.4 and 1.5 will be outlined. In particular, we consider the discretization of those integral equations in the case of a multiply connected domain. It is clear that the implementation in the case of a simply connected domain is straightforward by ignoring the sink contribution part. Books that describe the implementation of the BEM are among others [Brebbia & Walker 1980], [Brebbia & Dominguez 1989], and [Jawaon & Symm 1991].

If there are more than one boundary for a specified domain, then this kind of domain is referred to a multiply connected domain; otherwise it is simply connected domain. Actually, the cross-section of a microstructured optical fiber is a typical example of a multiply connected domain, while a single ellipse is a typical simply connected domain (see Figure 2.1).

Each separate curve Γ_m of the boundary of the multiply connected domain is divided into several elements which are defined by a set of nodal points. Let L_m to be the number of elements of this particular curve. A specific element is referred to with a superscript 'j' like Γ_m^j. The exact boundary is replaced by a polygon,

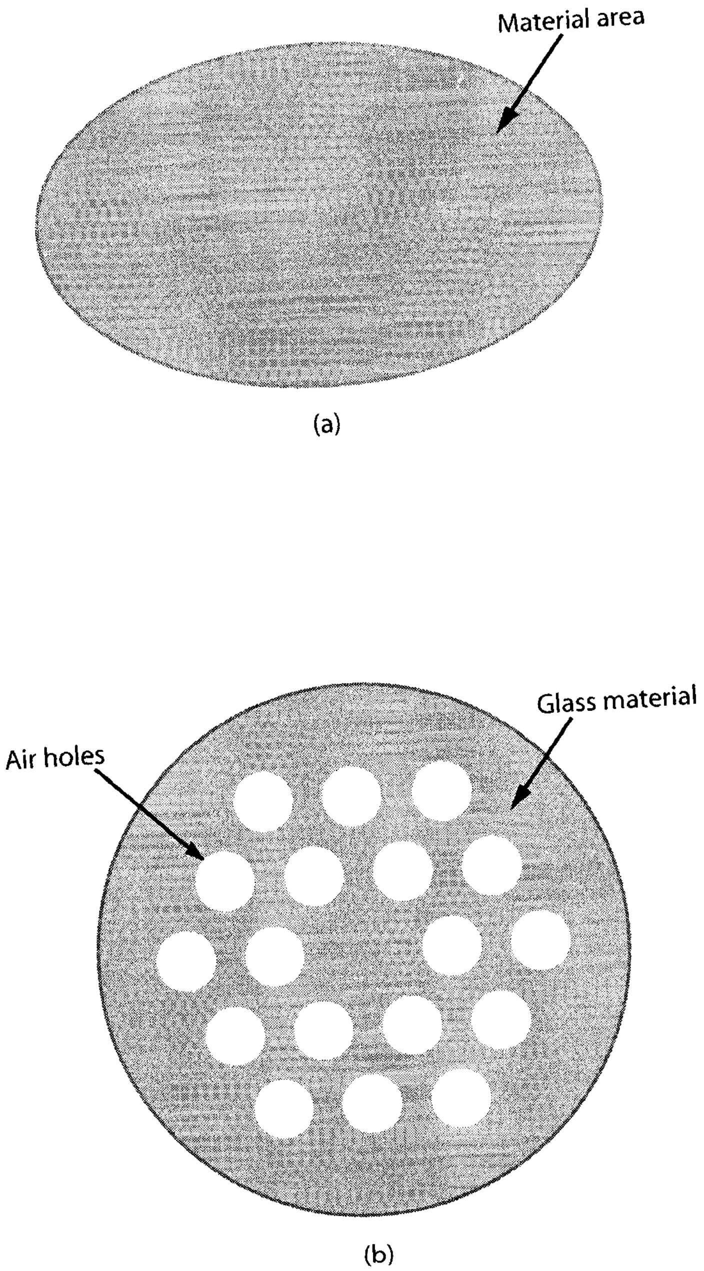


Figure 2.1 : Examples of simply connected and multiply connected domains:
(a) example of a simply connected domain,
(b)example of a multiply connected domain.

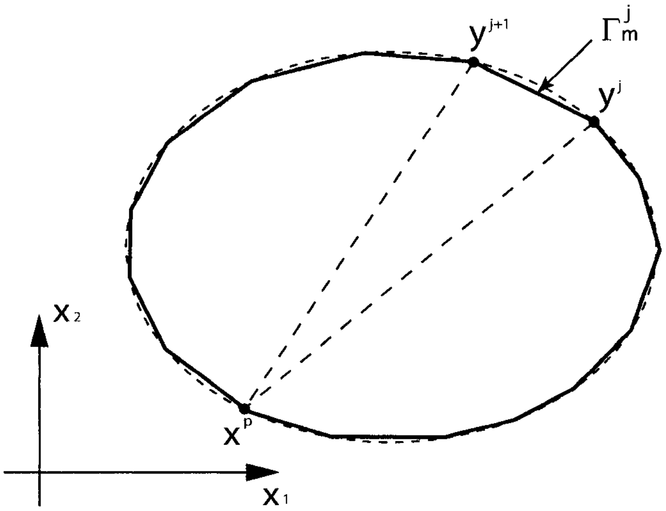


Figure 2.2 : The scheme of discretized boundary.

which consists of the union of all these elements (see Figure 2.2). The number of nodal points depends on the degree of polygonal approximation used. We apply the standard linear or quadratic Lagrangian polynomial approximations. Let N be the total number of collocation points, denoted by $\mathbf{x}^{\mathbf{i}}$, summed over all the boundary curves of the domain. The boundary velocity and tension are written in terms of their values at each nodal point.

Now, we define two functions, $\hat{\mathbf{v}}^{\mathbf{m}}$ and $\hat{\mathbf{b}}^{\mathbf{m}}$, which apply at a typical element "Γ_m^j"

$$\begin{aligned} \hat{\mathbf{v}}^{\mathbf{m}} &= \mathbf{\Phi}\, \boldsymbol{v}^{m,j}, \\ \hat{\mathbf{b}}^{\mathbf{m}} &= \mathbf{\Phi}\, \boldsymbol{b}^{m,j}, \end{aligned} \tag{2.16}$$

where $\mathbf{\Phi}$ is a $2 \times 2q$ matrix of the Lagrangian finite type polynomial $\phi(s)$, $-1 \le s \le 1$, that is:

$$\mathbf{\Phi}(s) = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \cdots & \phi_q & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \cdots & 0 & \phi_q \end{bmatrix}. \tag{2.17}$$

The number $q-1$ is equal to the degree of the polynomial approximation. For *linear* elements, such as the one illustrated in Figure 2.3 these polynomials are equal to:

$$\begin{aligned} \phi_1(s) &= \frac{1}{2}(1-s), \\ \phi_2(s) &= \frac{1}{2}(1+s), \end{aligned} \tag{2.18}$$

and, for *quadratic* elements (see Figure 2.5), they are given by:

$$\begin{aligned} \phi_1(s) &= \frac{1}{2}s(s-1), \\ \phi_2(s) &= 1-s^2, \\ \phi_3(s) &= \frac{1}{2}s(s+1). \end{aligned} \tag{2.19}$$

Moreover, the vectors $\boldsymbol{v}^{m,j}$ and $\boldsymbol{b}^{m,j}$ in Equation (2.16) are of length $2q$ and they consist of the velocity and tension vectors respectively of the q successive nodes of the element Γ_m^j. Such an element curve written in a parametric form, say $\mathbf{y}^{\mathbf{m}}(s)$, can be written in terms of the same interpolation functions, thus:

$$\mathbf{y}^{\mathbf{m}}(s) = \phi_i(s)\mathbf{y}^i, \tag{2.20}$$

where $\mathbf{y}^i$ denote the collocation points of the considered element.

Substituting the polynomials (2.16) into the integral Equation (1.76), subjected to the discretized boundary condition, we obtain the following equation for an arbitrary collocation point $\mathbf{x}^p \in \Gamma_m$:

$$
\begin{aligned}
c_{li}^{p} v_i^{m,p} \quad &+ \quad \sum_{k=0}^{M} \sum_{j=1}^{L_k} \left(\int_{\Gamma_k^j} q_{li}(\mathbf{x}^{\mathbf{p}},\ \mathbf{y}) \hat{v}_i^k \, \mathrm{d}\Gamma_y + n_l^{m,p} \int_{\Gamma_k^j} q_i \hat{v}_i^k \, \mathrm{d}\Gamma_y \right) \\
&+ \quad \hat{\varphi}_l^{mr,p} \sum_{j=1}^{L_m} \int_{\Gamma_m^j} \hat{\varphi}_i^{mr} \hat{v}_i^m \, \mathrm{d}\Gamma_y \qquad\qquad (2.21) \\
&= \quad \sum_{k=0}^{M} \sum_{j=1}^{L_k} \left(\int_{\Gamma_k^j} u_{li}(\mathbf{x}^{\mathbf{p}},\ \mathbf{y}) \hat{b}_i^k \, \mathrm{d}\Gamma_y + n_l^{m,p} \int_{\Gamma_k^j} u_i \hat{b}_i^k \, \mathrm{d}\Gamma_y \right), \qquad \mathbf{x}^p \in \Gamma_m \, ,
\end{aligned}
$$

where the index r sums the three types of rigid-body motion $\hat{\boldsymbol{\varphi}}^{\boldsymbol{m},\boldsymbol{r}}$ of the node $\mathbf{x}^{\mathbf{p}}$ and $\mathbf{n}^{\mathbf{m},\mathbf{p}}$, $\mathbf{v}^{\mathbf{m},\mathbf{p}}$ are the outer normal and the unknown velocity at node 'p' situated on Γ_m respectively. If we write the above equation in matrix form, one obtains the following expression:

$$
\begin{aligned}
c_{li}^{p} \begin{pmatrix} v_1^{m,p} \\ v_2^{m,p} \end{pmatrix} \quad &+ \quad \sum_{k=0}^{M} \sum_{j=1}^{L_k} \left[\int_{\Gamma_k^j} \begin{pmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{pmatrix} + \begin{pmatrix} n_1(\mathbf{x}^p) \\ n_2(\mathbf{x}^p) \end{pmatrix} \int_{\Gamma_k^j} \begin{pmatrix} q_1 & q_2 \end{pmatrix} \right] \hat{\mathbf{v}}^{\mathbf{k}} \, d\Gamma_y \\
&+ \quad \sum_{j=1}^{L_m} \int_{\Gamma_m^j} \begin{pmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{pmatrix} \hat{\mathbf{v}}^{\mathbf{m}} \, d\Gamma_y \\
&= \quad \sum_{k=0}^{M} \sum_{j=1}^{L_k} \left[\int_{\Gamma_k^j} \begin{pmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{pmatrix} + \begin{pmatrix} n_1(\mathbf{x}^p) \\ n_2(\mathbf{x}^p) \end{pmatrix} \int_{\Gamma_k^j} \begin{pmatrix} u_1 & u_2 \end{pmatrix} \right] \hat{\mathbf{b}}^{\mathbf{k}} \, d\Gamma_y \, ,
\end{aligned}
$$

$$(2.22)$$

where $\hat{\mathbf{v}}^{\mathbf{k}}$ and $\hat{\mathbf{b}}^{\mathbf{k}}$ are the velocity function and the surface tension function respectively in a typical element Γ_k^j (such as those expressed in Equation (2.16)). t_{ij} are the elements of the rigid-body motion matrix:

$$
\begin{aligned}
t_{11} &= \frac{1}{L} + \frac{(x_2 - x_2^{'})(y_2 - x_2^{'})}{I}, \\
t_{12} &= -\frac{(x_2 - x_2^{'})(y_1 - x_1^{'})}{I}, \\
t_{21} &= -\frac{(x_1 - x_1^{'})(y_2 - x_2^{'})}{I}, \\
t_{22} &= \frac{1}{L} + \frac{(x_1 - x_1^{'})(y_1 - x_1^{'})}{I}.
\end{aligned}
\tag{2.23}
$$

From Equation (2.22), actually we need to calculate the following five typical integrals for each element:

$$
\int_{-1}^{1} \phi_i q_{kl}(\mathbf{x}^{\mathbf{P}}, \mathbf{y})|J(\mathbf{y})|\, ds, \quad \text{and} \quad \int_{-1}^{1} \phi_i u_{kl}(\mathbf{x}^{\mathbf{P}}, \mathbf{y})|J(\mathbf{y})|\, ds, \tag{2.24}
$$

$$
\int_{-1}^{1} \phi_i \hat{\varphi}_k^{l,m}(\mathbf{y})|J(\mathbf{y})|\, ds, \tag{2.25}
$$

$$
\int_{-1}^{1} \phi_i q_l(\mathbf{y})|J(\mathbf{y})|\, ds, \quad \text{and} \quad \int_{-1}^{1} \phi_i u_l(\mathbf{y})|J(\mathbf{y})|\, ds, \tag{2.26}
$$

where $|J(\mathbf{y})|$ is the Jacobian, given by:

$$
|J(\mathbf{y})| = \sqrt{(\phi_i^{\prime} y_1^i)^2 + (\phi_i^{\prime} y_2^i)^2}, \tag{2.27}
$$

and the prime $'$ denotes the derivative with respect s. In the following, we discuss Equation (2.22) and integration (2.24), (2.25) and (2.26) in details for *linear* element and *quadratic* element.

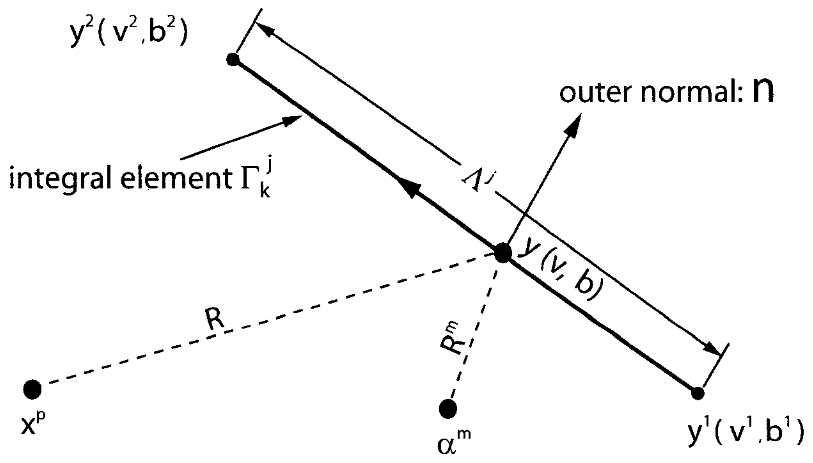


Figure 2.3 : Linear element Γ_k^j of the boundary Γ_k used for the interpolation of the velocity, the surface tension over the boundary.

2.2.1 Formulation for Linear Elements

We consider a linear variation of $\hat{\mathbf{v}}^m$ and $\hat{\mathbf{b}}^{\mathbf{m}}$ in Equation (2.16) over an element Γ_m^j. As we said above, the values of $\hat{\mathbf{v}}^{\mathbf{m}}$ and $\hat{\mathbf{b}}^{\mathbf{m}}$ at any place of the element can be expressed in terms of their nodal values and two linear interpolation functions (2.18). Furthermore the boundary coordinates $\hat{\mathbf{y}}^{\mathbf{m}}$ can also be written in terms of the same functions, thus we have:

$$\begin{aligned}
\hat{\mathbf{v}}^{\mathbf{m}}(s) &= \phi_1(s)\mathbf{v}^1 + \phi_2(s)\mathbf{v}^2, \\
\hat{\mathbf{b}}^{\mathbf{m}}(s) &= \phi_1(s)\mathbf{b}^1 + \phi_2(s)\mathbf{b}^2, \\
\hat{\mathbf{y}}^{\mathbf{m}}(s) &= \phi_1(s)\mathbf{y}^1 + \phi_2(s)\mathbf{y}^2,
\end{aligned} \tag{2.28}$$

where $\mathbf{y}^i$ are the coordinates of the nodal points of the element under consideration, and $\mathbf{v}^{\mathbf{i}}$, $\mathbf{b}^{\mathbf{i}}$ (i = 1, 2) are the values of velocity and surface tension respectively at nodal points $\mathbf{y}^{\mathbf{i}}$, which are shown in Figure 2.3. From Equation (2.17) we have the expression for $\mathbf{\Phi}$ given by:

$$\mathbf{\Phi}(s) = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 \\ 0 & \phi_1 & 0 & \phi_2 \end{bmatrix}. \tag{2.29}$$

Substituting Equation (2.28) into (2.22), one obtains

$$\begin{aligned} & c^p_{li} \begin{pmatrix} v_1^{m,p} \\ v_2^{m,p} \end{pmatrix} \\ & + \sum_{k=0}^{M} \sum_{j=1}^{L_k} \left[\begin{pmatrix} h_{11}^{kp,j} & h_{12}^{kp,j} & h_{13}^{kp,j} & h_{14}^{kp,j} \\ h_{21}^{kp,j} & h_{22}^{kp,j} & h_{23}^{kp,j} & h_{24}^{kp,j} \end{pmatrix} + \begin{pmatrix} \hbar_{11}^{kp,j} & \hbar_{12}^{kp,j} & \hbar_{13}^{kp,j} & \hbar_{14}^{kp,j} \\ \hbar_{21}^{kp,j} & \hbar_{22}^{kp,j} & \hbar_{23}^{kp,j} & \hbar_{24}^{kp,j} \end{pmatrix} \right] \boldsymbol{v}^{k,j} \\ & + \sum_{j=1}^{L_m} \begin{pmatrix} d_{11}^{mp,j} & d_{12}^{mp,j} & d_{13}^{mp,j} & d_{14}^{mp,j} \\ d_{21}^{mp,j} & d_{22}^{mp,j} & d_{23}^{mp,j} & d_{24}^{mp,j} \end{pmatrix} \boldsymbol{v}^{m,j} \\ & = \sum_{k=0}^{M} \sum_{j=1}^{L_k} \left[\begin{pmatrix} g_{11}^{kp,j} & g_{12}^{kp,j} & g_{13}^{kp,j} & g_{14}^{kp,j} \\ g_{21}^{kp,j} & g_{22}^{kp,j} & g_{23}^{kp,j} & g_{24}^{kp,j} \end{pmatrix} + \begin{pmatrix} \zeta_{11}^{kp,j} & \zeta_{12}^{kp,j} & \zeta_{13}^{kp,j} & \zeta_{14}^{kp,j} \\ \zeta_{21}^{kp,j} & \zeta_{22}^{kp,j} & \zeta_{23}^{kp,j} & \zeta_{24}^{kp,j} \end{pmatrix} \right] \boldsymbol{b}^{k,j}, \end{aligned} \tag{2.30}$$

where the superscript "j" denotes the values on the element j. The vector $\boldsymbol{v}^{k,j}$ and $\boldsymbol{b}^{k,j}$ are the velocity and tension vectors at the nodes of the element Γ^j_k, which are defined as follows:

$$\boldsymbol{v}^{k,j} = \begin{pmatrix} v_1^1 \\ v_2^1 \\ v_2^1 \\ v_2^2 \end{pmatrix}^{k,j} ; \qquad \boldsymbol{b}^{k,j} = \begin{pmatrix} b_1^1 \\ b_2^1 \\ b_2^1 \\ b_2^2 \end{pmatrix}^{k,j} .$$

The expressions of $h_{rs}^{kp,\,j}$, $\hbar_{rs}^{kp,\,j}$, $g_{rs}^{kp,\,j}$, $\zeta_{rs}^{kp,\,j}$ and $d_{rs}^{kp,\,j}$ ($r = 1,\ 2,\ \ s = 1,\ 2,\ 3,\ 4$) are listed as below:

$$\begin{aligned}
h_{l,\,2i-2+m}^{kp,\,j} &= \int_{\Gamma_k^j} [q_{l\,m}(\mathbf{x}^p,\ \mathbf{y})]\,\phi_i\,d\Gamma_y &&= \int_{-1}^{1} [q_{l\,m}(\mathbf{x}^p,\ \mathbf{y})]\,\phi_i|J|\,ds,\\
g_{l,\,2i-2+m}^{kp,\,j} &= \int_{\Gamma_k^j} [u_{l\,m}(\mathbf{x}^p,\ \mathbf{y})]\,\phi_i\,d\Gamma_y &&= \int_{-1}^{1} [u_{l\,m}(\mathbf{x}^p,\ \mathbf{y})]\,\phi_i|J|\,ds,\\
d_{l,\,2i-2+m}^{kp,\,j} &= \int_{\Gamma_k^j} [t_{l\,m}(\mathbf{x}^p,\ \mathbf{y})]\,\phi_i\,d\Gamma_y &&= \int_{-1}^{1} [t_{l\,m}(\mathbf{x}^p,\ \mathbf{y})]\,\phi_i|J|\,ds, \qquad (2.31)\\
\hbar_{l,\,2i-2+m}^{kp,\,j} &= n_l^p \int_{\Gamma_k^j} [q_m(\boldsymbol{\alpha}^m,\ \mathbf{y})]\,\phi_i\,d\Gamma_y &&= n_l^p \int_{-1}^{1} [q_m(\boldsymbol{\alpha}^m,\ \mathbf{y})]\,\phi_i|J|\,ds,\\
\zeta_{l,\,2i-2+m}^{kp,\,j} &= n_l^p \int_{\Gamma_k^j} [u_m(\boldsymbol{\alpha}^m,\ \mathbf{y})]\,\phi_i\,d\Gamma_y &&= n_l^p \int_{-1}^{1} [u_m(\boldsymbol{\alpha}^m,\ \mathbf{y})]\,\phi_i|J|\,ds,
\end{aligned}$$

where J is the Jacobian, $i = 1,\ 2$, and $l,\ m = 1,\ 2$. From Equation (2.25), (2.26), we can see that the integration of $d_{rs}^{kp,\,j}, \hbar_{rs}^{kp,\,j}$ and $\zeta_{rs}^{kp,\,j}$ are independent of $\mathbf{x^p}$. As a result, these integrations just need to be computed only once, when $\mathbf{x^p}$ is varied over all the collocation points.

In this case of linear elements, we can easily obtain the expressions of the Jacobian $|J|$

$$|J| = \sqrt{\left(\frac{\partial y_1}{\partial s}\right)^2 + \left(\frac{\partial y_2}{\partial s}\right)^2} = \frac{1}{2}|y^{j+1} - y^j| = \frac{1}{2}\Lambda^j, \qquad (2.32)$$

and the outer normal $\mathbf{n}$ of element Γ_k^j

$$\mathbf{n} = \frac{1}{|J|}\begin{pmatrix} -a_2 \\ a_1 \end{pmatrix} \qquad \text{with} \qquad a_i = \frac{1}{2}(y_i^1 - y_i^2), \qquad (2.33)$$

where Λ^j is the length of the element Γ_k^j (for convenience, we use $\mathbf{y^1}$, $\mathbf{y^2}$ for the nodal points $\mathbf{y^j}$, $\mathbf{y^{j+1}}$ in the element Γ_k^j, as shown in Figure 2.3).

Further analysis of (2.24) gives the following eight integral equations are to be calculated for each element Γ_k^j, say, with nodal points $\mathbf{y}^1$ and $\mathbf{y}^2$, related to observation node $\mathbf{x}^p$

$$\begin{aligned} I_1^G(\lambda) &= \int_{-1}^{1}(1+\lambda s)\frac{r_1^2}{R^2}ds; & I_2^G(\lambda) = \int_{-1}^{1}(1+\lambda s)\frac{r_2^2}{R^2}ds, \\ I_3^G(\lambda) &= \int_{-1}^{1}(1+\lambda s)\frac{r_1r_2}{R^2}ds; & I_4^G(\lambda) = -\int_{-1}^{1}(1+\lambda s)\ln|R|\,\mathrm{d}s, \end{aligned} \tag{2.34}$$

and

$$\begin{aligned} I_1^H(\lambda) &= \int_{-1}^{1}(1+\lambda s)\frac{r_1^3}{R^4}ds; & I_2^H(\lambda) = \int_{-1}^{1}(1+\lambda s)\frac{r_2^3}{R^4}ds, \\ I_3^H(\lambda) &= \int_{-1}^{1}(1+\lambda s)\frac{r_1^2r_2}{R^4}ds; & I_4^H(\lambda) = \int_{-1}^{1}(1+\lambda s)\frac{r_1r_2^2}{R^4}ds, \end{aligned} \tag{2.35}$$

where $\lambda = \pm 1$, and

$$r_i = a_i s + b_i, \qquad \text{with} \qquad b_i = \frac{1}{2}(2x_i^p - y_i^1 - y_i^2). \tag{2.36}$$

We note that

$$I_1^G(\lambda) + I_2^G(\lambda) = 2. \tag{2.37}$$

Thus, we need to evaluate a total of 14 integrals for each element "Γ_k^j" and nodal point $\mathbf{x}^p$.

From the integrals I_i^G, I_i^H, we can obtain the expression of h_{ij} and g_{ij} (omitting the superscription k, p, j) in Equation (2.30), thus:

$$\begin{aligned} h_{11} &= \frac{1}{2\pi}\left[-a_2I_1^H(-1) + a_1I_3^H(-1)\right]; & h_{12} &= \frac{1}{2\pi}\left[-a_2I_3^H(-1) + a_1I_4^H(-1)\right], \\ h_{13} &= \frac{1}{2\pi}\left[-a_2I_1^H(1) + a_1I_3^H(1)\right]; & h_{14} &= \frac{1}{2\pi}\left[-a_2I_3^H(1) + a_1I_4^H(1)\right], \\ h_{22} &= \frac{1}{2\pi}\left[-a_2I_4^H(-1) + a_1I_2^H(-1)\right]; & h_{21} &= h_{12}, \\ h_{24} &= \frac{1}{2\pi}\left[-a_2I_4^H(1) + a_1I_2^H(1)\right]; & h_{23} &= h_{14}, \end{aligned} \tag{2.38}$$

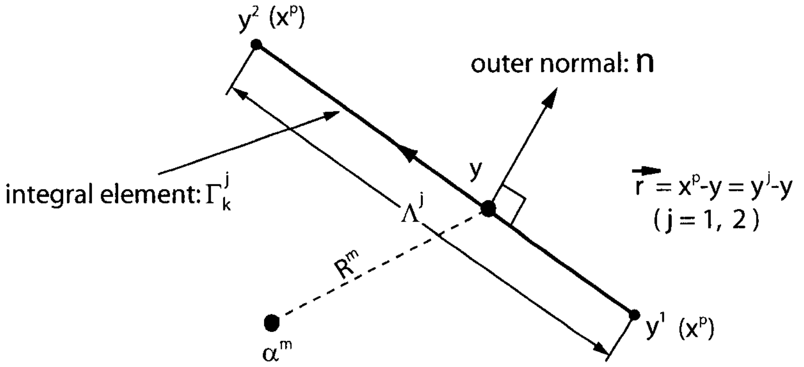


Figure 2.4 : In the case of that $\mathbf{x}^p$ is also a nodal point of the element under consideration.

and

$$\begin{aligned}
g_{11} &= \frac{\Lambda^j}{16\pi}\left[I_4^G(-1) + I_1^G(-1)\right]; & g_{12} &= \frac{\Lambda^j}{16\pi}\left[I_3^G(-1)\right], \\
g_{13} &= \frac{\Lambda^j}{16\pi}\left[I_4^G(1) + I_1^G(1)\right]; & g_{14} &= \frac{\Lambda^j}{16\pi}\left[I_3^G(1)\right], \\
g_{22} &= \frac{\Lambda^j}{16\pi}\left[I_4^G(-1) + I_2^G(-1)\right]; & g_{21} &= g_{12}^{pj}, \\
g_{24} &= \frac{\Lambda^j}{16\pi}\left[I_4^G(1) + I_2^G(1)\right]; & g_{23} &= g_{14}^{pj}.
\end{aligned} \tag{2.39}$$

These integrals are numerically approximated by applying a Gaussian quadrature formula. Hence, the integrals are approximated by a finite sum, the terms of which consist of integrand evaluations at a number of integration points, multiplied by an associated weighting factor. Normally, a four-point Gaussian quadrature formula is applied in the linear element solution (and a six-point Gaussian quadrature formula for quadratic element case) as shown in Appendix A.

However, when $\mathbf{x}^p$ is also a nodal point of the element under consideration, the integral equations in (2.24) have a singularity in the integration domain. So some extra effort is needed in order to approximate these integrals correctly. In this case (i.e., a linear element), the vector $\overrightarrow{r} = \mathbf{x}^p - \mathbf{y}$, which is equal to $\mathbf{y}^j - \mathbf{y}$, $(j = 1,\ 2)$, and the outward normal vector $\mathbf{n}$ are orthogonal (see Figure 2.4), and the integral equations in (2.24) can be calculated analytically. From Figure 2.4, it is clear that we have:

$$\int_{-1}^{1} \phi_k q_{lm}(\mathbf{y}^{\mathbf{j}},\ \mathbf{y})|J(\mathbf{y})|\,\mathrm{d}s = 0, \tag{2.40}$$

and

$$\begin{aligned}\int_{-1}^{1} \phi_k u_{lm}(\mathbf{y}^{\mathbf{j}},\ \mathbf{y})|J(\mathbf{y})|\,\mathrm{d}s &= \frac{1}{2\pi\Lambda^j}\left[(\delta_{l1}\delta_{m2}+\delta_{l2}\delta_{m1})a_l a_m + \delta_{lm}a_l^2\right] \\ &- \frac{\Lambda^j}{16\pi}\delta_{lm}\left[2\ln\Lambda^j - 1 - 2(\delta_{k1}\delta_{j1}+\delta_{k2}\delta_{j2})\right].\end{aligned} \tag{2.41}$$

Now, we have obtained the integration of h_{ij} and g_{ij} in Equation (2.30). Next we calculate the integration about the movement of a rigid-body d_{ij} in Equation (2.30). As mentioned above, these integrals are relatively simple because they are independent of the node $\mathbf{x}^p$. On the other hand, all of the integrals can be calculated analytically in this linear element case, that is:

$$\begin{aligned}
d_{11} &= \frac{\Lambda^j}{2}\left[\frac{1}{L} + \frac{(x_2^p - x_2^{'})}{3I}\left(2y_2^1 + y_2^2 - 3x_2^{'}\right)\right]; \quad d_{12} = \frac{\Lambda^j}{2}\left[\frac{(x_2^{'} - x_2^p)}{3I}\left(2y_1^1 + y_1^2 - 3x_1^{'}\right)\right], \\
d_{13} &= \frac{\Lambda^j}{2}\left[\frac{1}{L} + \frac{(x_2^p - x_2^{'})}{3I}\left(y_2^1 + 2y_2^2 - 3x_2^{'}\right)\right]; \quad d_{14} = \frac{\Lambda^j}{2}\left[\frac{(x_2^{'} - x_2^p)}{3I}\left(y_1^1 + 2y_1^2 - 3x_1^{'}\right)\right], \\
d_{21} &= \frac{\Lambda^j}{2}\left[\frac{(x_1^{'} - x_1^p)}{3I}\left(2y_2^1 + y_2^2 - 3x_2^{'}\right)\right]; \quad d_{22} = \frac{\Lambda^j}{2}\left[\frac{1}{L} + \frac{(x_1^p - x_1^{'})}{3I}\left(2y_1^1 + y_1^2 - 3x_1^{'}\right)\right], \\
d_{23} &= \frac{\Lambda^j}{2}\left[\frac{(x_1^{'} - x_1^p)}{3I}\left(y_2^1 + 2y_2^2 - 3x_2^{'}\right)\right]; \quad d_{24} = \frac{\Lambda^j}{2}\left[\frac{1}{L} + \frac{(x_1^p - x_1^{'})}{3I}\left(y_1^1 + 2y_1^2 - 3x_1^{'}\right)\right].
\end{aligned} \tag{2.42}$$

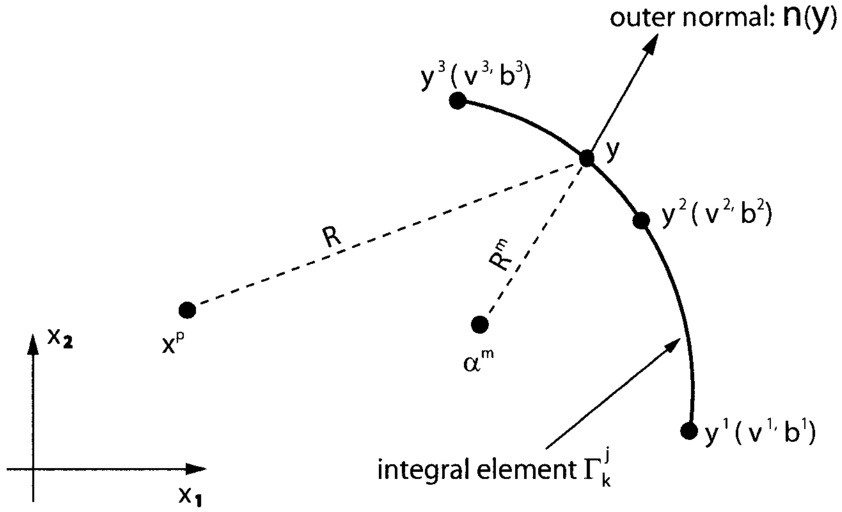


Figure 2.5 : Quadratic element for the interpolation of velocity, surface tension, and boundary coordinates.

The integration of the sink contributions in Equation (2.26) are a little bit more complex because the integrands are not simple functions. Actually, we cannot simplify them. Thus, we just use Gaussian quadrature formula to get the integral values. So far, we have discussed the methods that allow to obtain the integration results of h_{ij}, g_{ij}, and d_{ij} in equation 2.30. In the next section, we will discuss some special cases for the quadratic element approach.

2.2.2 Formulation for Quadratic Elements

We consider a quadratic variation of $\hat{\mathbf{v}}^{\mathbf{m}}$ and $\hat{\mathbf{b}}^{\mathbf{m}}$ over an element. Thus we need three points to define the element, as shown in Figure 2.5. In this case, we have Lagrangian interpolation polynomial matrix given by:

$$\mathbf{\Phi}(s) = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \phi_3 & 0 \\ 0 & \phi_1 & 0 & \phi_2 & 0 & \phi_3 \end{bmatrix}. \qquad (2.43)$$

As mentioned in Equation (2.16), the values of $\hat{\mathbf{v}}^{\mathbf{m}}$ and $\hat{\mathbf{b}}^{\mathbf{m}}$ at any point of the element can be expressed in terms of their nodal values and three quadratic interpolation functions ϕ_1, ϕ_2 and ϕ_3, thus:

$$\begin{aligned}
\hat{\mathbf{v}}^{\mathbf{m}}(s) &= \phi_1(s)\mathbf{v}^1 + \phi_2(s)\mathbf{v}^2 + \phi_3(s)\mathbf{v}^3, \\
\hat{\mathbf{b}}^{\mathbf{m}}(s) &= \phi_1(s)\mathbf{b}^1 + \phi_2(s)\mathbf{b}^2 + \phi_3(s)\mathbf{b}^3, \qquad (2.44) \\
\hat{\mathbf{y}}^{\mathbf{m}}(s) &= \phi_1(s)\mathbf{y}^1 + \phi_2(s)\mathbf{y}^2 + \phi_3(s)\mathbf{y}^3.
\end{aligned}$$

Substituting Equation (2.44) into Equation (2.22), one obtains the descretized form of the equation for a typical quadratic element, which yields:

$$\begin{aligned}
c_{li}^{p}\begin{pmatrix} v_1^{m,p} \\ \\ v_2^{m,p} \end{pmatrix} &+ \sum_{k=0}^{M}\sum_{j=1}^{L_k}\left[\begin{pmatrix} h_{11}^{kp,j} & h_{12}^{kp,j} & h_{13}^{kp,j} & h_{14}^{kp,j} & h_{15}^{kp,j} & h_{16}^{kp,j} \\ \\ h_{21}^{kp,j} & h_{22}^{kp,j} & h_{23}^{kp,j} & h_{24}^{kp,j} & h_{25}^{kp,j} & h_{26}^{kp,j} \end{pmatrix}\right. \\
&\qquad \left. + \begin{pmatrix} \hbar_{11}^{kp,j} & \hbar_{12}^{kp,j} & \hbar_{13}^{kp,j} & \hbar_{14}^{kp,j} & \hbar_{15}^{kp,j} & \hbar_{16}^{kp,j} \\ \\ \hbar_{21}^{kp,j} & \hbar_{22}^{kp,j} & \hbar_{23}^{kp,j} & \hbar_{24}^{kp,j} & \hbar_{25}^{kp,j} & \hbar_{26}^{kp,j} \end{pmatrix}\right]\boldsymbol{v}^{k,\,j} \\
&+ \sum_{j=1}^{L_m}\begin{pmatrix} d_{11}^{mp,\,j} & d_{12}^{mp,\,j} & d_{13}^{mp,\,j} & d_{14}^{mp,\,j} & d_{15}^{mp,\,j} & d_{16}^{mp,\,j} \\ \\ d_{21}^{mp,\,j} & d_{22}^{mp,\,j} & d_{23}^{mp,\,j} & d_{24}^{mp,\,j} & d_{25}^{mp,\,j} & d_{26}^{mp,\,j} \end{pmatrix}\boldsymbol{v}^{m,\,j} \\
&= \sum_{k=0}^{M}\sum_{j=1}^{L_k}\left[\begin{pmatrix} g_{11}^{kp,j} & g_{12}^{kp,j} & g_{13}^{kp,j} & g_{14}^{kp,j} & g_{15}^{kp,j} & g_{16}^{kp,j} \\ \\ g_{21}^{kp,j} & g_{22}^{kp,j} & g_{23}^{kp,j} & g_{24}^{kp,j} & g_{25}^{kp,j} & g_{26}^{kp,j} \end{pmatrix}\right. \\
&\qquad \left. + \begin{pmatrix} \zeta_{11}^{kp,j} & \zeta_{12}^{kp,j} & \zeta_{13}^{kp,j} & \zeta_{14}^{kp,j} & \zeta_{15}^{kp,j} & \zeta_{16}^{kp,j} \\ \\ \zeta_{21}^{kp,j} & \zeta_{22}^{kp,j} & \zeta_{23}^{kp,j} & \zeta_{24}^{kp,j} & \zeta_{25}^{kp,j} & \zeta_{26}^{kp,j} \end{pmatrix}\right]\boldsymbol{b}^{k,\,j},
\end{aligned}$$

$$(2.45)$$

where

$$\boldsymbol{v}^{k,\,j} = \begin{pmatrix} v_1^1 \\ v_2^1 \\ v_1^2 \\ v_2^2 \\ v_1^3 \\ v_2^3 \end{pmatrix}^{k,\,j} \qquad \text{and} \qquad \boldsymbol{b}^{k,\,j} = \begin{pmatrix} b_1^1 \\ b_2^1 \\ b_1^2 \\ b_2^2 \\ b_1^3 \\ b_2^3 \end{pmatrix}^{k,\,j},$$

where k, p, j are the same as that in Equation (2.30).

From Equation (2.44), we obtain the Jacobian $|J|$ for the quadratic element case:

$$|J| = \sqrt{\left(\frac{\partial y_1}{\partial s}\right)^2 + \left(\frac{\partial y_2}{\partial s}\right)^2} = \sqrt{(2a_1 s + b_1)^2 + (2a_2 s + b_2)^2}, \tag{2.46}$$

where

$$\begin{aligned} a_i &= -\frac{1}{2}(y_i^1 - 2y_i^2 + y_i^3), \\ b_i &= -\frac{1}{2}(y_i^3 - y_i^1). \end{aligned} \tag{2.47}$$

In this case, the outward normal vector $\mathbf{n}$ at any point $\mathbf{y}$ in a typical element Γ_m^j can be expressed as:

$$\mathbf{n}(\mathbf{y}) = \begin{pmatrix} \frac{y_2'}{\sqrt{(y_1')^2+(y_2')^2}} \\ \frac{-y_1'}{\sqrt{(y_1')^2+(y_2')^2}} \end{pmatrix} = \frac{1}{|J|} \begin{pmatrix} -2a_2 s - b_2 \\ 2a_1 s + b_1 \end{pmatrix}, \tag{2.48}$$

where the $'$ denotes the first derivative with respect to s.

Now, we go back to Equations (2.24), (2.25) and (2.26), and discuss how we can solve these integrals. For the integrations of a rigid-body motion (Equation 2.25) and sink contribution (Equation 2.26), there is not any special case that we need to consider. On the other hand, because of the quadratic element, the outer normal $\mathbf{n}(\mathbf{y})$ and the Jacobian J are no longer constants, so that no further simplification can be carried out on these integrations. Thus, we just use a six-point Gaussian integral formula to calculate these integrations. The integrals of Equation (2.24) with a quadratic element are quite different from those for a linear element. Let's write Equation (2.24) in detail in this case. The following two integrals are to be calculated for each element Γ_k^j, i.e., with the nodal points $\mathbf{y}^1$, $\mathbf{y}^2$ and $\mathbf{y}^3$ related to the node $\mathbf{x}^p$

$$\begin{aligned} h_{l,\,2i-2+m}^{pk,\,j} &= \int_{\Gamma_k^j} \phi_i q_{lm}\, d\Gamma_y \;=\; \frac{1}{\pi}\int_{-1}^{1} \phi_i \frac{r_l r_m}{R^4}\psi(s)\, ds, \\ g_{l,\,2i-2+m}^{pk,\,j} &= \int_{\Gamma_k^j} \phi_i u_{lm}\, d\Gamma_y \;=\; \frac{1}{4\pi}\int_{-1}^{1} \phi_i \left[\delta_{lm}\ln\frac{1}{R} + \frac{r_l r_m}{R^2}\right] |J|\, ds, \end{aligned} \tag{2.49}$$

where

$$\begin{aligned} i &= 1,\,2,\,3 \quad \text{and} \quad l,\,m = 1,\,2, \\ r_j &= a_j s^2 + b_j s + c_j; \qquad j = 1,\,2, \\ c_j &= x_j^p - y_j^2, \\ \psi(s) &= (r_1 n_1 + r_2 n_2)|J| \\ &= (a_1 b_2 - a_2 b_1)s^2 + 2(a_1 c_2 - a_2 c_1)s + (b_1 c_2 - b_2 c_1). \end{aligned} \tag{2.50}$$

These integrals can also be computed by using a six-point (or more precise) Gaussian quadrature formula. However, there is also a special case that needs to be considered. When the node $\mathbf{x}^p$ is also a nodal point of the element under consideration, say, $\mathbf{x}^p = \mathbf{y}^1$, $\mathbf{y}^2$, or $\mathbf{y}^3$, the integral (2.49) has a singularity. Fortunately, this kind of singularity can be easily removed after some analytical manipulation. In this case, from (2.50), we have the expression of r_j:

$$r_j = \begin{cases} (s+1)(a_j s + b_j - a_j), & \mathbf{x}^p = \mathbf{y}^1, \\ s(a_j s + b_j), & \mathbf{x}^p = \mathbf{y}^2, \\ (s-1)(a_j s + b_j + a_j), & \mathbf{x}^p = \mathbf{y}^3. \end{cases} \tag{2.51}$$

Substituting Equation (2.51) into Equation (2.49), we obtain the following expression:

$$\begin{aligned} h^{kp,j}_{l,2i-2+m} &= \frac{(a_1 b_2 - a_2 b_1)}{\pi} \int_{-1}^{1} \phi_i \frac{\tilde{r}_l \tilde{r}_m}{\tilde{R}^4}\, ds, \\ g^{kp,j}_{l,2i-2+m} &= \delta_{lm} \frac{1}{4\pi} I_k + \frac{1}{4\pi} \int_{-1}^{1} \phi_i \left[\delta_{lm} \ln \frac{1}{\tilde{R}} + \frac{\tilde{r}_l \tilde{r}_m}{\tilde{R}^2} \right] |J|\, ds, \end{aligned} \tag{2.52}$$

where

$$\tilde{r}_i = \begin{cases} a_i s + b_i - a_i, & \mathbf{x}^p = \mathbf{y}^1, \\ a_i s + b_i, & \mathbf{x}^p = \mathbf{y}^2, \\ a_i s + b_i + a_i, & \mathbf{x}^p = \mathbf{y}^3, \end{cases} \tag{2.53}$$

$$\tilde{R} = \sqrt{\tilde{r}_1^2 + \tilde{r}_2^2},$$

and

$$I_k = \begin{cases} \int_{-1}^{1} \phi_k \ln \frac{1}{1+s} |J(s)|\, ds, & \mathbf{x}^p = \mathbf{y}^1, \\ \int_{-1}^{1} \phi_k \ln \frac{1}{s} |J(s)|\, ds, & \mathbf{x}^p = \mathbf{y}^2, \\ \int_{-1}^{1} \phi_k \ln \frac{1}{1-s} |J(s)|\, ds, & \mathbf{x}^p = \mathbf{y}^3. \end{cases} \tag{2.54}$$

By changing the variable in I_k, the integral Equation (2.54) can be written as:

$$I_k = I_k^G - \xi \ln 2 \int_{-1}^{1} \phi_k(s)|J(s)|\, ds, \tag{2.55}$$

where

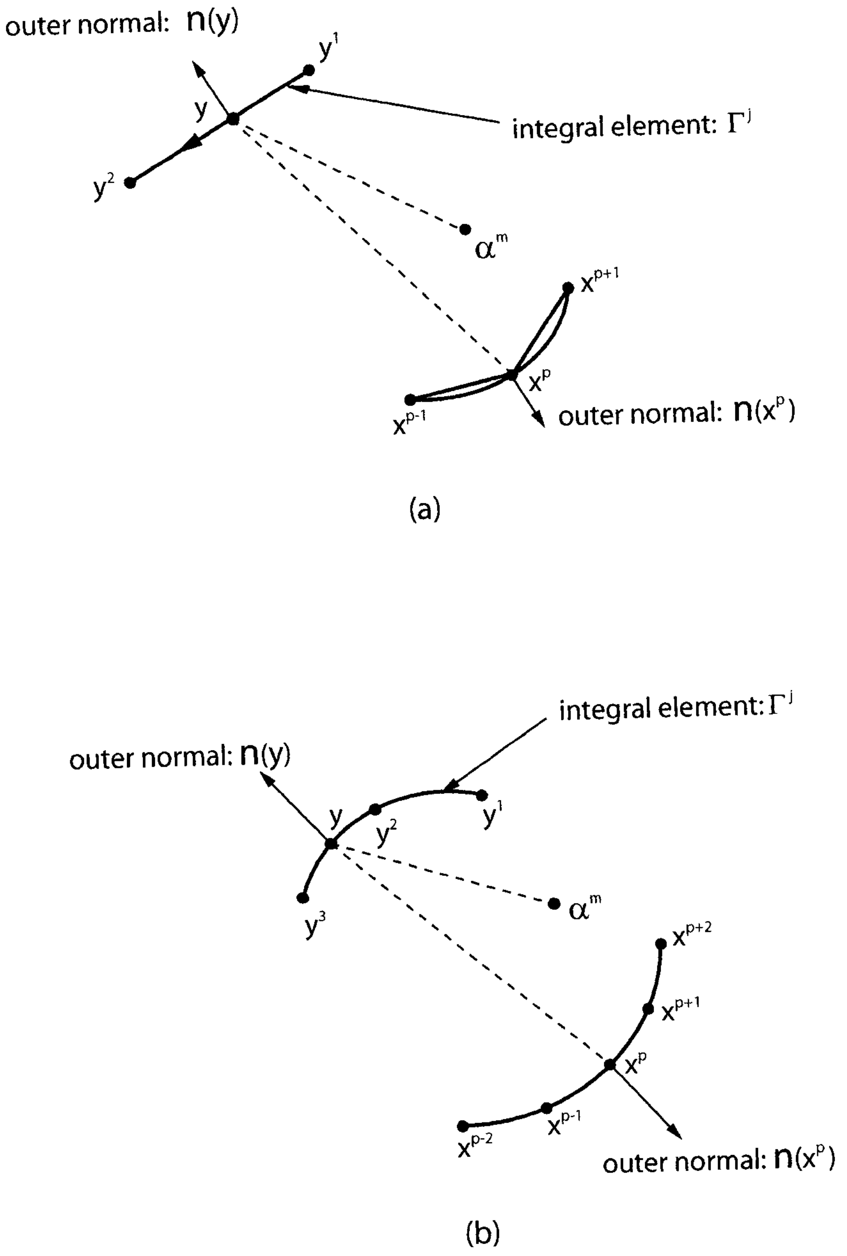
$$\begin{aligned}
I_k^G &= 2\int_0^1 \phi_k(2t-1)|J(2t-1)| \ln\frac{1}{t}\, dt, \qquad \xi = 1, \quad \mathbf{x}^p = \mathbf{y}^1, \\
I_k^G &= \int_0^1 \phi_k(-t)|J(-t)| \ln\frac{1}{t}\, dt + \int_0^1 \phi_k(t)|J(t)| \ln\frac{1}{t}\, dt, \qquad \xi = 0, \quad \mathbf{x}^p = \mathbf{y}^2, \\
I_k^G &= 2\int_0^1 \phi_k(1-2t)|J(1-2t)| \ln\frac{1}{t}\, dt, \qquad \xi = 1, \quad \mathbf{x}^p = \mathbf{y}^3.
\end{aligned} \tag{2.56}$$

We can see that the integrals of I_k^G have a logarithmic singularity, and can be calculated by using a Logarithmic Gaussian quadrature formula. Thus, the integrals in Equation (2.52) are not singular, therefore they can be computed using the Gaussian quadrature formula.

In order to obtain the values of $\hbar_{rs}^{kp,j}$ and $\zeta_{rs}^{kp,j}$, we need to know the outer normal vector $\mathbf{n}(\mathbf{x^p})$ at point $\mathbf{x^p}$. However, the way for getting $\mathbf{n}(\mathbf{x^p})$ is different from that used for boundary elements. Figure 2.6 illustrates the procedure. In an usual way, the outer normals are calculated by applying a second order (for a linear element) or fourth order (for a quadratic element) Lagrangian interpolation polynomial. The detail of this calculation is given in Section 2.3.

Thus, for each node $\mathbf{x^p}$, after obtaining the sub-matrix $\mathcal{H}^p$, $\mathscr{H}^p$, $\mathcal{D}^{m,p}$, $\mathcal{G}^p$, and $\mathscr{G}^p$ with elements $h_{il}^{kp,j}$, $\hbar_{il}^{kp,j}$, $d_{il}^{mp,j}$, g_{il}^{pj}, and $\zeta_{il}^{kp,j}$ respectively, we have the matrix equation for multiply-connected domain, i.e.,

$$\left(\mathcal{H}^p + \mathscr{H}^p\right)\boldsymbol{v} + \mathcal{D}^{m,p}\boldsymbol{v}^m = \left(\mathcal{G}^p + \mathscr{G}^p\right)\boldsymbol{b}, \tag{2.57}$$



(a)

(b)

Figure 2.6 : The approximation of outer normal of node $\mathbf{x^p}$:
(a) by fitting a quadratic polynomial,
(b) by applying fourth order Lagrangian interpolation polynomial.

where the superscript 'm' denotes the index of the boundary where node $\mathbf{x}^p$ is located, and $\boldsymbol{v}^m$ denotes the velocity on the boundary Γ_m. The $\boldsymbol{v}$ and $\boldsymbol{b}$ are the vectors of length $2N$ that represent the velocity and surface tension of all successive nodal points respectively, i.e.,

$$\begin{aligned} \boldsymbol{v} &= \left(v_1^1,\, v_2^1,\, v_1^2,\, v_2^2,\, \cdots,\, v_1^N,\, v_2^N\right)^T, \\ \boldsymbol{b} &= \left(b_1^1,\, b_2^1,\, b_1^2,\, b_2^2,\, \cdots,\, b_1^N,\, b_2^N\right)^T. \end{aligned} \tag{2.58}$$

2.2.3 Calculation of the Coefficients $c_{l\,i}^p$

In the previous section, we got the matrix Equation (2.57) for $\mathbf{x}^\mathbf{p}$ with either linear elements or quadratic elements. The $2 \times 2N$ matrix $\mathcal{H}^p$ and $\mathcal{G}^p$ are obtained from the point force contribution in Equation (2.24). Moreover, the coefficients $c_{l\,i}^p$ are incorporated in the matrix $\mathcal{H}^p$. The coefficients $c_{l\,i}^p$ are equal to $\frac{1}{2}\delta_{l\,i}$ for the boundary curve is smooth. In general, for the discretized boundary curve, which is discontinuous at the nodal points that connect two successive elements, the coefficients $c_{l\,i}^p$ are no longer equal to $\frac{1}{2}\delta_{l\,i}$. In that case, the coefficients $c_{l\,i}^p$ can also be obtained analytically [Hartmann 1980]. However, they are complicated to compute. The usual way of avoiding this complicated computation is to consider rigid-body motions. Actually, the above matrix equation have to apply to any physical problem for which the solution does not depend on the geometry, and a rigid-body motion does not change the surface tension nor the boundary curvature. Assume the whole body has a velocity $\mathbf{v}$ in the direction of one of Cartesian coordinates and the surface tension vector $\mathbf{b}$ does not change. Then we have the following equation

$$\mathcal{H}^p\,\boldsymbol{e} = 0 \qquad \text{with} \qquad \boldsymbol{e} = (1,\ 1,\ \cdots,\ 1,\ 1)^T, \tag{2.59}$$

with the coefficients $\mathcal{H}_{ij}^p$ ($i = 1,\ 2$ and $j = 2p-1,\ 2p$) that can be obtained by the following way

$$\begin{aligned}\mathcal{H}^p_{i,2p-1} &= -\sum_{\substack{j=1\\ j\neq p}}^{N} \mathcal{H}^p_{i,2j-1},\\ \mathcal{H}^p_{i,2p} &= -\sum_{\substack{j=1\\ j\neq p}}^{N} \mathcal{H}^p_{i,2j}.\end{aligned} \tag{2.60}$$

Varying the point $\mathbf{x}^p$ in the discretized Equation (2.57) over all collocation points, we obtain a full rank system of $2N$ linear algebraic equations with $2N$ unknowns

$$\mathscr{A}(\boldsymbol{x})\boldsymbol{v} = \mathscr{L}(\boldsymbol{x})\,\boldsymbol{b}(\boldsymbol{x}), \tag{2.61}$$

where $\boldsymbol{x}$ is the vector of length $2N$ that consists of all successive collocation points.

The unknowns $\boldsymbol{v}$ are obtained after solving the above square full rank system by BiConjugate Gradient or BiConjugate Gradients Stabilized method. An iteration method is used since the matrix $\mathscr{A}$ is not symmetric and does not have a sparse structure. Each nodal point is, in the BEM formulation connected to all the other nodes as can be observed from the above outline of the system derivation.

After solving Equation (2.61), we have to perform a time step. Here, we use two way to describe the boundary movement, namely, the Eulerian representation and the Lagrangian representation. In the Eulerian viewpoint (the collocation points on the boundary move in the direction normal to the boundary), we can obtain the new boundary level directly by using

$$\Delta\boldsymbol{x}\big|_{t=t_i} = \Delta t \cdot \boldsymbol{v}\big|_{t=t_i}. \tag{2.62}$$

If the Lagrangian representation is used (the boundary movement is modeled by considering the boundary curve as material points), then the trajectories of these points can be described by

$$\frac{d\boldsymbol{x}}{dt}\Big|_{t=t_i} = \boldsymbol{v}\Big|_{t=t_i}. \tag{2.63}$$

Using the relation (2.63) together with Equation (2.61) yields the following $2N$ nonlinear ODEs:

$$\frac{d\boldsymbol{x}}{dt} = \mathscr{A}^{-1}(\boldsymbol{x})\mathscr{L}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}). \tag{2.64}$$

In the next chapter, we discuss in detail the above system.

2.3 Calculation of the Surface Tension

In the previous sections, we have discussed how to obtain the matrices $\mathscr{A}$ and $\mathscr{L}$ in Equation (2.61). In this section, we discuss the numerical approximation of the body force $\boldsymbol{b} = -\kappa\boldsymbol{n}$. Firstly, we analyze the numerical problems that occur when the collocation points are getting close to each other during the boundary movement. In particular, we consider the cusp regions where the curvature becomes very large in magnitude. On the other hand, the nodes in this region have to lie close to each other because we expect large variations of the boundary velocity field there. Then, we discuss the method of the mesh redistribution and its numerical algorithm implementation.

2.3.1 The Approximation of the Outer Normal Vector and Curvature

A local method is used to approximate the surface tension $\mathbf{b}^i$ at boundary nodes. The accuracy of these approximations has to be of at least of the same order as those used for the BEM. In the case of linear elements, it is sufficient to use an approximate curvature accurate up to the second order. Therefore, both curvature and outer normal are calculated by fitting a quadratic polynomial passing through the nodal point, $\mathbf{x}^2$, and its neighbors $\mathbf{x}^1$ and $\mathbf{x}^3$. We have

$$\mathbf{x}(s) = \phi_i(s)\mathbf{x}^i. \tag{2.65}$$

For the approximate outer normal at nodal point $\mathbf{x}^2 = \mathbf{x}(0)$, one obtains

$$\begin{aligned} \mathbf{n}^2 = \mathbf{n}(0) &= \frac{1}{\sqrt{(x_1^{'})^2 + (x_2^{'})^2}} \begin{pmatrix} x_2^{'} \\ -x_1^{'} \end{pmatrix} \\ &\doteq \frac{1}{\sqrt{(x_1^3 - x_1^1)^2 + (x_2^3 - x_2^1)^2}} \begin{pmatrix} x_2^3 - x_2^1 \\ x_1^1 - x_1^3 \end{pmatrix}, \end{aligned} \tag{2.66}$$

and for the curvature we have

$$\begin{aligned} \kappa^2 = \kappa(0) &= \left. \frac{x_1^{'} x_2^{''} - x_2^{'} x_1^{''}}{\left[(x_1^{'})^2 + (x_2^{'})^2\right]^{\frac{3}{2}}} \right|_{s=0} \\ &\doteq 4\frac{(x_1^3 - x_1^1)(x_2^1 - 2x_2^2 + x_2^3) - (x_2^3 - x_2^1)(x_1^1 - 2x_1^2 + x_1^3)}{\left[(x_1^3 - x_1^1)^2 + (x_2^3 - x_2^1)^2\right]^{\frac{3}{2}}}. \end{aligned} \tag{2.67}$$

Combining Equations (2.66) and (2.67) gives the numerical approximation of $\mathbf{b}^2 = -\kappa\mathbf{n}^2$.

Since the quadratic solution is accurate to a third order, the same or higher order of approximation of the curvature and of the outer normal are required. In this case, we applied a fourth order Lagrangian interpolation polynomial. Therefore, the surface tension at, say $\mathbf{x}^3$, is found by a fit through its neighbors $\mathbf{x}^1$, $\mathbf{x}^2$, $\mathbf{x}^4$ and $\mathbf{x}^5$. Thus, the first and second derivatives with respect to s in Expressions (2.66) and (2.67) at node $\mathbf{x}^3$ $(s = 0)$ are obtained by the following approximation:

$$\begin{aligned} x_i^{'} &= \frac{1}{12}\Big(x_i^1 - 8x_i^2 + 8x_i^4 - x_i^5\Big), \\ x_i^{''} &= \frac{1}{12}\Big(- x_i^1 + 16x_i^2 - 30x_i^3 + 16x_i^4 - x_i^5\Big). \end{aligned} \tag{2.68}$$

Using these expressions, we then find the surface tension $\mathbf{b}^i$ at all nodes. Besides, the surface tension at any point of an arbitrary element may be obtained from a polynomial fit using Equation (2.19).

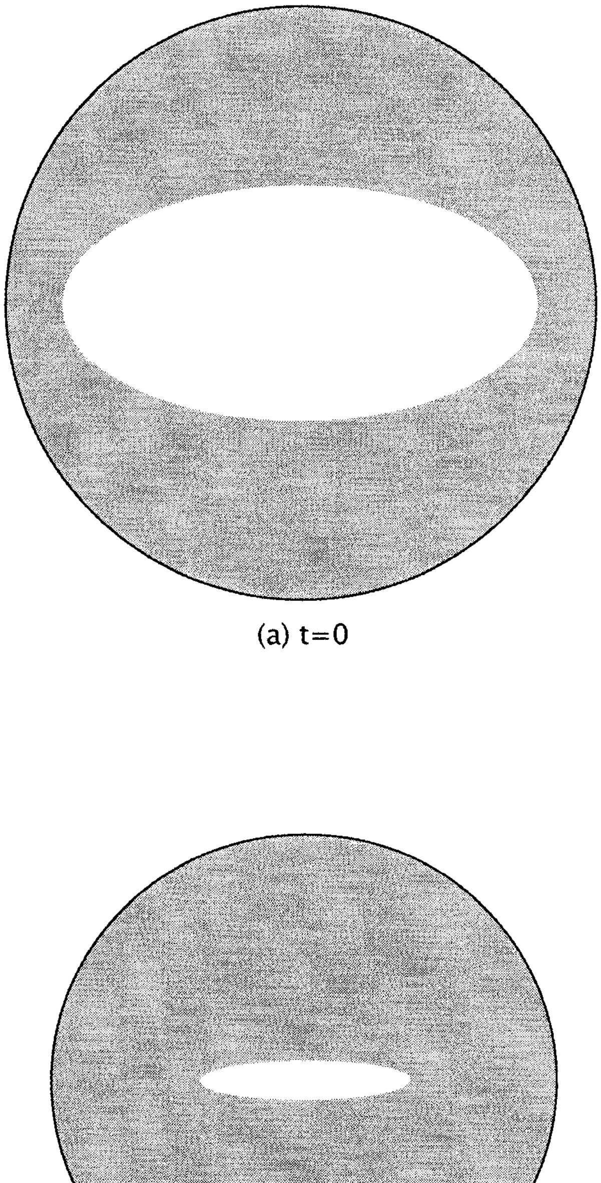
2.3.2 Problems Associated with the Approximation of the Curvature

As already mentioned, the problems arise when approximating the surface tension of a cusp region during the shape evolution. In the neighborhood of such a point, the nodes must lie close to each other since both curvature and velocity have a large variation in magnitude in relation to the relatively small curve length of that particular boundary part. During the deformation of such a cusp region those nodes are even coming closer together, which causes errors in the numerical approximation of the surface tension vector, especially for curvature. Hence, the computed surface tension can deviate considerably from the exact tension. These instabilities result from the following feedback effect: (1) small errors in the approximate collocation points produce local variations in the computed velocities of the nodes, (2) then causing uneven advancement of these points which yield large errors in the approximation. This process can lead to instabilities and wrong curves, and even a completely breakdown of numerical simulation.

The occurring oscillations in the approximate cusp curvature are due to numerical cancellation. This can be seen as follows: each collocation point, for both coordinate directions, has a discretization error. Regarding the origin of this error we can distinguish between the following three types of error source. A spatial error that comes from approximating the boundary by a polygon. Secondly, the error due to the usage of Gaussian quadrature formula to approximate the element integral. Finally, the time discretization error that arises from using a simple Forward Euler method or time integral method. Hence, we can say that the computed boundary deviates from the exact curve with a deviation of the order of the discretization error. When approximating the surface tension at nodal points where the mesh is fine, we loose some accuracy in the computation of both curvature and outer normal because the points are necessarily very close to each other. Then, both the numerator and denominator of the quotient of the approximate curvature are of the order the discretization error in magnitude. Hence the computed curvature can deviate considerably from the exact curvature.

This can be illustrated by considering the sintering of an elliptic hole. The initial shape is a circular disk with an elliptic hole centered at the origin. The radius of the disk is set equal to 1 and the ellipse is described, as an example, by $x_1^2/0.8^2+x_2^2/0.4^2 = 1$ ($a = 0.8$, $b = 0.4$). When $t = 0.4$, the outer side boundary becomes a smaller "circle" (not the exact circle), and the ellipse becomes smaller with long axis $a = 0.364$ and short axis $b = 0.0716$ (see Figure 2.7). At the beginning, all nodal points are equidistributed based on the arc length. If we do not redistribute these points, they become very close to each other while time is going, especially for the nodes in large curvature regions. In Table 2.1, we printed the node coordinates and calculated the approximate curvature by applying Equation (2.67). From this table, we can see that the nodes are very close to each other in the "cusp" region. Furthermore, we also printed the approximate curvature κ^i by applying Equation (2.67). In order to obtain these approximate curvature, we have to compute the following items

$$(x_j^{i+1} - x_j^i) \qquad \text{and} \qquad (x_j^i - x_j^{i-1}),$$



(a) t=0

(b) t=0.4

Figure 2.7 : The shrinkage of a circular disk with an elliptic hole: (a) the initial shape at $t = 0$, (b) the shape at $t = 0.4$.

occurring in the numerator, and

$$x_j^{i+1} - x_j^{i-1} = (x_j^{i+1} - x_j^i) + (x_j^i - x_j^{i-1}),$$

being part of the denominator. From this table, it is clear that the approximate curvatures deviate from the exact ones, and cannot reflect the real physical boundary. If we continue to use these values, the simulation will break down soon.

Table 2.1 : The collocation points of ellipse on the right with large curvature

i	x_1^i	x_2^i	$x_1^{i+1} - x_1^{i-1}$	$x_2^{i+1} - x_2^{i-1}$	$x_1^{i+1} - x_1^i$	$x_2^{i+1} - x_2^i$	κ
⋮	⋮	⋮					
15	0.287550	0.41958e-01					
16	0.333086	0.27006e-01	6.9313e-02	-2.8480e-02	4.5536e-02	-1.4952e-02	4.953
17	0.356863	0.13477e-01	2.9362e-02	-2.0857e-02	2.3777e-02	-1.3529e-02	16.900
18	0.362448	0.61485e-02	6.8520e-03	-1.1253e-02	5.5850e-03	-7.3286e-03	44.189
19	0.363715	0.22243e-02	1.3870e-03	-4.6719e-03	1.2670e-03	-3.9242e-03	32.926
20	0.363835	0.14767e-02	2.3900e-04	-1.7378e-03	1.2000e-04	-7.4765e-04	44.240
21	0.363954	0.48652e-03	1.5000e-04	-1.4614e-03	1.1900e-04	-9.9017e-04	64.041
22	0.363985	0.15300e-04	1.0000e-06	-9.3350e-04	3.1000e-05	-4.7122e-04	279.958
23	0.363955	-0.44698e-03	-1.4400e-04	-1.4517e-03	-3.0000e-05	-4.6228e-04	59.300
24	0.363841	-0.14365e-02	-2.3600e-04	-1.7315e-03	-1.1400e-04	-9.8947e-04	54.167
25	0.363719	-0.21784e-02	-1.3780e-03	-4.6450e-03	-1.2200e-04	-7.4198e-04	32.057
26	0.362463	-0.60814e-02	-6.7570e-03	-1.1177e-02	-1.2560e-03	-3.9030e-03	44.296
27	0.356962	-0.13355e-01	-2.9134e-02	-2.0784e-02	-5.5010e-03	-7.2736e-03	17.031
28	0.333329	-0.26865e-01	-6.9015e-02	-2.8473e-02	-2.3633e-02	-1.3510e-02	4.989
29	0.287947	-0.41828e-01					
⋮	⋮	⋮					

These points are from the inside elliptic hole at time t = 0.4. The cancellation in the computation of the "cusp" part curvature occurs when two successive nodal points in the two directions are subtracted (by computing the numerator of Equation (2.67)) and in the computation of $x_j^{i+1} - x_j^{i-1}$.

The above example shows that the mesh has to be checked after several time steps so that the collocation points do not get too close to each other in "cusp" region. On the other hand, it seems reasonable that these points lie close to each other because of the large variation of velocity and surface tension in such region. These two conflicting aspects work together in an algorithm that takes care of the mesh redistribution.

2.3.3 The Method of Mesh Redistribution

From the last section, we can see that the redistribution of nodal points is very important, otherwise we cannot obtain the correct simulation results. Van de Vorst proposed a method of redistributing the nodes [van de Vorst & Mattheij 1992]. However, this method is complicated and slow for most general cases. In this thesis, we put forward a simpler but effective method for node redistribution. Here, we assume that the boundary of the fluid region can be described by a parametric equation with respect to the arc length s, that is:

$$\mathbf{x}(s) \in \Gamma, \qquad 0 \leq s \leq s_e, \qquad \text{and} \qquad \mathbf{x}(0) = \mathbf{x}(s_e). \tag{2.69}$$

First, we derive the properties for the step-length between two successive nodes, which have to be fulfilled for all mesh points. Let $\mathbf{x}^{i-1} = \mathbf{x}(s_{i-1})$ and $\mathbf{x}^i = \mathbf{x}(s_i)$ be two given successive nodal points, and the distance between these points be $h_i = s_i - s_{i-1}$, which denotes the step-length. The next node $\mathbf{x}^{i+1}$ has to lie at a distance h_{i+1} from $\mathbf{x}_i$, such that the following conditions (2.70), (2.71) are satisfied. Let h_{min} and h_{max} be the two positive constants which give the bounds for step-length

$$h_{min} \leq h_{i+1} \leq h_{max}. \tag{2.70}$$

Furthermore, let the collocation points be uniformly (or quasi-uniformly) distributed on each boundary, thus:

$$h_{i+1} = h_i. \tag{2.71}$$

The above two equations are the basic idea of this redistribution. Next we discuss its

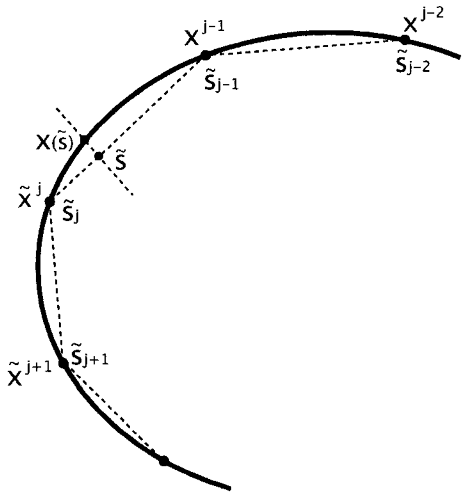


Figure 2.8 : The way to find new point $\mathbf{x}(\tilde{s})$.

numerical implementation.

It is reasonable that we start with an already discretized boundary. It is referred to as *old* nodes $\tilde{\mathbf{x}}^i$ $(1 \leq i \leq N)$. For a practical application of the above idea, we define a discretized version of the arc length $\tilde{s}$, which defines how a point $\mathbf{x}(\tilde{s})$ is to be found.

For the numerical application, we use the straight-line distance between two successive old nodes for a discretized arc length, given by:

$$\tilde{s}_i = \tilde{s}_{i-1} + \|\tilde{\mathbf{x}}^i - \tilde{\mathbf{x}}^{i-1}\|, \qquad (2 \leq i \leq N+1), \tag{2.72}$$

where $\tilde{s}_1 = 0$ and $\tilde{\mathbf{x}}^{N+1} = \tilde{\mathbf{x}}^1$. A nodal point with a certain arc length, say $\tilde{s}$, is found by fitting a quadratic polynomial through three successive nodes, say $\tilde{\mathbf{x}}^{j-1}$, $\tilde{\mathbf{x}}^j$, and

$\tilde{\mathbf{x}}^{j+1}$. The number j is taken such that

$$|\tilde{s}_j - \tilde{s}| \leq \min\Big(|\tilde{s}_{j-1} - \tilde{s}|,\ |\tilde{s}_{j+1} - \tilde{s}|\Big). \tag{2.73}$$

Assuming that $\tilde{s} \leq \tilde{s}_j$, the node $\mathbf{x}(\tilde{s})$ is at the intersection of this quadratic polynomial with the straight line in the direction normal to the straight line through the nodes '$j-1$' and 'j', and through the point on this line that has a distance $\tilde{s}_j - \tilde{s}$ with the node $\tilde{\mathbf{x}}^j$ (see Figure 2.8). After that, the old point 'j' is replaced by the new point $\mathbf{x}(\tilde{s})$.

Another point of concern that occurs in the case of simulation of a multiply connected domain is the possibility of the vanishing holes. The inner holes area may not become too small in order to avoid errors in the generation of this hole curve. Therefore, we have to check the inside hole area at each step. If the hole area is smaller than a certain value (normalized value: 1×10^{-5}), we let the hole vanish completely.

Chapter 3

Time Integration

When the movement of the boundary is modeled by considering the boundary curve as material points, the trajectories of these points can be described by a Lagrangian representation of the velocity. After solving Equation (2.61), we obtain $2N$ nonlinear ODEs. Here, we formally write these ODEs as follows:

$$\frac{d\boldsymbol{x}}{dt} = \mathcal{H}^{-1}(\boldsymbol{x})\mathcal{G}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x}). \tag{3.1}$$

In this chapter, we consider the numerical integration of this system of ODEs. Since the Backward Difference Formula (BDF) is suitable for both *stiff* and *non-stiff* problems, here we use a variable step, variable order BDF scheme for the system (3.1). Firstly, we discuss the calculation of the Jacobian, then we give the detail of the numerical implementation of integration of the system of ODEs (3.1).

3.1 Approximation of the Jacobian

Since the BDF method is an implicit linear multistep method, the corrector equation has to be solved by some iteration method. In our simulation a modified Newton iteration method is applied. This means that the Jacobian matrix $\mathscr{J}(\boldsymbol{x})$ of Equation

(3.1) is needed, as,

$$\mathscr{J}(\boldsymbol{x}) = \frac{\partial}{\partial \boldsymbol{x}}\left[\mathcal{H}^{-1}(\boldsymbol{x})\mathcal{G}(\boldsymbol{x})\boldsymbol{b}(\boldsymbol{x})\right]. \tag{3.2}$$

From the derivation of the formulation given in the previous sections, it is impossible to derive an analytical expression for the Jacobian matrix. A numerical approximation of the exact Jacobian (3.2) would be extremely time consuming. However, it is not necessary to have the exact Jacobian, since the BDF method uses a modified Newton method, i.e., the same Jacobian is used in subsequent Newton iteration, and for several time integral steps. Therefore, we will use the approximate form of the Jacobian. The derivation of this approximation is outlined for a simply connected domain in the remaining part of this section.

Let $\hat{\boldsymbol{x}}_{j,k}$ the vector of all boundary nodes whereby the nodal point 'j' is perturbed by a small value, say ξ ($\xi \ll 1$) in the k^{th}-direction ($k = 1,\ 2$), thus:

$$\hat{\boldsymbol{x}}_{j,k} = \left(x_1^1,\ x_2^1,\ \cdots,x_1^j+\xi,\ x_2^j,\ \cdots,x_1^N,\ x_2^N\right)^T.$$

Furthermore, we have the solution $\hat{\boldsymbol{v}}_{j,k}$ of the system (3.1) for this perturbed boundary

$$\mathcal{H}(\hat{\boldsymbol{x}}_{j,k})\hat{\boldsymbol{v}}_{j,k} = \mathcal{G}(\hat{\boldsymbol{x}}_{j,k})\boldsymbol{b}(\hat{\boldsymbol{x}}_{j,k}).$$

By applying a Taylor expansion around $\boldsymbol{x}$ of these quantities up to the first order, we have

$$\left(\mathcal{H}+\xi\delta\mathcal{H}_{j,k}\right)\left(\boldsymbol{v}+\xi\delta\boldsymbol{v}_{j,k}\right) = \left(\mathcal{G}+\xi\delta\mathcal{G}_{j,k}\right)\left(\boldsymbol{b}+\xi\delta\boldsymbol{b}_{j,k}\right), \tag{3.3}$$

where both $\delta\mathcal{H}_{j,k}$ and $\delta\mathcal{G}_{j,k}$ are the sparse matrices that contains the derivatives of matrix elements with respect to x_k^j. The non-zero elements of these matrices have row indices '$2j-1$' and '$2j$' and column indices from '$2j-1-p$' to '$2j+p$', where p is equal to 2 in the case of linear elements or when node 'j' is the mid-point of a quadratic element; p is equal to 4 when quadratic elements are used and node 'j' is one of the corners of the element. The vector $\delta\boldsymbol{b}_{j,k}$ has non-zero elements for the same indices as the column of the above matrices. Thus, by using the exact solution (3.1), and omitting higher order terms in Equation (3.3), we obtain the following first order approximation for the $(2j-k)^{\text{th}}$ column of $\mathscr{J}$,

$$\delta \boldsymbol{v}_{j,k} \doteq \mathcal{H}^{-1}\left(\mathcal{G}\;\delta \boldsymbol{b}_{j,k} + \delta\mathcal{G}_{j,k}\;\boldsymbol{b} - \delta\mathcal{H}_{j,k}\;\boldsymbol{v}\right). \tag{3.4}$$

The above approximation is not expensive to compute when compared to the numerical calculation of the exact value of the Jacobian. Because of the sparsity of the derivative matrices $\delta\mathcal{H}_{j,k}$, $\delta\mathcal{G}_{j,k}$ and of the vector $\delta\boldsymbol{b}_{j,k}$, the computational costs for obtaining the Jacobian $\mathscr{J}$ is of the order of more than ten times the cost of assembling the system of equations, where we note that assembling the system of equations is the most expensive part of the solution process. Therefore, this kind of approximation is not quite satisfactory.

If the terms $\delta\mathcal{H}_{j,k}\boldsymbol{v}$, and $\delta\mathcal{G}_{j,k}\boldsymbol{b}$ are omitted, the costs of computing the Jacobian $\mathscr{J}$ will be considerably reduced. Besides, this makes the computation very simple too. It is necessary to analyze and quantify the contribution of each term in the vector $\mathcal{G}\;\delta\boldsymbol{b}_{j,k} + \delta\mathcal{G}_{j,k}\;\boldsymbol{b} - \delta\mathcal{H}_{j,k}\;\boldsymbol{v}$ for reducing the computation time and complexity. However, this task is quite complex. Thus, we just give the final results; the detail of this analysis is given in [van de Vorst 1994]:

$$\begin{aligned}
\delta\mathcal{H}_{j,k}\;\boldsymbol{v} &= \mathcal{O}\left(\frac{1}{L_p}\right), \\
\delta\mathcal{G}_{j,k}\;\boldsymbol{b} &= \mathcal{G}\mathcal{O}\left(\frac{1}{e_{02}^2}+\frac{1}{e_{13}^2}\right), \\
\mathcal{G}\;\delta\boldsymbol{b}_{j,k} &= \mathcal{G}\mathcal{O}\left(\frac{e_{12}}{e_{02}^3}+\frac{e_{12}}{e_{13}^3}\right),
\end{aligned} \tag{3.5}$$

and

$$\delta\mathcal{G}_{j,k}\;\boldsymbol{b} - \delta\mathcal{H}_{j,k}\;\boldsymbol{v} = \mathcal{O}\left(\frac{e_{12}}{e_{02}^2}+\frac{e_{12}}{e_{13}^2}\right), \tag{3.6}$$

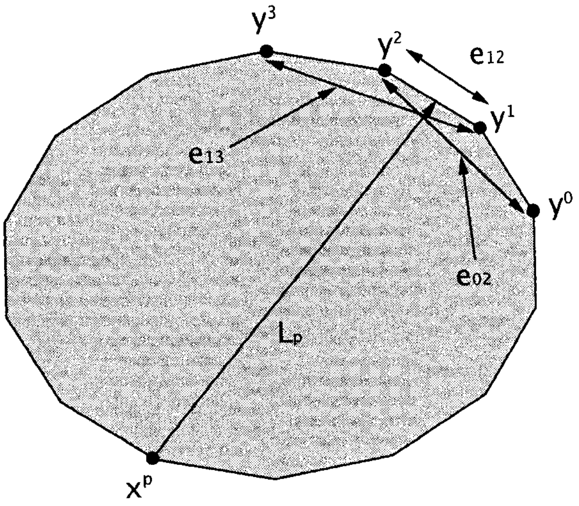


Figure 3.1 : Notation of the defined lengths when the collocation point $\mathbf{x}^{\mathbf{p}}$ is not a node of the considered element.

where the meanings of e_{02}, e_{12}, e_{13} and L_p can be found in Figure 3.1.

From the analysis given above and Equation (3.4), it follows that we can approximate the $(2j-k)^{\text{th}}$ column $\delta \boldsymbol{v}_{j,k}$ of the Jacobian $\mathscr{J}$ by:

$$\delta \boldsymbol{v}_{j,k} \doteq \mathcal{H}^{-1}\mathcal{G}\ \delta \boldsymbol{b}_{j,k}. \tag{3.7}$$

This approximation is satisfactory for treating both stiff and non-stiff problems. In the next section, we use this approximate Jacobian to calculate an example system with quadratic element.

3.2 Numerical Implementation of the BDF Method

After solving Equation (2.61), one obtains the nonlinear ODEs. Here we denote the ODEs by:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{v}. \tag{3.8}$$

In this section, firstly we briefly derive the so-called predictor-corrector formulation of the BDF method. Then, the estimate of the local truncation error will be discussed. Finally, we will give the way of how controlling the time step size and method order.

3.2.1 Predictor-Corrector Formulation

To find the solution of ODEs at the time $t = t_i$, the original q^{th}-order BDF method needs the actual position values of the boundary nodes at previous time steps $t_{i-1}, \cdots, t_{i-q}$ and the velocity of the boundary at $t = t_{i-1}$ as well, thus:

$$\boldsymbol{x}_i = \sum_{j=1}^{q} \alpha_j \boldsymbol{x}_{i-j} + h_i \beta_0 \boldsymbol{v}_{i-1} , \tag{3.9}$$

where α_j, β_0 are certain constants [Gear 1971] [Hairer & G.Wanner 1996]. When this q^{th}-order BDF method is expressed in the so-called *Nordsieck representation*, the boundary at $t = t_{i-1}$, and the first till the q^{th} derivatives (with respect to time t) of the boundary are required. The Nordsieck vector at time $t = t_{i-1}$, say $\mathbf{z}_{i-1}$, is expressed by

$$\mathbf{z}_{i-1} = \left(\boldsymbol{x}_{i-1},\ h_i \dot{\boldsymbol{x}}_{i-1},\ \frac{h_i^2}{2\,!} \ddot{\boldsymbol{x}}_{i-1},\ \ldots,\ \frac{h_i^q}{q\,!} \boldsymbol{x}_{i-1}^{(q)} \right) , \tag{3.10}$$

where h_i is the applied time step. With this Nordsieck vector, we can easily obtain the predictor of the next time step $t_i = t_{i-1} + h_i$, thus:

$$\boldsymbol{x}_i = \left(\boldsymbol{x}_{i-1} + h_i \dot{\boldsymbol{x}}_{i-1} + \frac{h_i^2}{2!} \ddot{\boldsymbol{x}}_{i-1} + \cdots + \frac{h_i^q}{q\,!} \boldsymbol{x}_{i-1}^{(q)} \right) . \tag{3.11}$$

From Equation (3.10), we can see the advantage of this representation, namely, when the step size h_i is changed, the Nordsieck vector for this new step size is easy to find. In this thesis, the implementation of BDF is based on the Nordsieck representation with a given step size. Next, we briefly outline the derivation of the formulation that will be applied.

Applying the appropriate matrix transformation [Gear 1971], we obtain the predictor equation of $\mathbf{z}_i$ at time $t = t_i$, say $\mathbf{z}_i^{[0]}$,

$$\mathbf{z}_i^{[0]} = \mathbf{z}_{i-1}\mathbb{A}, \tag{3.12}$$

where $\mathbb{A}$ is given by

$$\mathbb{A} = \begin{pmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 2 & 1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & 3 & 3 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & q-2 & \frac{(q-2)(q-3)}{2\,!} & \frac{(q-2)(q-3)(q-4)}{3\,!} & \cdots & 1 & 0 & 0 \\ 1 & q-1 & \frac{(q-1)(q-2)}{2\,!} & \frac{(q-1)(q-2)(q-3)}{3\,!} & \cdots & q-1 & 1 & 0 \\ 1 & q & \frac{q(q-1)}{2\,!} & \frac{q(q-1)(q-2)}{3\,!} & \cdots & \frac{q(q-1)}{2\,!} & q & 1 \end{pmatrix}. \tag{3.13}$$

Correspondingly, the corrector equation of $\mathbf{z}_i$ can be written as:

$$\mathbf{z}_i^{[m+1]} = \mathbf{z}_i^{[m]} + \mathbf{P}^{-1}\boldsymbol{g}\left(\boldsymbol{x}_i^{[m]}\right)\mathcal{L}, \tag{3.14}$$

where $\mathbf{z}_i^{[m]}$, the Nordsieck history matrix on the m^{th} iteration, is given by:

$$\mathbf{z}_i^{[m]} = \left(\boldsymbol{x}_i^{[m]},\ h_i\dot{\boldsymbol{x}}_i^{[m]},\ \frac{h_i^2}{2\,!}\ddot{\boldsymbol{x}}_i^{[m]},\ \ldots,\ \frac{h_i^q}{q\,!}\boldsymbol{x}_i^{[m](q)}\right), \tag{3.15}$$

the $2N \times 2N$ matrix $\mathbf{P}$ is given by:

$$\mathbf{P} = \mathbf{I} - h_i\beta_0\mathscr{J}, \tag{3.16}$$

where the vector $\boldsymbol{g}$ corresponding to the position $\boldsymbol{x}_i^{[m]}$ is defined by:

$$\boldsymbol{g}\left(\boldsymbol{x}_i^{[m]}\right) = h_i \boldsymbol{v}\left(\boldsymbol{x}_i^{[m]}\right) - h_i \frac{d\boldsymbol{x}_i^{[m]}}{dt}\bigg|_{t_i}, \tag{3.17}$$

and the vector $\mathcal{L}$ is given as:

$$\mathcal{L} = \left(l_0,\ l_1,\ \cdots,\ l_q\right). \tag{3.18}$$

In Equation (3.16), $\mathbf{I}$ is the identity matrix, and $\mathscr{J}$ is the approximate Jacobian matrix expressed by Equation (3.7). The coefficients β_0 and l_i in Equation (3.16) and (3.18) are derived in [Gear 1971]. However, to facilitate estimation of the local truncation error, a different iteration procedure than that given by Equation (3.14) is used. The new formulation, $\mathbf{z}_i^{[m+1]}$ is written as:

$$\mathbf{z}_i^{[m+1]} = \left(\mathbf{z}_i^{[m+1]} - \mathbf{z}_i^{[m]}\right) + \left(\mathbf{z}_i^{[m]} - \mathbf{z}_i^{[m-1]}\right) + \cdots + \left(\mathbf{z}_i^{[1]} - \mathbf{z}_i^{[0]}\right) + \mathbf{z}_i^{[0]},$$

or

$$\mathbf{z}_i^{[m+1]} = \mathbf{z}_i^{[0]} + \sum_{j=0}^{m} \left(\mathbf{z}_i^{[j+1]} - \mathbf{z}_i^{[j]}\right). \tag{3.19}$$

Substituting the difference $\mathbf{z}_i^{[j+1]} - \mathbf{z}_i^{[j]}$ obtained from Equation (3.14) into Equation (3.19) yields:

$$\mathbf{z}_i^{[m+1]} = \mathbf{z}_i^{[0]} + \sum_{j=0}^{m} \mathbf{P}^{-1} g\left(\boldsymbol{x}_i^{[j]}\right) \mathcal{L} = \mathbf{z}_i^{[0]} + \boldsymbol{e}_i^{[m+1]} \mathcal{L}, \tag{3.20}$$

where $\boldsymbol{e}_i^{[m+1]}$ is defined as:

$$\boldsymbol{e}_i^{[m+1]} = \sum_{j=0}^{m} \mathbf{P}^{-1} \boldsymbol{g}\left(\boldsymbol{x}_i^{[j]}\right). \tag{3.21}$$

It is clear from the above equation that

$$\boldsymbol{e}_i^{[m+1]} = \boldsymbol{e}_i^{[m]} + \mathbf{P}^{-1}\boldsymbol{g}\left(\boldsymbol{x}_i^{[m]}\right). \tag{3.22}$$

By using Equation (3.20), the vector $\boldsymbol{g}$ in Equation (3.17) can be written as:

$$\boldsymbol{g}\left(\boldsymbol{x}_i^{[m]}\right) = h_i\boldsymbol{v}\left(\boldsymbol{x}_i^{[m]}\right) - h_i\dot{\boldsymbol{x}}_i^{[0]} - \boldsymbol{e}_i^{[m]}. \tag{3.23}$$

Therefore, the predictor-corrector formulation for the BDF method can be summarized as follows:

Predictor:

$$\begin{aligned} \mathbf{z}_i^{[0]} &= \mathbf{z}_{i-1}\mathbb{A}, \\ \boldsymbol{e}_i^{[0]} &= 0, \end{aligned} \tag{3.24}$$

Corrector:

$$\begin{aligned} \boldsymbol{g}\left(\boldsymbol{x}_i^{[m]}\right) &= h_i\boldsymbol{v}\left(\boldsymbol{x}_i^{[m]}\right) - h_i\dot{\boldsymbol{x}}_i^{[0]} - \boldsymbol{e}_i^{[m]}, \\ \boldsymbol{e}_i^{[m+1]} &= \boldsymbol{e}_i^{[m]} + \mathbf{P}^{-1}\boldsymbol{g}\left(\boldsymbol{x}_i^{[m]}\right), \\ \boldsymbol{x}_i^{[m+1]} &= \boldsymbol{x}_i^{[m]} + l_0\boldsymbol{e}_i^{[m+1]}, \end{aligned} \tag{3.25}$$

Final Results:

$$\begin{aligned} \boldsymbol{e}_i &= \boldsymbol{e}_i^{[M]}, \\ \mathbf{z}_i &= \mathbf{z}_i^{[0]} + \boldsymbol{e}_i\boldsymbol{\mathcal{L}}, \end{aligned} \tag{3.26}$$

where $m = 0,\ 1,\ \cdots,\ M-1$, and M is the maximum number of iterations.

3.2.2 Estimation of the Local Truncation Error

The local truncation error is defined to be the amount by which the exact solution $\boldsymbol{x}(t)$ of the ODE system fails to satisfy the difference equations of the numerical scheme. That is, for linear multistep methods (Equation (3.9)), the local truncation error vector $\boldsymbol{d}_i$ at t_i is the residual in the difference formula when the approximations $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ are replaced by the exact solution and its derivative.

In particular, for the BDF method of order q, the local truncation error is given by:

$$\boldsymbol{d}_i = C_{q+1} h_i^{q+1} \boldsymbol{x}^{(q+1)}(t_i) + \mathcal{O}(h_i^{q+2}), \tag{3.27}$$

where the terms of $C_{q+1}(= 1/(q+1))$ and $C_{q+1} h_i^{q+1} \boldsymbol{x}^{(q+1)}(t_i)$ are called the error constant and the principal truncation error respectively [Lambert 1973].

The $(q+1)^{\text{th}}$ derivative at t_i, say $\boldsymbol{x}^{(q+1)}(t_i)$, is estimated as follows: at each time step, the method updates the Nordsieck history matrix $\mathbf{z}_i$:

$$\mathbf{z}_i = \left(\boldsymbol{x}_i,\ h_i \dot{\boldsymbol{x}}_i,\ \frac{h_i^2}{2\,!} \ddot{\boldsymbol{x}}_i,\ \ldots,\ \frac{h_i^q}{q\,!} \boldsymbol{x}_i^{(q)} \right). \tag{3.28}$$

For the method of order q, the last column of $\mathbf{z}_i$, say $z_i(q)$, contains the vector $h_i^q \boldsymbol{x}_i^{(q)}/q!$, which is the approximation to the $h_i^q q!\ \boldsymbol{x}^{(q)}(t_i)/q!$. By using the mean value theorem for the derivatives, we obtain the difference between $z_i(q)$ and that in predictor [Radhakrishnan & Hindmarsh 1993], say $z_i^{[0]}(q)$, i.e.,

$$z_i(q) - z_i^{[0]}(q) = \frac{h_i^{q+1}}{q\,!} \boldsymbol{x}_i^{(q+1)} + \mathcal{O}\left(h_i^{q+2}\right). \tag{3.29}$$

However, Equation (3.26) gives the following expression for $z_i(q) - z_i^{[0]}(q)$:

$$z_i(q) - z_i^{[0]}(q) = l_q \boldsymbol{e}_i\,. \tag{3.30}$$

Omitting the higher order of $\mathcal{O}(h_i^{q+2})$ in Equation (3.29) and comparing it to Equation (3.30), one obtains the following approximation:

$$h_i^{q+1}\boldsymbol{x}_i^{(q+1)} \doteq q!\, l_q\, \boldsymbol{e}_i \ , \tag{3.31}$$

Substituting the above equation into Equation (3.27), and neglecting higher-order terms gives the following estimate for $\boldsymbol{d}_i$:

$$\boldsymbol{d}_i = C_{q+1}\, q!\, l_q \boldsymbol{e}_i \ . \tag{3.32}$$

If we specify the error tolerance for each nodal point, i.e., we have an error tolerance vector $ET = (et_1, et_2, \cdots, et_{2N})$, the condition for the solution $\boldsymbol{x}_i$ to be acceptable can be written as:

$$\left\|\boldsymbol{d}_i\right\| \equiv \sqrt{\frac{1}{2N}\sum_{n=1}^{2N}\left(\frac{d_{i,n}}{et_n}\right)^2} \leq 1, \tag{3.33}$$

where $d_{i,n}$ is the n^{th} component of the vector $\boldsymbol{d}_i$, and $\| \bullet \|$ denotes the weighted root-mean-square (RMS) norm. From Equation (3.32), the above condition can be written as:

$$\left\|\boldsymbol{e}_i\right\| \equiv \sqrt{\frac{1}{2N}\sum_{n=1}^{2N}\left(\frac{e_{i,n}}{et_n}\right)^2} \leq \frac{1}{C_{q+1}\, q!\, l_q}. \tag{3.34}$$

In the next sub-section, we use Equation (3.33) or (3.34) to discuss the step size and the method order selection.

3.2.3 Step Size and Method for Selecting the Order of the Scheme

In order to minimize complications associated with the order and the step size selection, the new order q' is restricted to the values $q-1,\ q,$ and $q+1$, where q is the current order. For the each new order q', the step size $h'(q')$ that satisfies exactly the local error bound is obtained by assuming that the highest derivative remains constant. Based solely on efficiency considerations, the method order that produces the largest h' is used on the next time. For the convenience of discussion, we define a quantity factor $\mathcal{D}_q$ corresponding to the method order q:

$$\mathcal{D}_q = \|\boldsymbol{e}_i\|/\tau(q), \tag{3.35}$$

where $\tau(q) = 1/(C_{q+1}\, q!\, l_q)$. Thus, the accuracy condition (3.34) reduces to

$$\mathcal{D}_q \leq 1. \tag{3.36}$$

For the case $q' = q$, the new step size $h'(q)$ is computed by setting $\mathcal{D}_q(h') = 1$ (see Equation (3.36)), so that the accuracy condition is satisfied exactly. From Equation (3.27), one obtains

$$\frac{1}{\mathcal{D}_q} = \left(\frac{h'(q)}{h_i}\right)^{q+1}, \tag{3.37}$$

or the ratio of new step size to the old one, say α,

$$\alpha_{same} \equiv \frac{h'(q)}{h_i} = \left(\frac{1}{\mathcal{D}_q}\right)^{\frac{1}{q+1}}. \tag{3.38}$$

The subscript "*same*" indicates that the same order will be used on the next step.

For the case $q' = q-1$, the local truncation error vector $\boldsymbol{d}_i(q-1)$ is of order q, where the variable '$q-1$' indicates the method order is to estimated, i.e.,

$$\boldsymbol{d}_i(q-1) = C_q\, h_i^q\, \boldsymbol{x}^{(q)}(t_i), \tag{3.39}$$

where $C_q = 1/q$. The last column of $\mathbf{z}_i$, say $z_i(q)$ contains the vector $h_i^q \boldsymbol{x}^{(q)}/q!$ (see Equation (3.28)), so that $\boldsymbol{d}_i(q-1)$ is easily calculated. By using the RMS norm equation (3.33), the error test for $q' = q-1$ becomes

$$\mathcal{D}_{q-1} \equiv \sqrt{\frac{1}{2N}\sum_{n=1}^{2N}\left(\frac{C_q h_i^q x_{i,n}^{(q)}}{et_n}\right)^2} = \frac{1}{\tau(q-1)}\sqrt{\frac{1}{2N}\sum_{n=1}^{2N}\left(\frac{z_{i,n}(q)}{et_n}\right)^2} \leq 1, \tag{3.40}$$

where the coefficient $\tau(q-1)$ is defined as $1/(C_q\, q!)$, and $z_{i,n}(q)$ is the n^{th} element of $z_i(q)$.

The step size $h'(q-1)$ for the method order $q-1$ is also obtained by setting $\mathcal{D}_{q-1}(h') = 1$. Because $\boldsymbol{d}_i(q-1)$ varies as h_i^q, the resulting step size ratio can be given with the same procedure as that which was used for the case $q' = q$, i.e.,

$$\alpha_{down} \equiv \frac{h'(q-1)}{h_i} = \left(\frac{1}{\mathcal{D}_{q-1}}\right)^{\frac{1}{q}}. \tag{3.41}$$

The subscript "*down*" denotes that the order is to be reduced by 1.

For the case $q' = q+1$, the local truncation error $\boldsymbol{d}_i(q+1)$ is of order $q+2$ and is given by

$$\boldsymbol{d}_i(q+1) = C_{q+2} h_i^{q+2} \boldsymbol{x}^{(q+2)}(t_i), \tag{3.42}$$

where $C_{q+2} = 1/(q+2)$. This case is more difficult than the previous cases because Equation (3.42) involves the derivative of order $q+2$. By using the mean value theorem for derivative, we can obtain the difference of $l_q\boldsymbol{e}$ between last two steps (see

Equation (3.31)), i.e.,

$$\begin{aligned} l_q\nabla \boldsymbol{e}_i \equiv l_q\boldsymbol{e}_i - l_q\boldsymbol{e}_{i-1} &\cong \frac{h_i^{q+1}}{q\,!}\boldsymbol{x}_i^{(q+1)} - \frac{h_i^{q+1}}{q\,!}\boldsymbol{x}_{i-1}^{(q+1)} \\ &= \frac{h_i^{q+2}}{q\,!}\boldsymbol{x}_i^{(q+2)} + \mathcal{O}(h_i^{q+3}). \end{aligned} \tag{3.43}$$

Hence the error test for $q' = q + 1$ becomes

$$\mathcal{D}_{q+1} \equiv \sqrt{\frac{1}{2N}\sum_{n=1}^{2N}\left(\frac{C_{q+2}\,q!\,l_q\nabla e_{i,n}}{et_n}\right)^2} = \frac{1}{\tau(q+1)}\sqrt{\frac{1}{2N}\sum_{n=1}^{2N}\left(\frac{\nabla e_{i,n}}{et_n}\right)^2} \leq 1, \tag{3.44}$$

where we have again used the RMS norm, and $\nabla e_{i,n}$ is the n^{th} component of vector $\nabla \boldsymbol{e}_i$. The test coefficient $\tau(q + 1)$ is defined as $1/(C_{q+2}\, l_q\, q!)$. In order to obtain the new step size $h'(q + 1)$ for order $q + 1$, we follow the same procedure as for $h'(q)$ or $h'(q - 1)$. The resulting ratio is given by:

$$\alpha_{up} \equiv \frac{h'(q+1)}{h_i} = \left(\frac{1}{\mathcal{D}_{q+1}}\right)^{\frac{1}{q+2}}. \tag{3.45}$$

The subscript "*up*" indicates that the order has to be increased by 1.

After a suitable ratio α has been computed, the new step size h' for next is calculated:

$$h' = \alpha h_i\,. \tag{3.46}$$

Based on the stability and accuracy considerations, after the step size h' and method order q are selected, we use this step size to calculate for $q + 1$ times. According to our test, we take the maximum order $q = 3$. Anyway, if the step size and/or the method is changed, the Nordsieck history matrix $\mathbf{z}_i$ and iteration matrix $\mathbf{P}$ should be modified. The detail about this modification can be found in [Gear 1971] and [Radhakrishnan & Hindmarsh 1993].

In order to continue after a node redistribution with the same order and step size as before redistribution, the Nordsieck vector for those new nodes are required. However, it is complicated and time consuming to calculate the first until q^{th} derivatives with respect to time t. Besides, an interpolation error would be introduced. Hence, we restart time integration without further information after a node redistribution.

3.3 Numerical Results and Discussion

Herein, we present some numerical results obtained by using the time integration method, and they are compared with the analytical results to demonstrate the correctness of what is described in the previous section. To do so, we take a circular disk with a circular hole as an example (see Figure 3.2, outer radius $R_o = 1$, inner radius $R_i = 0.5$ at $t = 0$).

In this example, firstly we study the behavior of the time step size. According to the general properties of sintering problem, the evolving velocity is expected to decrease as time advances. So, if we specify a certain tolerance value for the whole evolution process, the step size should be increased with time going. In Figure 3.3, we plotted the relation between step size h and time t, ignoring the node redistribution. From this figure we can see this property.

Next, we show the validity of the approximated evaluation of the Jacobian matrix. For the error control in our simulation, we use a global absolute tolerance equal to 5×10^{-6}. In Figure 3.4, we plotted the absolute error development between the exact analytical solution and the numerical solution by using a approximate Jacobian matrix.

From these two Figures, we can see that the difference between numerical solutions and exact solutions are of the order of the time integration tolerance.

More specifically, we print some steps after a node redistribution (rd). In our simulation, we restart the time integration after a node redistribution without further information; i.e., the starting order of the method is equal to 1, and the initial step

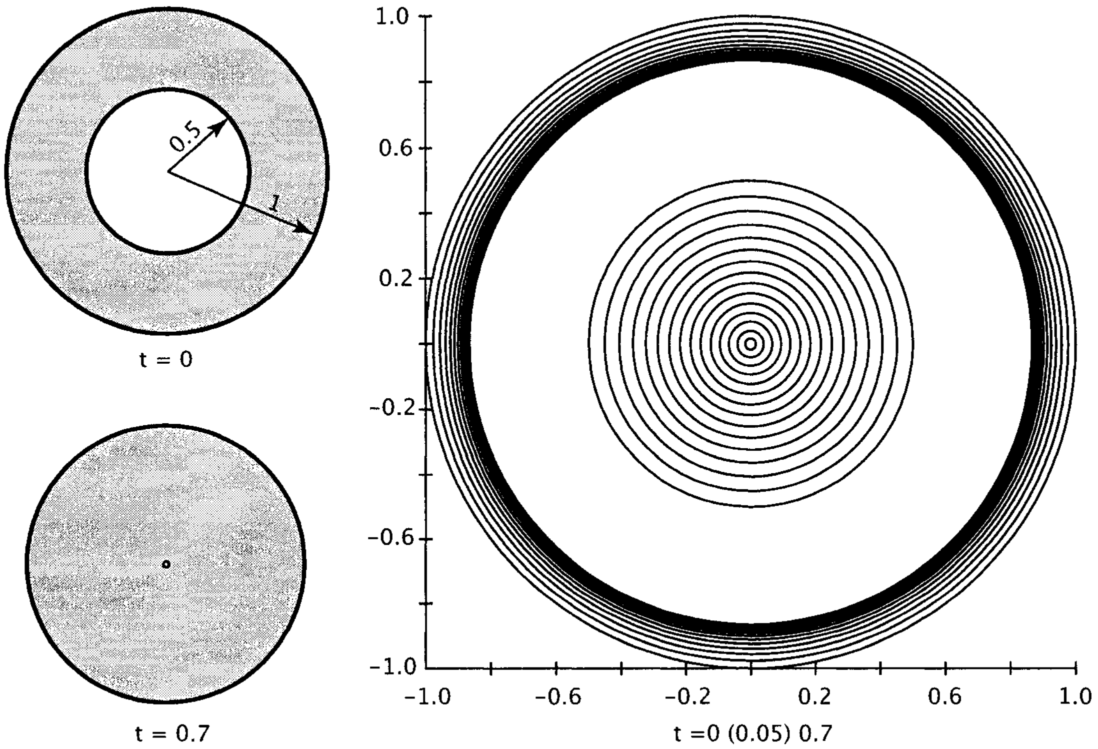


Figure 3.2 : The shrinkage of a circular disk with a circular hole centered at the origin.

size is given (=0.001). By '*Area*' we denote the total area of the fluid domain (which has to be preserved during the evolution). The relative change of this total area compared to the original shape is also printed.

From the above table, it is clear that we can obtain very precise results with time integration method in simulation, even in the late stage of the simulation, for which the condition number of matrix $\mathscr{A}$ in Equation (2.61) becomes large. On the other hand, the step size can be larger than that used in a simple step method. However, when the size of the matrix $\mathscr{A}$ is large ($> 300 \times 300$), it takes a lot of time to calculate the Jacobian matrix $\mathscr{J}$, the corrector matrix $\mathbf{P}^{-1}$ (see Equation (3.14)), and the Newton correcting iteration. For example, if the total number of nodal points of the boundaries is 200 (the size of $\mathscr{A} = 400 \times 400$), and the Power Mac

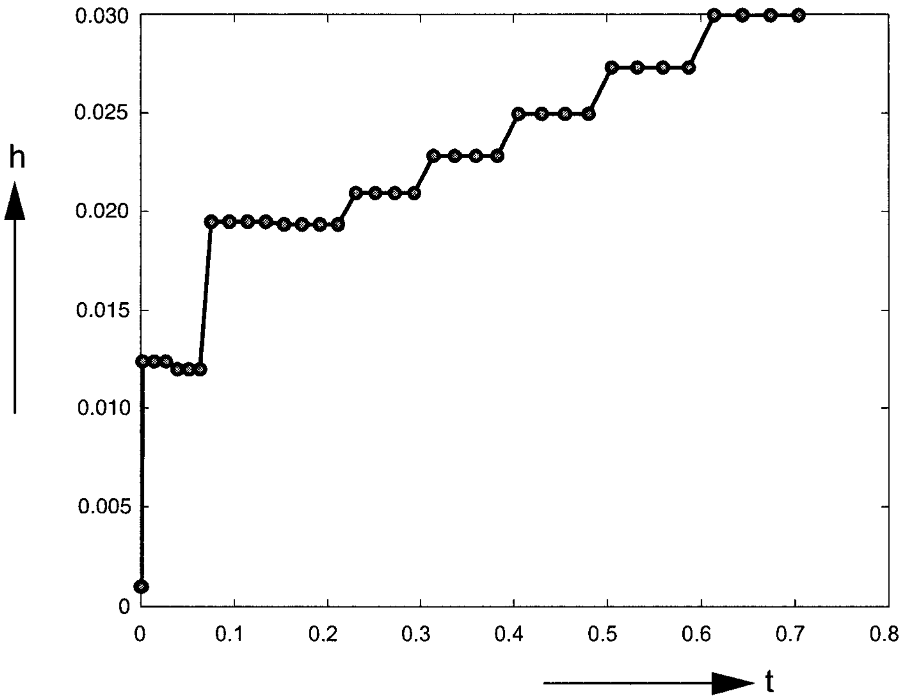


Figure 3.3 : The behavior of the step size in the evolution process.

G4 computer system (with CPU of 733MHz and RAM of 640MB) is used, it takes about 9 seconds for assembling the system linear equations, 3 seconds for getting the velocity solution with Bi-CGStab iteration method (see Appendix C), and 200 seconds for the calculation of matrices $\mathscr{J}$, $\mathbf{P}^{-1}$ and Newton correcting iteration. The time integration method needs more than 20 times the cost of assembling the system equations. Therefore, if the system is not too big (the size of $\mathscr{A} < 300 \times 300$), we can chose a time integration method to obtain more precise results; for larger system, we would prefer a simple step method (simple Forward Euler method) with a small step size (0.0005 — 0.005) to obtain simulation results.

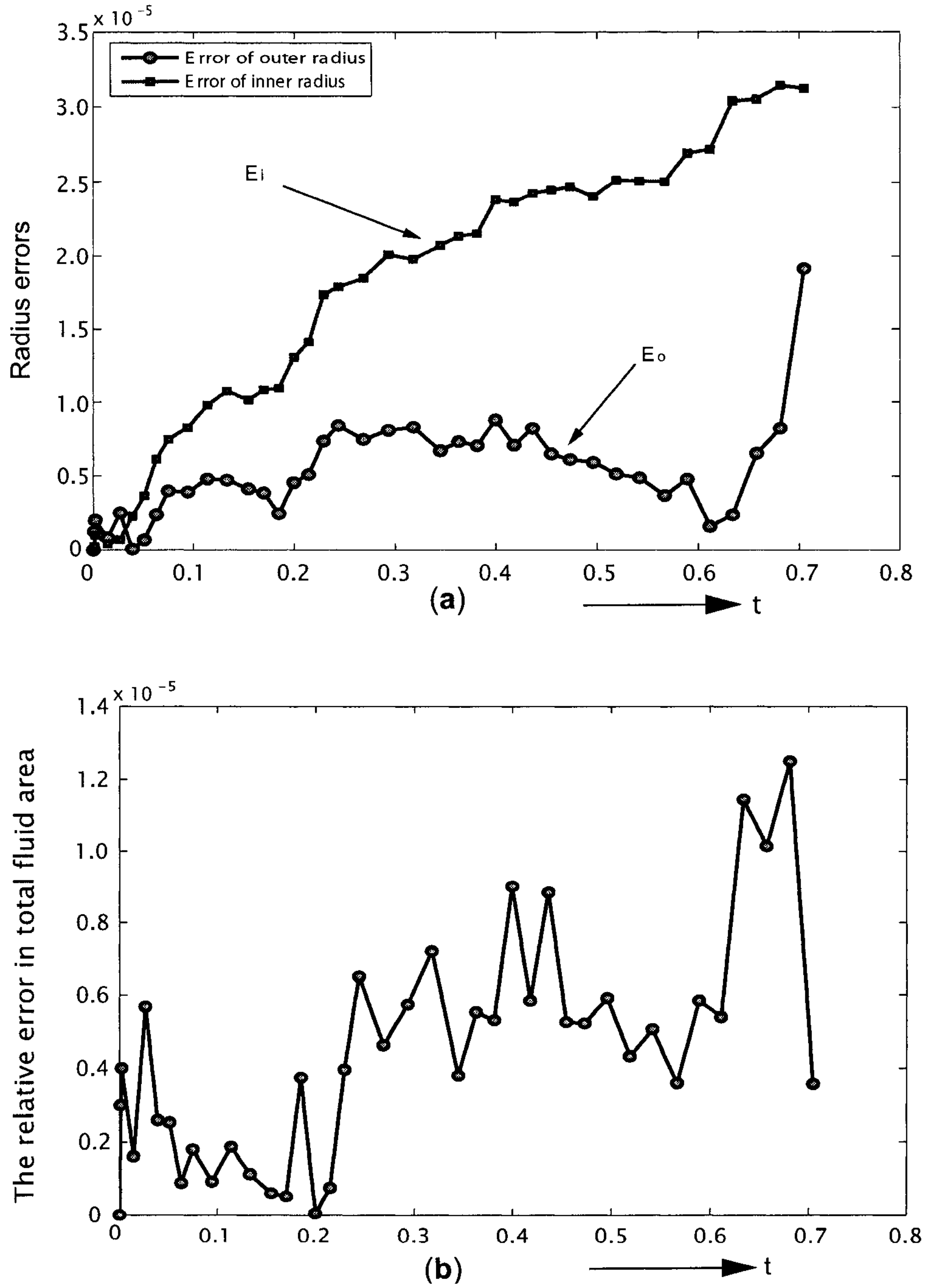


Figure 3.4 : The error between the exact solution and numerical solution by applying time integration method:
(a) the errors of the inner and outer radius, (b) the error of area change.

Table 3.1 : The Time steps for the sintering of circular disk with a circular hole.

i		t_i	h	p	*Area*	*Error*
1	rd	0	0.001000	1	2.3561945	0.0000
2		0.001000	0.001000	1	2.3562016	3.0e-06
3		0.002000	0.012375	2	2.3562039	4.0e-06
4		0.014375	0.012375	2	2.3561983	1.6e-06
5		0.026750	0.012375	2	2.3562079	5.7e-05
6		0.039126	0.011969	2	2.3562006	2.6e-06
7		0.051094	0.011969	2	2.3562004	2.5e-06
8		0.063063	0.011969	2	2.3561966	8.9e-07
9		0.075031	0.019478	3	2.3561902	1.8e-06
10		0.094509	0.019478	3	2.3561923	9.3e-06
11		0.113988	0.019478	3	2.3561901	1.9e-06
12		0.133466	0.019478	3	2.3561918	1.1e-06
13	rd	0.152944	0.001000	1	2.3561937	3.3e-07
14		0.153944	0.001000	1	2.3561963	7.7e-07
15		0.154944	0.014958	2	2.3561931	6.0e-07
⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	rd	0.541744	0.001000	1	2.35618094	5.8e-06
39		0.564559	0.001000	1	2.35618182	5.3e-06
40		0.565559	0.022466	2	2.35618602	3.6e-06
41		0.566559	0.022466	2	2.35618072	5.8e-06
42		0.589024	0.022466	2	2.35618177	5.4e-06
43		0.611490	0.023543	3	2.35616752	1.1e-05
44		0.633955	0.023543	3	2.35617057	1.0e-05
45		0.681042	0.023543	3	2.35616503	1.3e-05
46		0.704585	0.023543	3	2.35618609	3.6e-06

Note. After a node redistribution (rd), the time integration is restarted without further information.

Chapter 4

Simulation of 2-D Shapes

In this chapter, we present some results of numerical simulations for two-dimensional sintering geometries. Firstly, we simulate the sintering of a circular disk with a circular hole, and compare the results to the analytical solutions. Then, the simulation of a circular disk with an elliptic hole is given, and we show how the elliptic hole evolves. After that, we give several examples of cases when the inside hole(s) are very close to the outside boundary. Finally, we give the simulation for microstructured optical fibers.

4.1 Simulation of a Circular Disk with a Circular Hole

In this section we consider the evolution of a circular fluid disk with a circular hole centered at the origin. We denote by r_o the radius of outer circle and by r_i the inner radius. For this fluid domain, the solution of the Stokes equations can be found analytically, giving the radial velocity [van de Vorst 1993], as:

$$v_r(r) = \frac{r_o r_i}{2(r_i - r_o)r}, \qquad r_i \leqslant r \leqslant r_o. \tag{4.1}$$

From this solution and the condition $r_o > r_i$, it follows that the interior hole will finally vanish.

Also from this analytical solution of radial fluid velocity, we can deduce the exact solution for the inner and outer radius as the functions of time. By using both the conservation of the fluid surface and the Lagrangian representation in Equation (2.63) for the velocity, we can obtain the following ODEs for the inner and outer radius [van de Vorst 1993]:

$$\begin{aligned} \dot{r}_i &= \frac{(3+4r_i^2)^{\frac{1}{2}}}{2\left[2r_i - (3+4r_i^2)^{\frac{1}{2}}\right]} \qquad r_i(0) = R_i, \\ \dot{r}_o &= \frac{(4r_o^2-3)^{\frac{1}{2}}}{2\left[(4r_o^2-3)^{\frac{1}{2}} - 2r_o\right]} \qquad r_o(0) = R_o, \end{aligned} \tag{4.2}$$

and the time function with respect to r_i

$$t = 2(R_i - r_i) + (3+4r_i^2)^{\frac{1}{2}} - (3+4R_i^2)^{\frac{1}{2}}. \tag{4.3}$$

As mentioned in the Chapter 1, the values in this project are normalized by their characteristic values respectively. Now, we give out an general ideal about the realistic values of characteristic parameters corresponding to the typical sintering system of silica glass at $T \approx 1400$ °C: characteristic length $l \approx 100$ μm (the order of magnitude of the diameter of an optical fiber), the viscosity $\eta \approx 10^7$ $\mathrm{N\cdot s\cdot m^{-2}}$, and the surface tension $\gamma \approx 10^{-1}$ $\mathrm{N\cdot m^{-1}}$ [Guy 1976]. From these values, it follows that the characteristic velocity v_c is of the order of 10^{-8} $\mathrm{m\cdot s^{-1}}$, which gives the characteristic time t_c of the order of 10^4 s.

In this case, the initial outer radius (R_o) of the disk is taken equal to 1, and the inner radius (R_i) is set to 0.5. In Figure 3.2, we plotted the shrinkage of circular hole as time increasing. More specifically, both the initial geometry ($t = 0$) and the shape at time $t = 0.7$ are plotted in the figure. The evolutions of these two boundaries

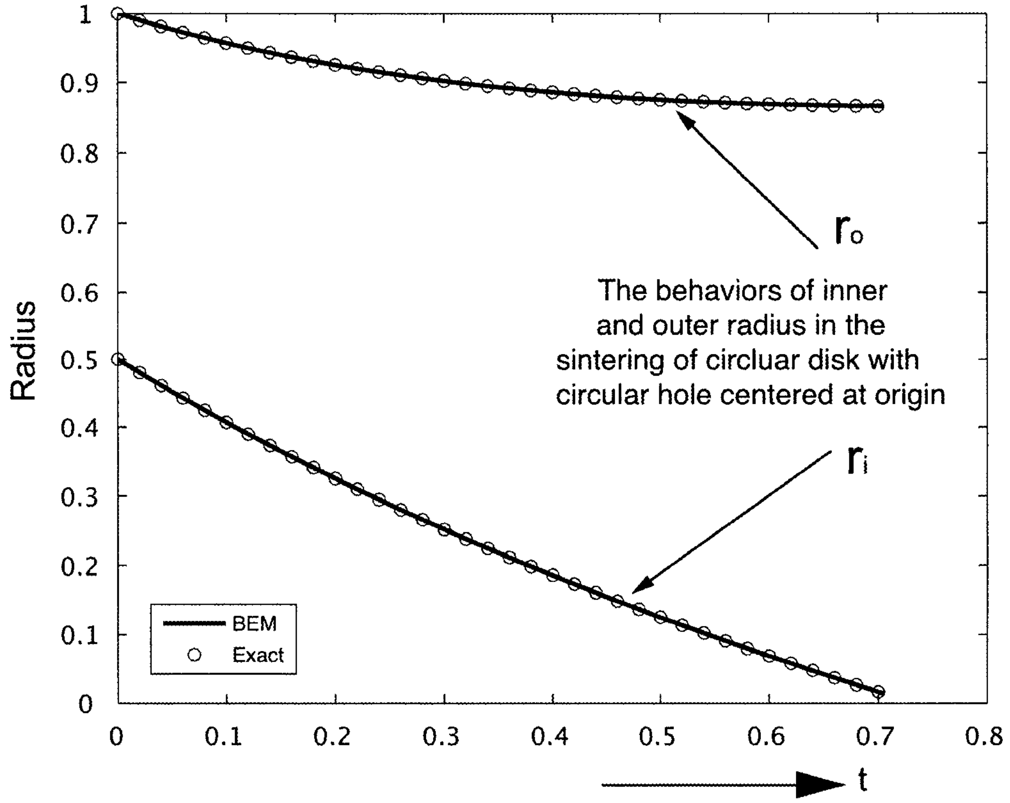


Figure 4.1 : The numerical evolution behaviors of inner and outer radius of the circular disk, which match very well with the exact analytical solutions.

are also plotted after successive time steps of 0.05. As can be observed, the inner hole shrinks and disappears after a certain period of time, then the domain becomes simply connected.

Figure 4.1 shows the exact solutions given by Equation (4.2) compared with the numerical ones. As can be observed, they match very well. The moment that the hole disappears is also very reasonably obtained from simulation: from Equation (4.3), by setting $r_i = 0$, we derive that the hole is filled up exactly at $t = \sqrt{3} - 1 \approx 0.73$; the latter value can be found by numerical code.

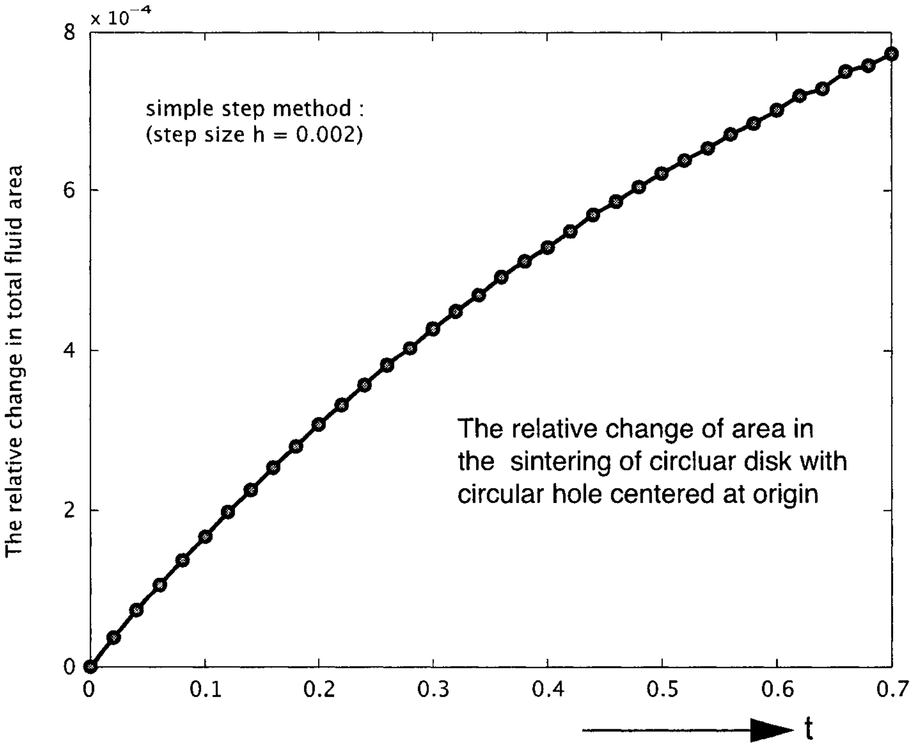


Figure 4.2 : The sintering error in total fluid area with simple step method.

In Chapter 3, we gave the evolution error for time integration method. Here, we give the sintering evolution error for simple step method (see Equation. (2.62)) with step size $h = 0.002$. In Figure 4.2, we can see that the relative change error in total area with simple step method is also very small (of $\mathcal{O}(h^2)$). So the simple step method, compared time integration method, has an advantage in the big system.

4.2 Simulation the Shrinkage for Elliptic Holes

This example is the deformation of a circular disk with an elliptic hole centered at the origin. This particular example is chosen since Hopper [Hopper 1991] analytically solved the shrinkage of an elliptic hole in an infinite region of fluid. From that solution, he found that an elliptic hole shrinks at a constant aspect ratio. Here, we use a circular disk to replace the infinite region. The radius of the disk is set equal to 1, and the ellipse is described by $x_1^2/a_1^2 + x_2^2/a_2^2 = 1$ with $a_1 = 0.3, a_2 = 0.2$ at $t = 0$.

In Figure 4.3 (a), we plotted this particular initial shape and the shape at the time at $t = 0.4$. Furthermore, the deformation is shown at times $t = 0$ (0.05) 0.4. In this example, we can see that the shrinkage of the inner elliptic hole proceeds as an elliptic hole, which is similar to what was obtained by Hopper [Hopper 1991] for the infinite fluid region case. In Figure 4.3 (b), we show the development of the axial ratio (a_2/a_1) of this inner ellipse as a function of time. As can be seen in the figure, this is almost linear decreasing function, even when the hole has reached the moment of vanishing. This result is different from what was obtained from Hopper's solution.

The linear decreasing of the axial ratio is valid only when the ends of the ellipse are not to close to the outer boundary. This is illustrated in Figure 4.4. Here we plotted the evolution of an enlarged elliptic hole in the same size circular disk at time $t = 0$ (0.05) 0.55. The ellipse is described by the equation $x_1^2/a_1^2 + x_2^2/a_2^2 = 1$ with $a_1 = 0.9, a_2 = 0.5$ at $t = 0$. The figure shows that the shrinkage of inner hole proceeds as ellipse-like shapes too.

In this case, the axial ratio of this elliptic hole does not decrease at a linear rate as can be observed from figure 4.4 (b). This differential behavior can be attributed to the influence of the outer boundary at the ends of the ellipse.

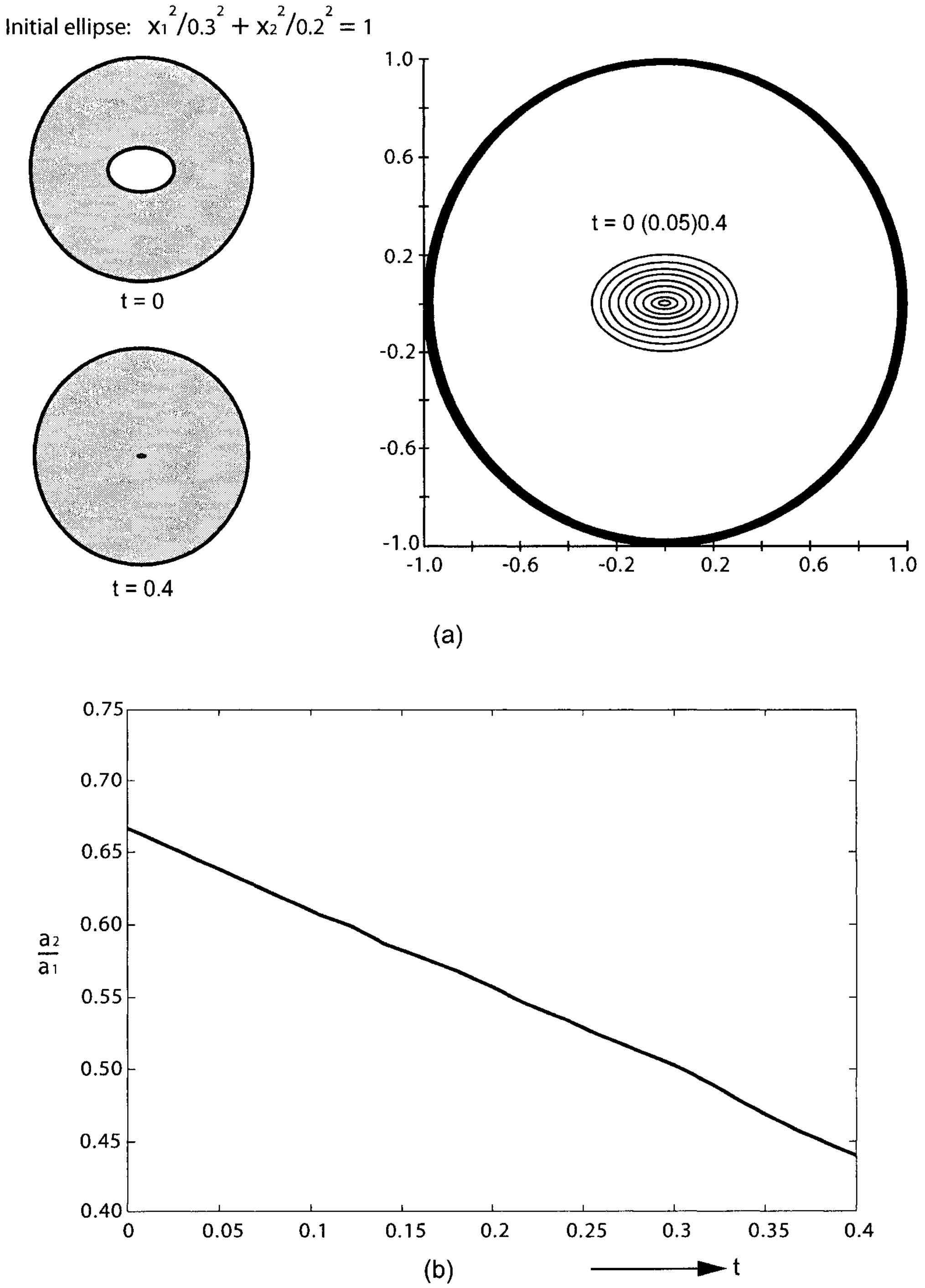


Figure 4.3 : The shrinkage of a circular disk with a small elliptic hole centered at the origin: (a) the shrinkage of boundaries, (b) the the axial ratio (a_2/a_1) of inner ellipse.

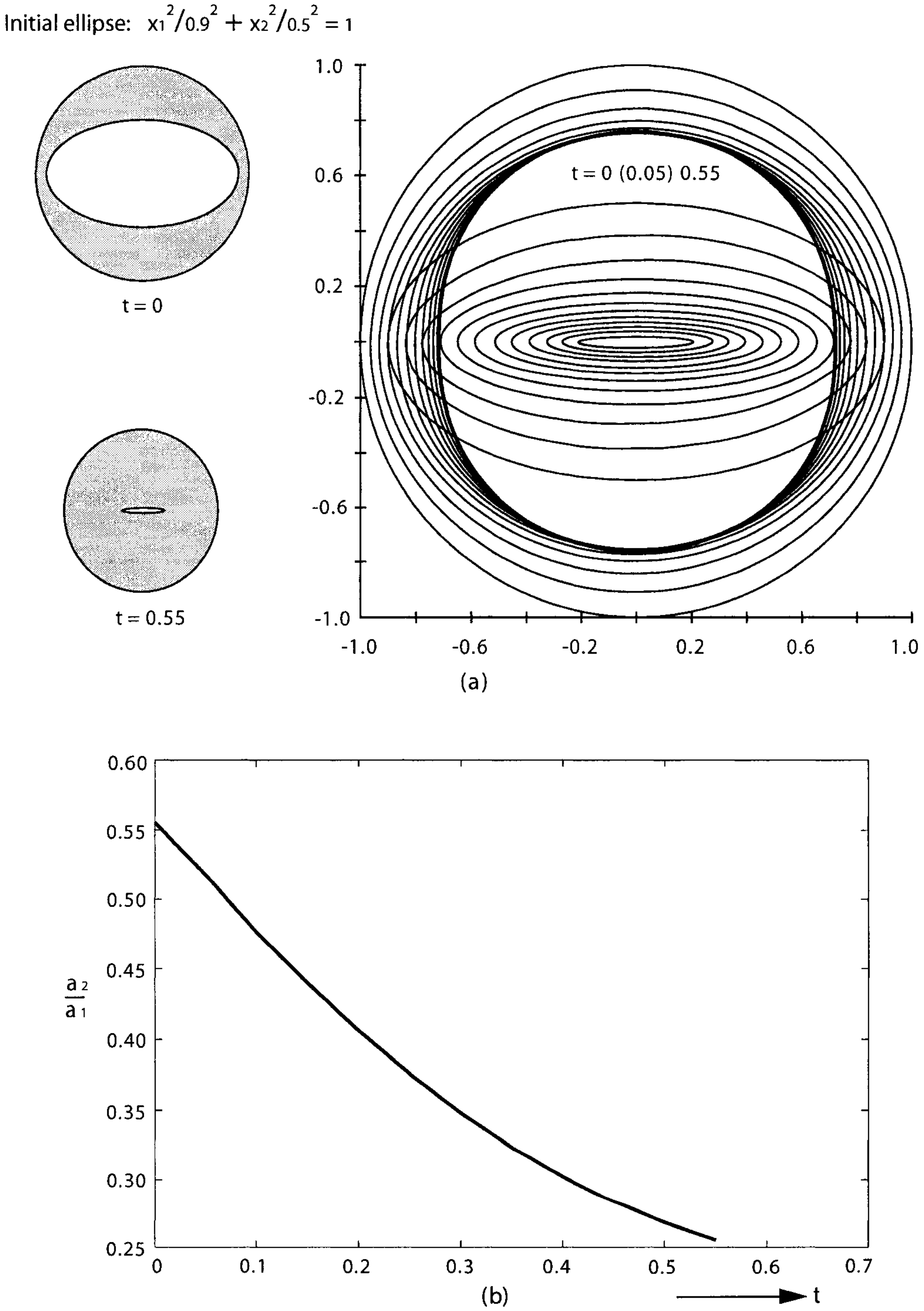


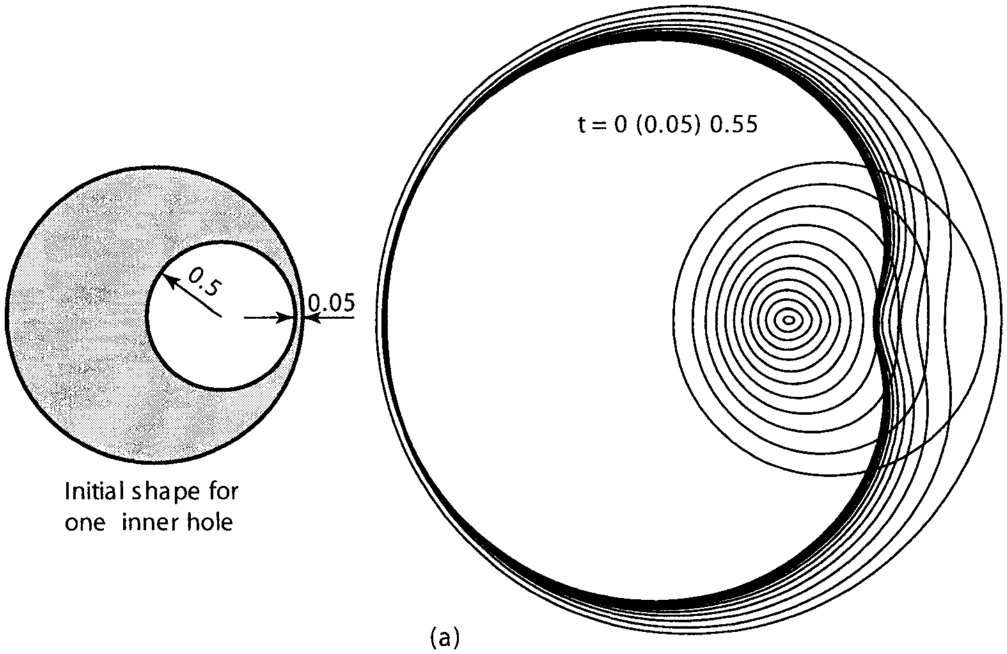
Figure 4.4 : The shrinkage of a circular disk with a big elliptic hole centered at the originy: (a) the shrinkage of boundaries, (b) the axial ratio (a_2/a_1) of inner ellipse.

4.3 Evolution of Holes Close to the Outer Boundary

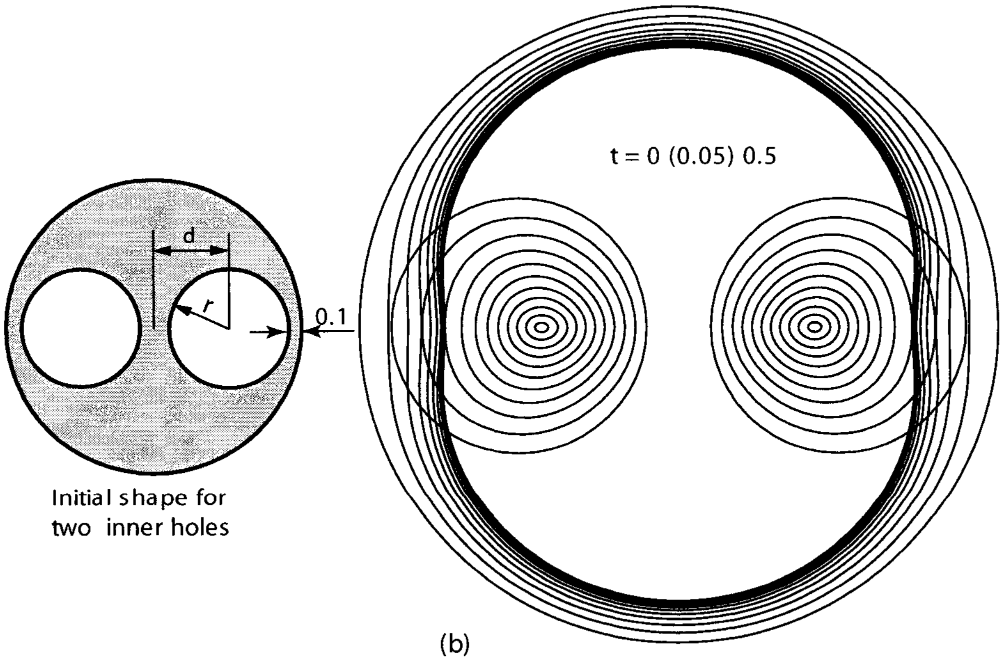
In the previous sections, we discussed the shrinkage of the inner holes centered the origin. In this section, we discuss the cases of inner holes close to the outer boundary. The interesting question in this particular geometry is whether the holes are going outside the fluid domain, i.e., whether the domain will become simply connected. In order to simulate this problem, Firstly, we take the example of one inner hole close to outer boundary, i.e., the circular hole has initial radius of 0.5 and is located at $x_1 = 0.45$ (the circular disk has radius 1 and is centered at the origin, and it is same for all following examples). Thus, the closet distance of the hole to the outer boundary is 0.05.

In Figure 4.5-(a), we plotted the deformations of boundaries from time $t = 0$ to 0.55. As can be seen, the hole remains in the fluid domain, and is completely filled up about $t = 0.6$. When this example is compared to that of Figure 3.2, we also can see that the shrinkage rate of this inner hole is faster. We note that these two geometries contain the same amount of the fluid and equally sized interior hole. Only the positions of the center of the inner hole differ. An explanation for the different shrinkage rate is that for a centered hole, all fluid has to be moved in the direction to the origin; however, in the case of a decentered hole, the hole is mainly filled up by the fluid flow that appears in the right part of the domain.

In Figure 4.5-(b), -(c) and -(d), we plotted the deformations for two holes, three holes and four holes respectively (holes are evenly distributed around circle, (b): two holes with $d = 0.5$ and $r = 0.4$; (c) three holes with $d = 0.5$ and $r = 0.4$; (d) four holes with $d = 0.55$ and $r = 0.35$). As can be observed, the boundary evolutions are similar to that for a single inner hole, i.e., (1) all the inner holes remain in the fluid domain and are filled up after a certain time; (2) the deformations of the closest parts to the inner holes are relatively larger than that in other parts.


0.5
0.05
Initial shape for one inner hole
t = 0 (0.05) 0.55

(a)


d
r
0.1
Initial shape for two inner holes
t = 0 (0.05) 0.5

(b)

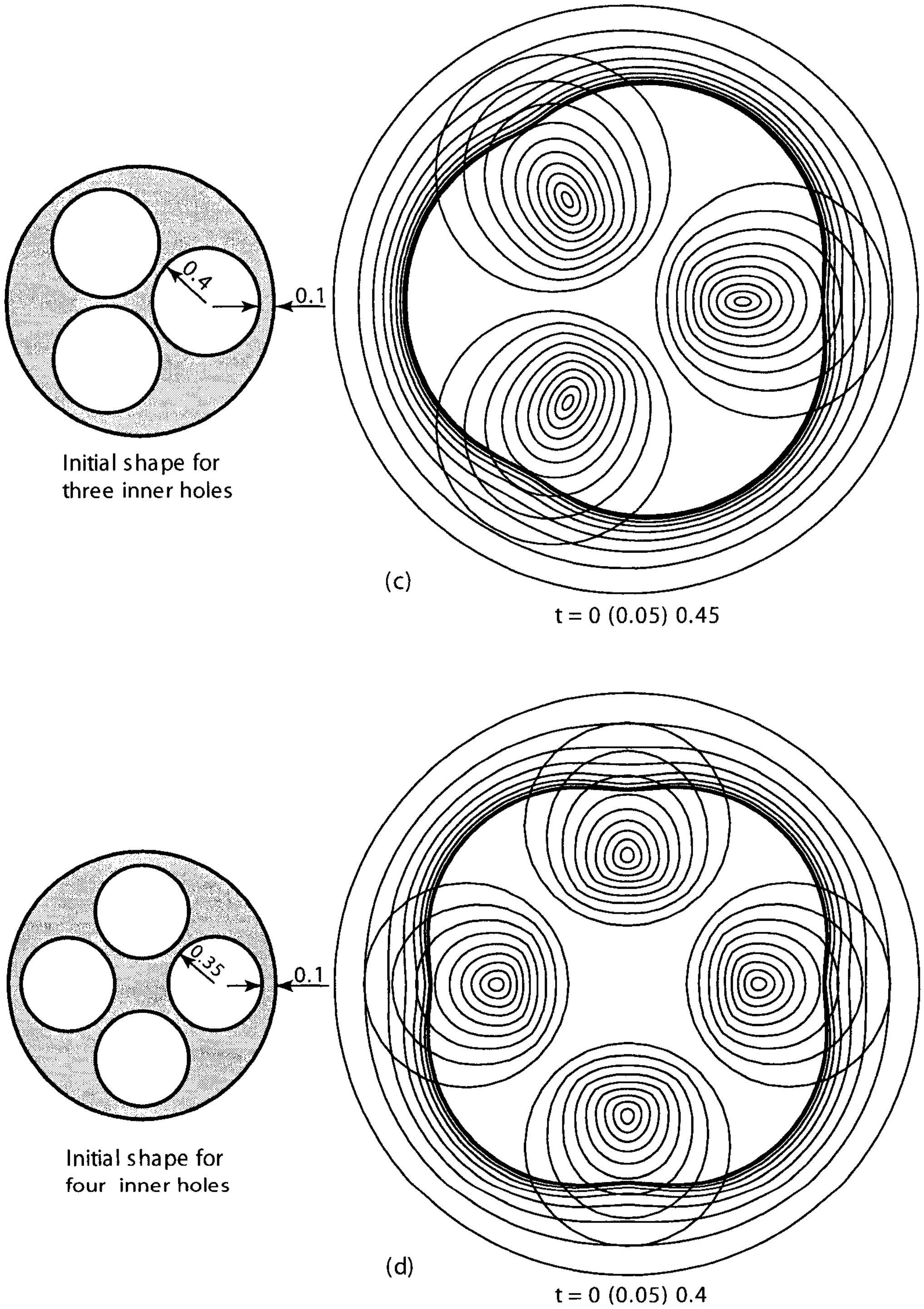


Figure 4.5 : The shrinkage of inner circular holes close to the outer boundary: (a) 1 hole ($r = 0.5$), (b) 2 holes ($r = 0.4$), (c) 3 holes ($r = 0.4$), (d) 4 holes ($r = 0.35$).

4.4 Simulation of Multi-Layer Inner Holes

As we mentioned before, microstructured optical fibers have been extensively studied because of their special properties. There is a need to know the fiber deformation when it is heated. In this section, we give some simulations for this kind of structure.

Most of the time, the inner holes of a microstructured optical fiber are set with multi-layer hexagons centered at origin, the number of holes for each layer being 6, 12, 18, $\cdots$, respectively (see Figure 4.6).

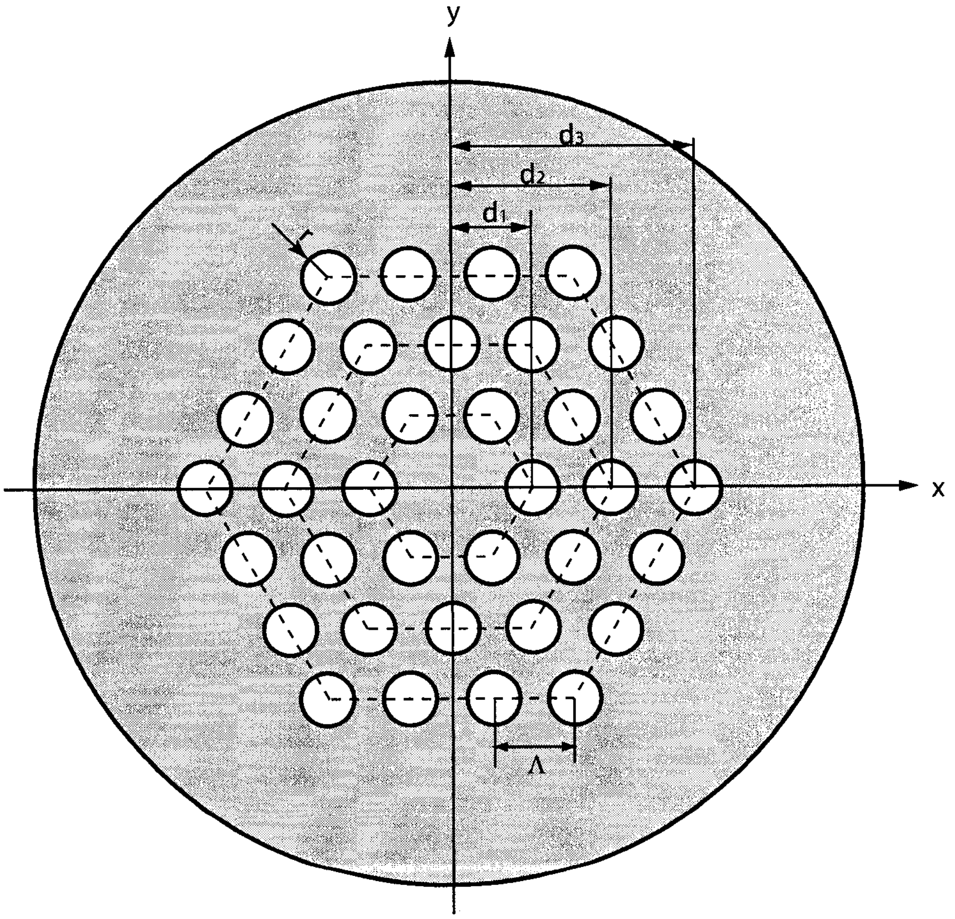


Figure 4.6 : The layout of microstructure optical fiber (3 layers for example). (Usually, $d_1 = \Lambda$, $d_2 = 2\Lambda$, $d_3 = 3\Lambda$)

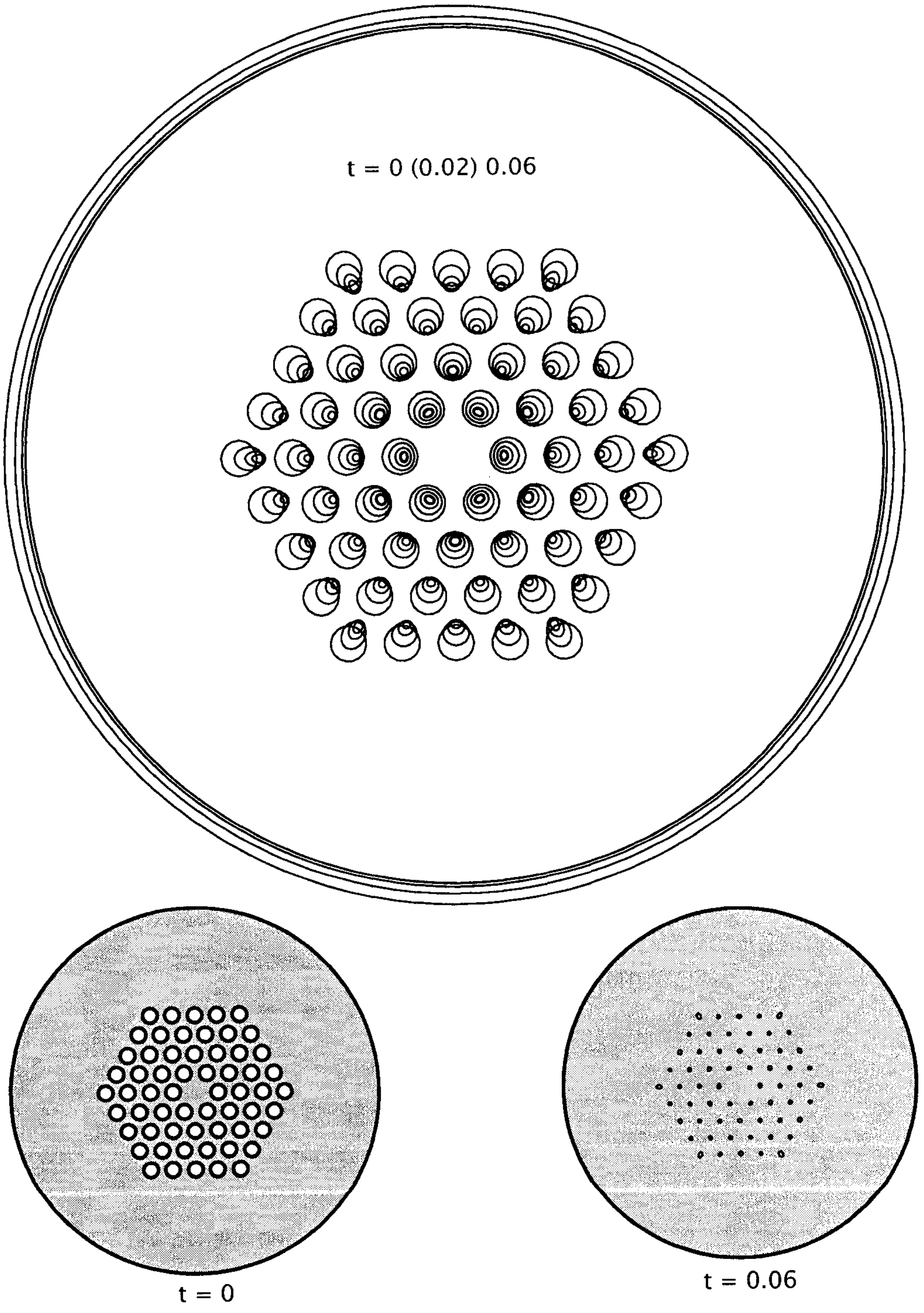


Figure 4.7 : The deformation of 4-layer inner holes.

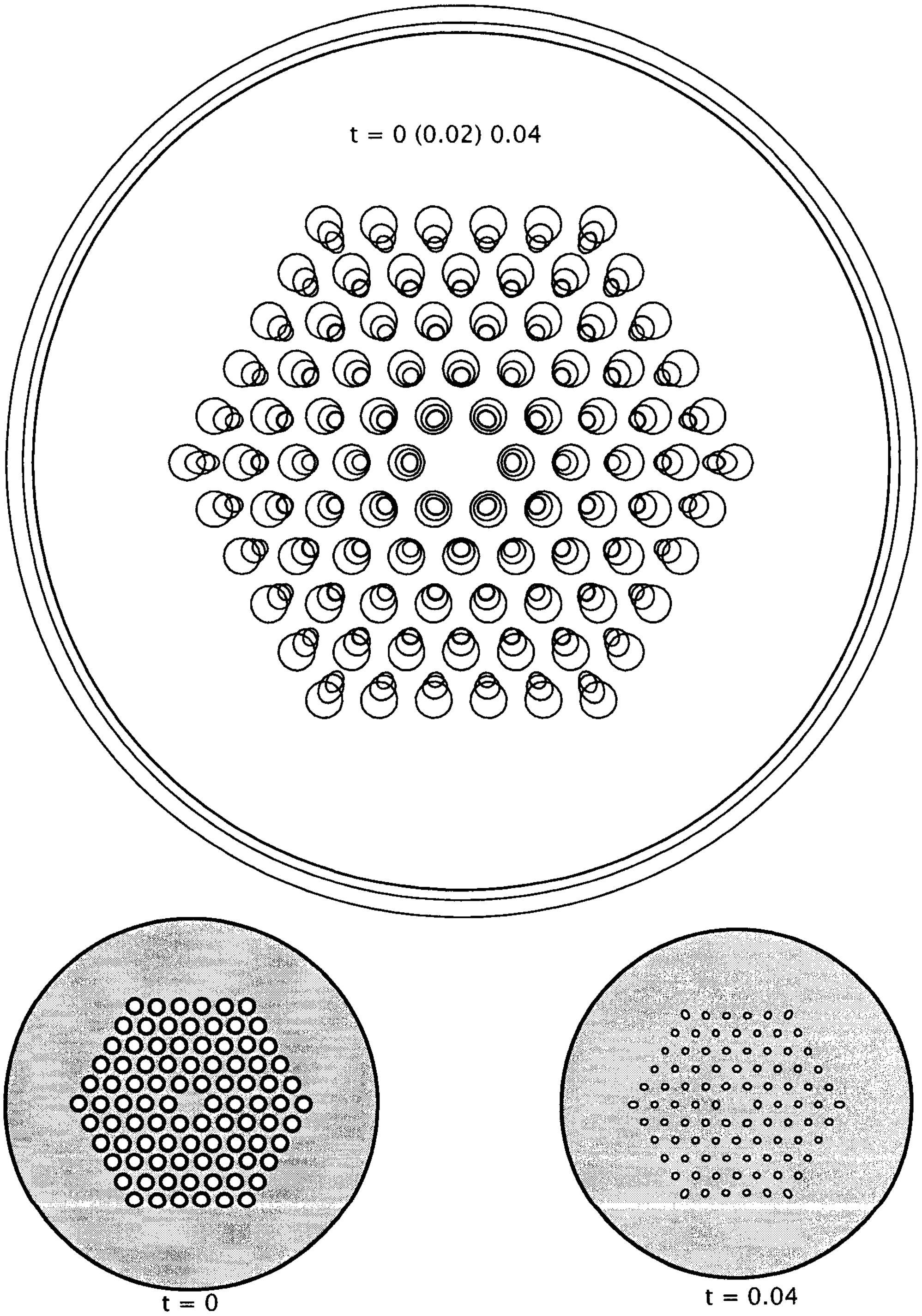


Figure 4.8 : The deformation of 5-layer inner holes.

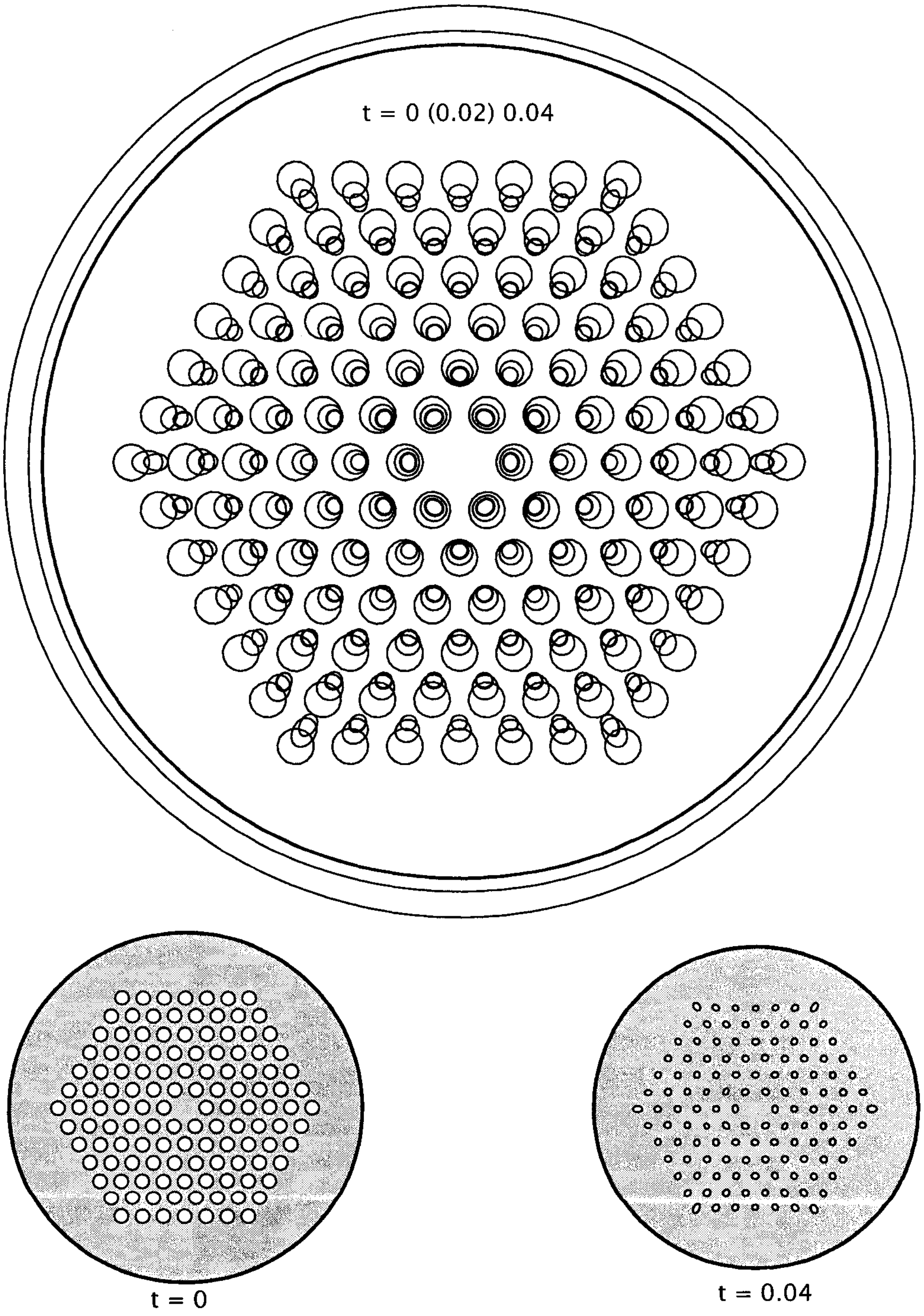


Figure 4.9 : The deformation of 6-layer inner holes.

In this section, we give three examples for four layers, five layers and six layers respectively. In Figure 4.7, we plotted the deformation for four layer inner holes, i.e., the radius of the hole is $r = 0.04$, and the distances of the layers are 0.12, 0.24, 0.36 and 0.48 respectively. In this Figure, we can see that the inner holes almost vanish at time $t = 0.06$. In Figures 4.8 and 4.9, we also plotted the deformation for five and six layers of hole respectively (with the same distance between layers 0.12 and radius of hole $r = 0.04$). As can be seen from these three simulations, the outer boundary almost always keeps its circular shape (just shrinking) during the sintering procedure, while the deformations of inner holes are different, depending on both the position of the hole and the number of layers. This simulation can be used to predict how the inner holes deform, and to estimate the time for certain deformation tolerance when we make the devices with this kind of fiber.

Conclusion

In this thesis, we have developed the mathematical model and numerical implementation for simulating 2-D viscous sintering problems. Since we are only interested in the boundary motion of the fluid domain, we chose the BEM to solve this sintering problem. During the sintering procedure, the nodal points on the boundaries become closer to each other. A large error in calculating the boundary curvature is expected for the cusp regions if the nodes are too close. We thus chose to redistribute these nodes based on the equal arc length after certain time of evolution. This method proved to be very effective.

There are two ways to describe the boundary motion, i.e., the Euler and the Lagrange representations. In this thesis, we introduced both of them to get the new boundary. Especially we introduced in great detail the time integration method (from the Lagrangian viewpoint) to obtain the new boundary geometry. Although it is more precise than a simple step method, it is quite time-consuming when the system is large.

Based on above ideas, we developed a software package with a user friendly interface for this simulation using C++ language. It is worth mentioning that we used BiConjugate Gradient Stabilized (Bi-CGStab) (see Appendix C) iteration method rather than the exact decomposition way to solve the linear algebraic equations. The Bi-CGStab method is almost 100 times faster than the lower and upper triangular matrix (LU) decomposition method or the orthogonal-triangular matrix (QR) decomposition method in solving 200 × 200 linear equations. On the other hand, the

Bi-CGStab method is more precise than the LU or QR decomposition methods when the system is ill-conditioned.

By using our program, we simulated the boundary evolutions of 2-D multiply connected domains in several cases. When the results may be compared to the corresponding analytical solutions, we found that they matched quite well, and that the relative change in total fluid area is less than 8×10^{-4} (for a simple step method with $h = 0.002$) and 1.5×10^{-5} (for a time integration method).

The package that we have developed, is able to simulate all kinds of two-dimensional multiply connected domains once the initial boundaries are given. The main limitation comes from the computer resources. Note, however, that, in this simulation, we assume that all the boundaries are always separated during the sintering process; we do not give the treatment for touching boundaries during evolution. Hence, a further improvement of this simulation capabilities may be achieved by designing the algorithms for detecting and handling the case of such touching boundaries and for the effect of positive/negative pressure in the holes.

Bibliography

ARFKEN, G., *Mathematical Methods for Physicists, 3rd ed*, Academic Press, Orlando (1985).

ARIS, R., *Vectors, Tensors, and the Basica Equations of Fluid Mechanics*, Prentice-Hall, Englewood Cliffs NY. (1962).

ARNOLD, D. N. & WENDLAND, W. L., On the Asymptotic Convergence of Collocation Methods. *Math. Comp.*, **41** (1983).

BAKER, C. T. H., *The Numerical Treatment of Integral Equations*, Clarendon Press, Oxford (1977).

BATCHELOR, G. K., *An Introduction to Fluid Dynamics*, Cambridge University Press, Cambridge (1967).

BECKER, A. A., *The Boundary Element Method in Engineering*, McGraw-Hill Book Company, London (1992).

BREBBIA, C. A. & DOMINGUEZ, J., *Boundary Elements: An Introductory Course*, Computational Mechanics Publications, Southampton Boston (1989).

BREBBIA, C. A. & WALKER, S., *The Boundary Element Techniques in Engineering*, Butterworth, London (1980).

CABRAL, J. J. S. P., WROBEL, L. C. & BREBBIA, C. A., A BEM Formulation Using B-Splines: I- Uniform Blending Functions. *Engng. Analysis*, **7** (1990).

CABRAL, J. J. S. P., WROBEL, L. C. & BREBBIA, C. A., A BEM Formulation Using B-Splines: II- Multiple Knots and Non-Uniform Blending Functions. *Engng. Analysis*, **8** (1991).

COOKER, M. J., PEREGRINE, D. H., VIDAL, C. & DOLD, J. W., The Interaction Between a Solitary wave and a Submerged Semicircular Cylinder. *J. Fluid Mech.*, **215** (1990).

COSGROVE, G. J., STROZIER, J. A. & SEIGLE, L. L., An Approximate Analytical Model for the Late-Stage Sintering of an Array of Rods by Viscous Flow. *J. Appl. Phys*, **47** (1976).

FRENKEL, J., Viscous Flow of Crystalline Bodies Under the Action of Surface Tension. *J. Phys*, **9** (1945).

GEAR, C. WILLIAM, *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice-Hall Inc. (1971).

GRAAF, J. DE, Mathematical Addenda to Hopper's Model of Plane Stokes Flow Driven by Capillarity on a Free Surface. In HELMINCK, G. F., ed., *Proc. Conf. on Mathematical Physics* (1992).

GRILLI, S. T., SKOURUP, J. & SVENDSEN, I. A., An efficient boundary element method for nonlinear water wave. *Engng. Analysis*, **6** (1989).

GUY, A. J., *Essentials of Materials Science*, McGraw-Hill (1976).

HAIRER, E. & G.WANNER, *Solving Ordinary Differential Equation II, Stiff and Differential-Algebraic Problems, Second Revised Edition*, Springer-Verlag Berlin Heidelberg (1996).

HARTMANN, F., Computing the C-matrix in Non-smooth Boundary Points. In BREBBIA, C. A., ed., *New Developments in Boundary Element Methods*, 367–379 (1980).

HOPPER, R. W., Coalescence of Two Equal Cylinders - Exact Results for Creeping Viscous Plane Flow Driven by Capillarity. *J. Am. Ceram. Soc*, **67** (1984).

HOPPER, R. W., Plane Stokes Flow Driven by Capillarity on a Free Surface, 2: Futher Developments. *J. Fluid Mech.*, **230** (1991).

HOPPER, R. W., Stokes Flow of a Cylinder and Half-Space Driven by Capillary. *J. Fluid Mech.*, **243** (1992).

HSIAO, G. C., KOPP, P. & WENDLAND, W. L., Some Applications of a Galerkin-collocation Method for Boundary Integral Equations of the First Kind. *Math. Meth. in the Appl. Sci.*, **6** (1984).

INGHAM, D. B. & KELMANSON, M. A., *Boundary Integral Equation Analyses of Singular, Potential and Biharmonic Problems*, Springer-Verlag Berlin (1984).

JAGOTA, A. & DAWSON, P. R., Micromechanical Modeling of Powder Compacts II. Truss Formulation of Discrete Packings. *Acta Metall*, **36** (1988).

JAGOTA, A. & DAWSON, P.R., Simulation of the Viscous Sintering of Two Particles. *J. Am. Ceram. Soc*, **73** (1990).

JAWAON, M. A. & SYMM, G. T., *Integral Equation Methods in Potential Theory and Elastostatics*, Butterworth-Heinemann, Boston (1991).

KIM, S. & KARRILA, S. J., *Microhydrodynamiccs: Principles and Selected Applications*, Butterworth-Heinemann (1991).

KORWIN, D. M., LANGE, S. R., EATON, W. C., JOSEPH, I. *et al.*, A Study of Sintering Behavior of Glass in Two Geometric Configurations. In *Proc. Conf. on 16th Intnl. Congress on Glass* (1992).

KRESS, R., *Linear Integral Equations*, Springer-Verlag, New York (1989).

KUIKEN, H. K., Viscous Sintering: the Surface Tension-Driven Flow of a Liquid from under the Influence of Curvature Gradients at Its Surface. *J. Fluid Mech.*, **214** (1990).

KUIKEN, H. K., Deformation Surface and Viscous Sintering. In HUNT, J. C. R., ed., *The Mathematics and the Computation of Deforming Surface*, Oxford University Press, Oxford (1992).

LADYZHENSKAYA, O. A., *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach, New York (1963).

LAMBERT, J. D., *Computational Methods in Ordinary Differential Equations*, Wiley & Sons New York (1973).

LEE, S. H. & LEAL, L. G., The Motion of a Sphere in the Presence of a Deformable Interface. Part 2: Numerical Study of the Translation of a Sphere Normal to an Interface. *J. Colloid Interface Sci.*, **87** (1982).

MACKENZIE, J. K. & SHUTTLEWORTH, R., A Phenomenological Theory of Sintering. *Proc. Phys. Soc. Lond.*, **62** (1949).

POWER, H. & MIRANDA, G., Second Kind Integral Equation Formulation of Stokes' Flow Past a Particle of Arbitrary Shape. *SIMA J. Appl. Math.*, **47** (1987).

POZRIKIDIS, C., *Boundary Integral and Singularity Methods for Linearized Viscous Flow*, Cambridge University Press (1992).

PROKERT, G., Quasistatic Model for the Motion of a Viscous Capillary Liquid Drop. *NASA STI/Recon Technical Report N*, **95** (1993).

RADHAKRISHNAN, K. & HINDMARSH, A. C. (1993). *Description and Use of Lsode, the Livermore Solver for Ordinary Differential Equations*. Technical Report, Lawrence Livermore National Laboratory.

RALLISON, J. M. & ACRIVOS, A., A Numerical Study of the Deformation and Burst of a Viscous Drop in an Extensional Flow. *J. Fluid Mech.*, **89** (1978).

REED, J. S., *Introduction to the Principle of Ceramic Processing*, Wiley Interscience, Chichester (1988).

RICHARDSON, S., Two-Dimensional Slow Viscous Flows with Time-Dependent Free Boundary Driven by Surface Tension. *Euro. J. App. Math.*, **3** (1992).

ROSS, J. W., MILLER, W. A. & WEATHERLY, G. C., Dynamic Computer Simulation of Viscous Flow Sintering Kinetics. *J. Appl. Phys*, **52** (1981).

SCHERER, G. W., Sintering of Low-Density Glasses: I. Theory. *J. Am. Ceram. Soc*, **60** (1977).

SCHERER, G. W., Viscous Sintering of a Bimodeal Pore-Size Distribution. *J. Am. Ceram. Soc*, **67** (1984).

SONNEVELD, P., CGS: A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems. *SIAM J. Sci. Statist. Comput*, **10** (1989).

VAN DE VORST, G. A. L., Integral Method for a Two-Dimensional Stokes Flow with Shrinking Holes Applied to Viscous Sintering. *J. Fluid Mech.*, **257** (1993).

VAN DE VORST, G. A. L. (1994). *Modelling and Numerical Simulation of Viscous Sintering*. Doctor Thesis, Eindhoven University of Technology.

VAN DE VORST, G. A. L. & MATTHEIJ, R. M. M., Numerical Analysis of a 2-D Viscous Sintering Problem with Non-Smooth Boundaries. *Computing*, **49** (1992).

VAN DER VORST, H. A., Bi-CGSTAB : A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems. *SIAM j. Sci. Statist. Comput.*, **13** (1992).

WALDRAM, J. R., *The Theory of Thermodynamics*, Cambridge University Press, Cambridge (1985).

WEIDMANN, J., *Linear Operators in Hilbert Spaces*, Springer-Verlag, New York (1980).

WILKINSON, J. H., *The Algebraic Eigenvalue Problem*, Clarendon Press, Oxford (1965).

WU, WANG-YI, *Fluid Mechanics*, Beijing University Press (1982).

YANG, DE-QUAN & ZHAO, ZHONG-SHENG, *Theory of Boundary Element and Its Applications*, Beijing Institute of Technology Press (2002).

YOUNGERN, G. K. & ACRIVOS, A., Stokes Flow Past a Particle of Arbitrary Shape: A Numerical Method of Solution. *J. Fluid Mech.*, **69** (1975).

Appendix A

Numerical Integration Schemes

A large number of numerical integration schemes may be found in the literature, many of them designed for special problems. Gaussian integration formulae being general, simple, and very accurate, they have been used in Chapter 2.

In the following, one dimensional Gaussian integration formulae for non-singular functions are presented first. They can be used for integration over elements. Then, we give the one dimensional integration in which a logarithmic singularity is located at one end of the integration domain. These formulae have been used for the elements that include the source point. In these cases the fundamental solutions present a logarithmic singularity.

A.1 One Dimensional Gaussian Quadrature

The integral in this case can be written as [Brebbia & Dominguez 1989],

$$I = \int_{-1}^{1} f(\xi) d\xi = \sum_{i=1}^{n} w_i f(\xi_i) + E_n, \tag{A.1}$$

where n in the number of the integration points, ξ_i is the coordinate of the i^{th} integration point, w_i is the associated weighting factor and E_n is the residual, i.e.,

$$E_n = \frac{2^{2n+1}(n!)^4}{(2n+1)[(2n)!]^3}\frac{d^{2n}f(\xi)}{d\xi^{2n}}, \qquad (-1 < \xi < 1). \tag{A.2}$$

Formula (A.1) is based on the representation of $f(\xi)$ by means of Legendre polynomials $P_n(\xi)$. The ξ_i value is the coordinate at point i where P_n is zero and for which the weights are given by

$$w_i = \frac{2}{(1-\xi_i^2)\left[\frac{dP_n(\xi)}{d\xi}\right]^2_{\xi=\xi_i}}. \tag{A.3}$$

Values of ξ_i and w_i are listed in Table A.1. Notice that for ξ_i values symmetric with respect to $\xi = 0$, w_i is the same.

A.2 One Dimensional Logarithmic Gaussian Quadrature

Kernels including a logarithmic singularity at one end of the integration domain can be integrated using following formula [Brebbia & Dominguez 1989]

$$I = \int_0^1 \ln(\frac{1}{\xi})f(\xi)d\xi \cong \sum_{i=1}^{n} w_i f(\xi_i), \tag{A.4}$$

where the integration point coordinate ξ_i and weighting factor w_i are given in Table A.2

Table A.1 : The values of ξ_i and w_i for one dimensional Gaussian quadrature

$\pm\xi_i$	w_i	$\pm\xi_i$	w_i
		$n = 8$	
$n = 2$		0.1834 3464 2496	0.3626 8378 3378
0.5773 5026 9190	1.0000 0000 0000	0.5255 3240 9916	0.3137 0664 5878
		0.7966 6647 7414	0.2223 8103 4453
$n = 3$		0.9602 8985 6498	0.1012 2853 6290
0.0000 0000 0000	0.8888 8888 8888		
0.7745 9666 9241	0.5555 5555 5555	$n = 9$	
		0.0000 0000 0000	0.3302 3935 5001
$n = 4$		0.3242 5342 3404	0.3123 4707 7040
0.3399 8104 3585	0.6521 4515 4863	0.6133 7143 2701	0.2606 1069 6403
0.8611 3631 1594	0.3478 5484 5137	0.8360 3110 7327	0.1806 4816 0695
		0.9681 6023 9508	0.0812 7438 8362
$n = 5$			
0.0000 0000 0000	0.5688 8888 8889	$n = 10$	
0.5384 6931 0106	0.4786 2867 0499	0.1488 7433 8982	0.2955 2422 4715
0.9061 7984 5939	0.2369 2688 5056	0.4333 9539 4129	0.2692 6671 9310
		0.6794 0956 8299	0.2190 8636 2516
$n = 6$		0.8650 6336 6689	0.1494 5134 9151
0.2386 1918 6083	0.4679 1393 4573	0.9739 0652 8517	0.0666 7134 4309
0.6612 0938 6466	0.3607 6157 3048		
0.9324 6951 4203	0.1713 2449 2379	$n = 12$	
		0.1252 3340 8511	0.2491 4704 5813
$n = 7$		0.3678 3149 8998	0.2334 9253 6538
0.0000 0000 0000	0.4179 5918 3673	0.5873 1795 4287	0.2031 6742 6723
0.4058 4515 1377	0.3818 3005 0505	0.7699 0267 4194	0.1600 7832 8543
0.7415 3118 5599	0.2797 0539 1489	0.9041 1725 6370	0.1069 3932 5995
0.9491 0791 2343	0.1294 8496 6169	0.9815 6063 4247	0.0471 7533 6387

Table A.2 : The values of ξ_i and w_i for one dimensional logarithmic Gaussian quadrature

n	ξ_i	w_i	n	ξ_i	w_i
2	0.11200 88062	0.71853 93190	8	0.13320 24416(-1)	0.16441 66047
	0.60227 69081	0.28146 06810		0.79750 42901(-1)	0.23752 56100
3	0.63890 79309(-1)	0.51340 45522		0.19787 10293	0.22684 19844
	0.36899 70637	0.39198 00412		0.35415 39943	0.17575 40790
	0.76688 03039	0.94615 40657(-1)		0.52945 85752	0.11292 40302
4	0.41448 48020(-1)	0.38346 40681		0.70181 45299	0.57872 21072(-1)
	0.24527 49143	0.38687 53178		0.84937 93204	0.20979 07374(-1)
	0.55616 54536	0.19043 51270		0.95332 64501	0.36864 07104(-2)
	0.84898 23945	0.39225 48713(-1)	9	0.10869 33608(-1)	0.14006 84387
5	0.29134 47215(-1)	0.29789 34718		0.64983 66634(-1)	0.20977 22052
	0.17397 72133	0.34977 62265		0.16222 93980	0.21142 71499
	0.41170 25203	0.23448 82900		0.29374 99040	0.17715 62339
	0.67731 41746	0.98930 45952(-1)		0.44663 18819	0.12779 92280
	0.89477 13610	0.18911 55214(-1)		0.60548 16628	0.78478 90261(-1)
6	0.21634 00584(-1)	0.23876 36626		0.75411 013716	0.39022 50498(-1)
	0.12958 33912	0.30828 65733		0.87726 58288	0.13867 29555(-1)
	0.31402 04499	0.24531 74266		0.96225 05594	0.24080 41036(-2)
	0.53865 72174	0.14200 87566	10	0.90426 30962(-2)	0.12095 51320
	0.75691 53374	0.55454 62232(-1)		0.53971 26622(-1)	0.18636 35426
	0.92266 88514	0.10168 95869(-1)		0.13531 18246	0.19566 08733
7	0.16719 35541(-1)	0.19616 93894		0.24705 24163	0.17357 71422
	0.10018 56779	0.27030 26442		0.38021 25396	0.13569 56730
	0.24629 42462	0.23968 18730		0.52379 23180	0.93646 75854(-1)
	0.43346 34933	0.16577 57748		0.66577 52055	0.55787 72735(-1)
	0.63235 09880	0.88943 22714(-1)		0.79419 04160	0.27159 81090(-1)
	0.81111 86267	0.33194 30436(-1)		0.89816 10912	0.95151 82605(-2)
	0.94084 8167	0.59327 87015(-2)		0.96884 79887	0.16381 57633(-2)

Note. Numbers are to be multiplied by the power of 10 *in parentheses.*

Appendix B

Conjugate Gradient Squared

The Conjugate Gradient Squared (CGS) method is a well-known and widely used iterative method for solving non-symmetric linear systems of equations $Ax = b$. The algorithm [Sonneveld 1989] is given in Table B.1.

Table B.1 : The algorithm of Conjugate Gradient Squared method

```
$r_0 = b - Ax_0;\quad q_0 = p_{-1} = 0;$
$\rho_{-1} = 1;\quad \rho_0 = \tilde{r}_0^T r_0;$ (An arbitrarily chosen vectors $\tilde{r}_0$ such that $\tilde{r}_0^T r_0 \neq 0$ )
$for\ n = 1, 2, 3, \cdots do$
    $\beta = \rho_n / \rho_{n-1};$
    $u_n = r_n + \beta q_n;$
    $p_n = u_n + \beta(q_n + \beta p_{n-1});$
    $v_n = Ap_n;$
    $\sigma = \tilde{r}_0^T v_n;\quad \alpha = \rho_n / \sigma$
    $q_{n+1} = u_n - \alpha v_n;$
    $f_{n+1} = u_n + q_{n+1};$
    $r_{n+1} = r_n - \alpha A f_{n+1};$
    $x_{n+1} = x_n + \alpha f_{n+1};$
    $\rho_{n+1} = \tilde{r}_0^T r_{n+1};$
    $c_{n+1} = r_{n+1}^T r_{n+1};$
    if $(|e_{n+1}| < tol)$ then
        Stop
    end if
end for
```

Appendix C

Bi-Conjugate Gradient Stabilized

The BiConjugate Gradient Stabilized method (Bi-CGStab) was developed to avoid the often irregular convergence patterns of the CGS method (Appendix B). The algorithm [van der Vorst 1992] is given in Table C.1.

Table C.1 : The algorithm of Bi-Conjugate Gradient Stabilized method

```
$r_0 = b - Ax_0; \quad \tilde{r}_0 = r_0;$
$\rho_0 = \alpha = \omega_0 = 1;$
$v_0 = p_0 = 0;$
$for\ n = 1, 2, 3, \cdots do$
    $\rho_n = \tilde{r}_0^T r_{n-1};$
    $\beta = (\rho_n/\rho_{n-1})(\alpha/\omega_{n-1});$
    $p_n = r_{n-1} + \beta(p_{n-1} - \omega_{n-1} v_{n-1});$
    $v_n = Ap_n;$
    $\alpha = \rho_n/(\tilde{r}_0^T v_n);$
    $s = r_{n-1} - \alpha v_n;$
    $t = As;$
    $\omega_n = t^T s/(t^T t);$
    $x_n = x_{n-1} + \omega_n s + \alpha p_n;$
    $r_n = s - \omega_n t;$
    If $(r_n^T r_n < tol)$ then
        Stop
    end if
end for
```