

# **The Dual Reciprocity Boundary Element Method**

To Sandra, Carolyn and Ruth

## The Ballad of the Boundary

### PART I<sup>1</sup>

There was a young man from Spain  
Who couldn't analyse his frame  
Then suddenly, "Discretize,  
Assemble and Triangularize  
And Solve, on an Electronic Brain!"

It's incredible, I'll hasten  
To the extension of it's application-  
Shells, viscoelasticity,  
Circulation of the North Sea  
And Beams on an Elastic Foundation!

So Paul, Bob, Stu, Dave, all present  
At 13, University Crescent  
Called GEORGE<sup>2</sup> on the telephone  
And wore their fingers to the bone  
To serve the Great Finite Element!

### PART II

(Some time and many megabytes later)

But there are problems with many an application-  
For example, Moving Meshes and Convection  
And it uses too much data  
We must find something better  
That permits a reduction of the Dimension!

In the simple case this is all very well  
But how to get rid of the infernal cell?  
Just take them to the Boundary  
Using Dual Reciprocity  
For any equation, it's as clear as a bell!

So here in Ashurst Tropical are we  
Students and staff work as hard as can be  
At VAX and microcomputer  
Meanwhile, I am the Director  
The Great Boundary Element is me!

---

<sup>1</sup>This poem is wholly my responsibility, Carlos and Luiz have no blame whatsoever. Part I was originally published with the title "The Ballad of the Great Finite Element" as a frontispiece to my PhD thesis, (Univ. Southampton, 1976), Paul.

<sup>2</sup>The operating system of the SRC ICL 1906A computer then installed at the Atlas Laboratory, Harwell, Oxon.

# **International Series on Computational Engineering**

## **Aims:**

Computational Engineering has grown in power and diversity in recent years, and for the engineering community the advances are matched by their wider accessibility through modern workstations.

The aim of this series is to provide a clear account of computational methods in engineering analysis and design, dealing with both established methods as well as those currently in a state of rapid development.

The series will cover books on the state-of-the-art development in computational engineering and as such will comprise several volumes every year covering the latest developments in the application of the methods to different engineering topics. Each volume will consist of authored work or edited volumes of several chapters written by the leading researchers in the field. The aim will be to provide the fundamental concepts of advances in computational methods as well as outlining the algorithms required to implement the techniques in practical engineering analysis.

The scope of the series covers almost the entire spectrum of solid mechanics, fluid mechanics and electromagnetics. As such, it will cover Stress Analysis, Inelastic Problems, Contact Problems, Fracture Mechanics, Optimization and Design Sensitivity Analysis, Plate and Shell Analysis, Composite Materials, Probabilistic Mechanics, Fluid Mechanics, Groundwater Flow, Hydraulics, Heat Transfer, Geomechanics, Soil Mechanics, Wave Propagation, Acoustics, Electromagnetics, Electrical Problems, Bioengineering, Knowledge Based Systems and Environmental Modelling.

## **Series Editor:**

Dr C.A. Brebbia  
Wessex Institute of Technology  
Computational Mechanics Institute  
Ashurst Lodge  
Ashurst  
Southampton SO4 2AA  
UK

## **Associate Editor:**

Dr M.H. Aliabadi  
Wessex Institute of Technology  
Computational Mechanics Institute  
Ashurst Lodge  
Ashurst  
Southampton SO4 2AA  
UK

---

## **Editorial Board:**

Professor H. Antes  
Institut für Angewandte Mechanik  
Technische Universität Braunschweig  
Postfach 3329  
D-3300 Braunschweig  
Germany

Professor D. Beskos  
Civil Engineering Department  
School of Engineering  
University of Patras  
GR-26110 Patras  
Greece

Professor H.D. Bui  
Laboratoire de Mecanique des Solides  
Ecole Polytechnique  
91128 Palaiseau Cedex  
France

Professor D. Cartwright  
Department of Mechanical Engineering  
Bucknell University  
Lewisburg University  
Pennsylvania 17837  
USA

Professor A.H-D. Cheng  
University of Delaware  
College of Engineering  
Department of Civil Engineering  
137 Dupont Hall  
Newark, Delaware 19716  
USA

Professor J.J. Connor  
Department of Civil Engineering  
Massachusetts Institute of Technology  
Cambridge  
MA 02139  
USA

**Professor J. Dominguez**  
Escuela Superior de Ingenieros Industriales  
Av. Reina Mercedes  
41012 Sevilla  
Spain

**Professor G.S. Gipson**  
School of Civil Engineering  
Engineering South 207  
Oklahoma State University  
Stillwater, OK 74078-0327  
USA

**Professor S. Grilli**  
The University of Rhode Island  
Department of Ocean Engineering  
Kingston, RI 02881-0814  
USA

**Professor D.B. Ingham**  
Department of Applied Mathematical Studies  
School of Mathematics  
The University of Leeds  
Leeds LS2 9JT  
UK

**Professor P. Molinaro**  
Ente Nazionale per l'Energia Elettrica  
Direzione Degli Studi e Ricerche  
Centro di Ricerca Idraulica e Strutturale  
Via Ornato 90/14  
20162 Milano  
Italy

**Professor Dr. K. Onishi**  
Department of Mathematics II  
Science University of Tokyo  
Wakamiya-cho 26  
Shinjuku-ku  
Tokyo 162  
Japan

**Professor H. Pina**  
Instituto Superior Tecnico  
Av. Rovisco Pais  
1096 Lisboa Codex  
Portugal

**Dr. A.P.S. Selvadurai**  
Department of Civil Engineering  
Room 277, C.J. Mackenzie Building  
Carleton University  
Ottawa  
Canada K1S 5B6

**Professor A. Giorgini**  
Purdue University  
School of Civil Engineering  
West Lafayette, IN 47907  
USA

**Professor W.G. Gray**  
Department of Civil Engineering and  
Geological Sciences  
University of Notre Dame  
Notre Dame, IN 46556  
USA

**Dr. S. Hernandez**  
Department of Mechanical Engineering  
University of Zaragoza  
Maria de Luna  
50015 Zaragoza  
Spain

**Professor G.D. Manolis**  
Aristotle University of Thessaloniki  
School of Engineering  
Department of Civil Engineering  
GR-54006, Thessaloniki  
Greece

**Dr. A.J. Nowak**  
Silesian Technical University  
Institute of Thermal Technology  
44-101 Gliwice  
Konarskiego 22  
Poland

**Professor P. Parreira**  
Departamento de Engenharia Civil  
Avenida Rovisco Pais  
1096 Lisboa Codex  
Portugal

**Professor D.P. Rooke**  
DRA (Aerospace Division)  
Materials and Structures Department  
R50 Building  
RAE Farnborough  
Hampshire GU14 GTD  
UK

**Professor R.P. Shaw**  
S.U.N.Y. at Buffalo  
Department of Civil Engineering  
School of Engineering and Applied Sciences  
212 Ketter Hall  
Buffalo, New York 14260  
USA

**Professor P. Skerget**  
University of Maribor  
Faculty of Technical Sciences  
YU-62000 Maribor  
Smetanova 17  
P.O. Box 224  
Yugoslavia

**Professor M.D. Trifunac**  
Department of Civil Engineering, KAP 216D  
University of Southern California  
Los Angeles, CA 90089-2531  
USA

**Dr P.P. Strona**  
Centro Ricerche Fiat S.C.p.A.  
Strada Torino, 50  
10043 Orbassano (TO)  
Italy

**Professor N.G. Zamani**  
University of Windsor  
Department of Mathematics and Statistics  
401 Sunset  
Windsor  
Ontario  
Canada N9B 3P4

# The Dual Reciprocity Boundary Element Method

P.W. Partridge  
C.A. Brebbia  
L.C. Wrobel

Computational Mechanics Publications  
Southampton Boston

*Co-published with*  
Elsevier Applied Science  
London New York



P.W. Partridge  
Wessex Institute of Technology  
Ashurst Lodge  
Ashurst  
Southampton  
SO4 2AA  
U.K.

(also from the Dept. of Civil Engineering,  
University of Brasilia, Brazil)

L.C. Wrobel  
Wessex Institute of Technology  
Ashurst Lodge  
Ashurst  
Southampton  
SO4 2AA  
U.K.

Co-published by

Computational Mechanics Publications  
Ashurst Lodge, Ashurst, Southampton, UK

Computational Mechanics Publications Ltd  
Sole Distributor in the USA and Canada:

Computational Mechanics Inc.  
25 Bridge Street, Billerica, MA 01821, USA

and

Elsevier Science Publishers Ltd  
Crown House, Linton Road, Barking, Essex IG11 8JU, UK

Elsevier's Sole Distributor in the USA and Canada:

Elsevier Science Publishing Company Inc.  
655 Avenue of the Americas, New York, NY 10010, USA

**British Library Cataloguing-in-Publication Data**

A Catalogue record for this book is available  
from the British Library

ISBN 1-85166-700-8 Elsevier Applied Science, London, New York

ISBN 1-85312-098-7 Computational Mechanics Publications, Southampton

ISBN 0-945824-82-3 Computational Mechanics Publications, Boston, USA

Library of Congress Catalog Card Number 91-70442

No responsibility is assumed by the Publishers for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

©Computational Mechanics Publications 1992

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the publisher.

C.A. Brebbia  
Wessex Institute of Technology  
Ashurst Lodge  
Ashurst  
Southampton  
SO4 2AA  
U.K.

# Contents

Preface	xv
<b>1 Introduction</b>	<b>1</b>
<b>2 The Boundary Element Method for Equations <math>\nabla^2 u = 0</math> and <math>\nabla^2 u = b</math></b>	<b>11</b>
2.1 Introduction . . . . .	11
2.2 The Case of the Laplace Equation . . . . .	13
2.2.1 Fundamental Relationships . . . . .	13
2.2.2 Boundary Integral Equations . . . . .	16
2.2.3 The Boundary Element Method for Laplace's Equation . . . . .	18
2.2.4 Evaluation of Integrals . . . . .	22
2.2.5 Linear Elements . . . . .	24
2.2.6 Treatment of Corners . . . . .	25
2.2.7 Quadratic and Higher-Order Elements . . . . .	28
2.3 Formulation for the Poisson Equation . . . . .	35
2.3.1 Basic Relationships . . . . .	35
2.3.2 Cell Integration Approach . . . . .	35
2.3.3 The Monte Carlo Method . . . . .	37
2.3.4 The Use of Particular Solutions . . . . .	41
2.3.5 The Galerkin Vector Approach . . . . .	43
2.3.6 The Multiple Reciprocity Method . . . . .	45
2.4 Computer Program 1 . . . . .	47
2.4.1 MAINP1 . . . . .	48
2.4.2 Subroutine INPUT1 . . . . .	51
2.4.3 Subroutine ASSEM2 . . . . .	54
2.4.4 Subroutine NECMOD . . . . .	57
2.4.5 Subroutine SOLVER . . . . .	58
2.4.6 Subroutine INTERM . . . . .	59
2.4.7 Subroutine OUTPUT . . . . .	60
2.4.8 Results of a Test Problem . . . . .	60
2.5 References . . . . .	66
<b>3 The Dual Reciprocity Method for Equations of the Type <math>\nabla^2 u = b(x, y)</math></b>	<b>69</b>
3.1 Equation Development . . . . .	69

3.1.1	Preliminary Considerations . . . . .	69
3.1.2	Mathematical Development of the DRM for the Poisson Equation	70
3.2	Different $f$ Expansions . . . . .	75
3.2.1	Case $f = r$ . . . . .	76
3.2.2	Case $f = 1 + r$ . . . . .	77
3.2.3	Case $f = 1$ at One Node and $f = r$ at Remaining Nodes . . . . .	77
3.3	Computer Implementation . . . . .	78
3.3.1	Schematized Matrix Equations . . . . .	78
3.3.2	Sign of the Components of $r$ and its Derivatives . . . . .	81
3.4	Computer Program 2 . . . . .	83
3.4.1	MAINP2 . . . . .	85
3.4.2	Subroutine INPUT2 . . . . .	86
3.4.3	Subroutine ALFAF2 . . . . .	88
3.4.4	Subroutine RHSVEC . . . . .	89
3.4.5	Comparison of Results for a Torsion Problem using Different Approximating Functions . . . . .	92
3.4.6	Data and Output for Program 2 . . . . .	94
3.5	Results for Different Functions $b = b(x, y)$ . . . . .	99
3.5.1	The Case $\nabla^2 u = -x$ . . . . .	99
3.5.2	The Case $\nabla^2 u = -x^2$ . . . . .	100
3.5.3	The Case $\nabla^2 u = a^2 - x^2$ . . . . .	101
3.5.4	Results using Quadratic Elements . . . . .	102
3.6	Problems with Different Domain Integrals on Different Regions . . . . .	102
3.6.1	The Subregion Technique . . . . .	102
3.6.2	Integration over Internal Region . . . . .	106
3.7	References . . . . .	107
<b>4</b>	<b>The Dual Reciprocity Method for Equations of the Type <math>\nabla^2 u = b(x, y, u)</math></b>	<b>109</b>
4.1	Introduction . . . . .	109
4.2	The Convective Case . . . . .	112
4.2.1	Results for the Case $\nabla^2 u = -\partial u / \partial x$ . . . . .	115
4.2.2	Results for the Case $\nabla^2 u = -(\partial u / \partial x + \partial u / \partial y)$ . . . . .	116
4.2.3	Internal Derivatives of the Problem Variables . . . . .	116
4.3	The Helmholtz Equation . . . . .	118
4.3.1	DRM Formulations . . . . .	119
4.3.2	DRM Results for Vibrating Beam . . . . .	121
4.3.3	Results for Non-Inversion DRM . . . . .	130
4.4	Non-Linear Cases . . . . .	131
4.4.1	Burger's Equation . . . . .	132
4.4.2	Spontaneous Ignition: The Steady-State Case . . . . .	134
4.4.3	Non-Linear Material Problems . . . . .	139
4.5	Computer Program 3 . . . . .	146
4.5.1	MAINP3 . . . . .	148

4.5.2	Subroutine ALFAF3 . . . . .	152
4.5.3	Subroutine RHSMAT . . . . .	153
4.5.4	Subroutine DERIVXY . . . . .	156
4.5.5	Results of Test Problems . . . . .	158
4.6	Three-Dimensional Analysis . . . . .	165
4.6.1	Equations of the Type $\nabla^2 u = b(x, y, z)$ . . . . .	166
4.6.2	Equations of the Type $\nabla^2 u = b(x, y, z, u)$ . . . . .	169
4.7	References . . . . .	171
<b>5</b>	<b>The Dual Reciprocity Method for Equations of the Type <math>\nabla^2 u = b(x, y, u, t)</math></b>	<b>175</b>
5.1	Introduction . . . . .	175
5.2	The Diffusion Equation . . . . .	176
5.3	Computer Program 4 . . . . .	177
5.3.1	MAINP4 . . . . .	179
5.3.2	Subroutine ASSEMB . . . . .	182
5.3.3	Subroutine VECTIN . . . . .	184
5.3.4	Subroutine BOUND <sub>C</sub> . . . . .	184
5.3.5	Results of a Test Problem . . . . .	185
5.3.6	Data Input . . . . .	188
5.3.7	Computer Output . . . . .	190
5.3.8	Further Applications . . . . .	195
5.3.9	Other Time-Stepping Schemes . . . . .	196
5.4	Special $f$ Expansions . . . . .	198
5.4.1	Axisymmetric Diffusion . . . . .	198
5.4.2	Infinite Regions . . . . .	199
5.5	The Wave Equation . . . . .	201
5.5.1	Infinite and Semi-Infinite Regions . . . . .	204
5.6	The Transient Convection-Diffusion Equation . . . . .	206
5.7	Non-Linear Problems . . . . .	207
5.7.1	Non-Linear Materials . . . . .	209
5.7.2	Non-Linear Boundary Conditions . . . . .	212
5.7.3	Spontaneous Ignition: Transient Case . . . . .	215
5.8	References . . . . .	220
<b>6</b>	<b>Other Fundamental Solutions</b>	<b>223</b>
6.1	Introduction . . . . .	223
6.2	Two-Dimensional Elasticity . . . . .	224
6.2.1	Static Analysis . . . . .	226
6.2.2	Treatment of Body Forces . . . . .	230
6.2.3	Dynamic Analysis . . . . .	232
6.3	Plate Bending . . . . .	238
6.4	Three-Dimensional Elasticity . . . . .	246
6.4.1	Computational Formulation . . . . .	246
6.4.2	Gravitational Load . . . . .	250

6.4.3	Centrifugal Load . . . . .	252
6.4.4	Thermal Load . . . . .	253
6.5	Transient Convection-Diffusion . . . . .	255
6.6	References . . . . .	263
<b>7</b>	<b>Conclusions</b>	<b>267</b>
	<b>Appendix 1</b>	<b>269</b>
	<b>Appendix 2</b>	<b>273</b>
	<b>The Authors</b>	<b>283</b>

## Preface

The boundary element method (BEM) is now a well-established numerical technique which provides an efficient alternative to the prevailing finite difference and finite element methods for the solution of a wide range of engineering problems. The main advantage of the BEM is its unique ability to provide a complete problem solution in terms of boundary values only, with substantial savings in computer time and data preparation effort.

An initial restriction of the BEM was that the fundamental solution to the original partial differential equation was required in order to obtain an equivalent boundary integral equation. Another was that non-homogeneous terms accounting for effects such as distributed loads were included in the formulation by means of domain integrals, thus making the technique lose the attraction of its “boundary-only” character.

Many different approaches have been developed to overcome these problems. It is our opinion that the most successful so far is the dual reciprocity method (DRM), which is the subject matter of this book. The basic idea behind this approach is to employ a fundamental solution corresponding to a simpler equation and to treat the remaining terms, as well as other non-homogeneous terms in the original equation, through a procedure which involves a series expansion using global approximating functions and the application of reciprocity principles.

The dual reciprocity procedure is completely general and is applied in this book to a large number of problems. The first two chapters of the book are introductory, and contain a review of the theory of boundary integral equations and boundary elements illustrated by application to Laplace's equation, and a discussion of different methods of treating such domain sources as appear in Poisson's equation. Chapter 3 presents the basic theory of the DRM, used in this case to approximate the domain integral resulting from known non-homogeneous terms in Poisson's equation. Chapter 4 generalizes the technique by applying it to problems which can be interpreted as described by a Poisson equation with a non-homogeneous term which is itself a function of the main variable. Linear and non-linear problems are discussed, including convective terms, the Helmholtz and Burger equations, and others. Chapter 5 treats transient problems such as diffusion and wave propagation, and shows that the same type of approximation can also be applied to the time-derivative term. Problems governed by more complex equations with different fundamental solutions are considered in

chapter 6, which deals with two-dimensional elasticity, plate bending and transient convection-diffusion.

Most of the material contained in this book is the result of our own research work, and our idea for collecting it together in book form was to draw the attention of the engineering and applied mathematics communities to this powerful technique. It is our hope that more and more researchers will get involved with the DRM, and to help them fully understand and appreciate how the mathematical theory develops into workable computer programs the book includes some simple codes written in modular form. The codes can be easily optimized and adapted by the reader to solve problems of his particular interest.

We would like to thank our colleagues and co-workers at Computational Mechanics for many helpful comments and suggestions. Special thanks are due to Dr. Andrew Nowak for his comments on the draft text. We are also indebted to Dr. Dubravko Nardini without whose original studies in elastodynamics, described in chapter 6, this book would not have been possible.

Southampton, 1991

The authors

# Chapter 1

## Introduction

Classical methods for solving continuum problems in engineering generally make use of some form of domain discretization - a grid in the case of the finite difference method, and a series of elements in the case of the finite element method.

Finite difference techniques approximate the derivatives in the differential equations which govern each problem using some type of truncated Taylor expansion and thus express them in terms of the values at a number of discrete mesh points. This results in a series of algebraic equations to which boundary conditions are applied in order to solve the problem. Although the internal discretization scheme generally employed in the finite difference approach is comparatively straightforward, the main difficulties of the technique lie in the consideration of curved geometries and the application of boundary conditions. For the case of general boundaries, the regular finite difference grid is unable to accurately reproduce the geometry of the problem. In addition, the introduction of boundary conditions involving derivatives requires the use of fictitious external points or the definition of a lower-order expansion which decreases the accuracy of the results.

On the more positive side, finite difference codes are comparatively economical to run because of the simplicity of matrix generation and manipulation. Because of this, they are still widely used in applications such as fluid dynamics which requires very refined meshes and a large number of repeated operations.

Finite element techniques were originally developed for solid mechanics applications in an effort to obtain a better representation of the geometry of the problem and to **simplify the introduction of the boundary conditions**. The method involves the approximation of the variables over small parts of the domain, called elements, in terms of polynomial interpolation functions. A weighted residual statement may be written in order to distribute the error introduced by this approximation over each element or alternatively this operation can be seen as the minimization of an energy functional. This results in influence matrices which express the properties of each element in terms of a discrete number of nodal values. Assembling all of these together produces a global matrix which represents the properties of the continuum. Application of the boundary conditions can then be carried out in a comparatively simple way. Not only can general boundary geometries be represented by elements

## 2 The Dual Reciprocity

with curved sides, but any derivative can be expressed in terms of the interpolation functions already proposed. The method is much more versatile than finite differences in that meshes can be easily graded and general types of boundary conditions are simple to incorporate.

The disadvantages of finite elements are that large quantities of data are required to discretize the full domain, particularly for three-dimensional problems, and that the technique sometimes gives inaccurate results. This is particularly so for cases of discontinuous functions, singularities or functions which vary rapidly. There are also difficulties when modelling infinite regions and moving boundary problems.

The boundary element method was developed as a response to the above difficulties. The method requires only discretization of the boundary thus reducing the quantity of data necessary to run a problem. Its basic idea consists of relating the variables at different boundary points by the use of analytical functions, *i.e.* the fundamental solution, resulting in a series of influence coefficients which can be arranged in matrix form; the boundary conditions are then applied in a similar way as done in finite elements. Boundary elements can be of a general type and retain the facility of modelling curved boundaries in spite of having reduced the dimensionality of the problem by one. Because of the problem formulation in terms of fundamental solutions, discontinuities and singularities can be modelled without special difficulties. As the solution is required only on the boundary, the technique is ideally suited for free and moving surface problems. Another important advantage of the method is that it can deal with problems extending to infinity without having to truncate the domain at a finite distance.

It is interesting to note that while finite differences involve only the approximation of the differential equations governing the problem, finite elements require integration by parts of the domain terms resulting from the representation of the variables using polynomial functions. In boundary elements, on the other hand, although the problem is expressed using only boundary integrals, these are more complex than those present in finite elements. Because of this, special integration techniques are necessary to obtain accurate results and much of the early work on the new approach concentrated on the computation of the integrals.

The origin of the boundary element method can be traced to the work carried out by small groups of researchers in the 1960's on the applications of boundary integral equations to potential flow and stress analysis problems. The first attempts were based on using a series of sources on the boundary, the values of which were assumed constant over a certain region or element. The source representation, as well as that based on dipoles, is now called the *indirect* boundary element approach. More recently, the *direct* approach, based on the use of physical variables such as potentials and fluxes or displacements and tractions, has surpassed the indirect techniques. The early, constant source applications of both the indirect and direct approaches had the disadvantage of giving inaccurate results in many applications, such as stress analysis. This difficulty was one of the factors that contributed to make the original boundary integral formulation unpopular with engineers and scientists.

The 1970's saw an ever increasing activity in boundary integral research and ap-

plications. By the middle of the decade it was becoming clear that the boundary integral approach offered an excellent alternative to finite elements for the solution of many practical problems. It was at this point that the idea of using curvilinear boundary elements with the representation of the variables in terms of interpolation functions was born. This offered a technique with the same degree of versatility as finite elements for representing the geometry of the problem. Many researchers also started to point out the advantages of using direct-type formulations instead of the indirect boundary integral approach.

In 1978 the first book with boundary elements in its title was published, and the first international conference on the topic organized. These events led to the clear understanding of the most important characteristics of the boundary element method, and the differences between it and the more classical boundary integral formulations, *i.e.*

1. The emphasis of boundary elements on mixed-type variational principles or weighted residual formulations to produce the direct integral equations.

The weighted residual formulation starts from the governing equations of the problem and is extremely convenient when working with non-linear or time-dependent problems. This approach facilitates the use of different types of error minimization on the boundary such as collocation, which is normally used, Galerkin, least squares, etc. The formulation is of a general nature and is not restricted to the use of Green's functions as fundamental solutions.

2. The method emphasizes the development of higher-order boundary elements, especially curved ones, which allow for the proper representation of the surface of general boundaries. Higher-order elements are important in many practical cases and their proper understanding is essential to obtain a representation of rigid body modes in stress analysis for instance, as well as to achieve convergence of the solution.
3. The lower degree of continuity required by boundary element formulations which admit the presence of discontinuities and help to better represent singularities. This important characteristic results in a simpler representation of complex boundary conditions and constraints without the high degree of continuity needed in finite elements or finite differences.

The international conferences on boundary elements organized practically every year since 1978 brought together all the new work on the subject and have contributed to the better understanding of the method. Numerous papers on non-linear and time-dependent problems have been presented at these meetings, many of them pointing out the difficulties of extending the technique to such applications. The main drawback in these cases was the need to discretize the domain into a series of internal cells to deal with the terms not taken to the boundary by application of the fundamental solution, such as non-linear terms. This additional discretization destroyed some of the attraction of the method in terms of the data required to run the problem and the complexity of the extra operations involved.

## 4 The Dual Reciprocity

It was then realized that a new approach was needed to deal with domain integrals in boundary elements. Several methods have been proposed by different authors. The most important of them are:

1. **Analytical Integration of the Domain Integrals.** This approach, although producing very accurate results, is only applicable to a limited number of cases for which the integrals can be evaluated analytically.
2. **The Use of Fourier Expansions.** The Fourier expansion method is not straightforward to apply in many cases as the calculation of the coefficients can be computationally cumbersome, although the method has been applied with some success to relatively simple cases.
3. **The Galerkin Vector Technique.** This approach uses a primitive, higher-order fundamental solution and Green's identity to transform certain types of domain integrals into equivalent boundary integrals. The main difficulty of the approach is that it can only solve comparatively simple cases. It has been extended to deal with other applications giving origin to the technique discussed in the next paragraph.
4. **The Multiple Reciprocity Method.** This is an extension of the Galerkin vector technique which utilizes as many higher-order fundamental solutions as required rather than using just one. The main difficulty is that the method cannot be easily applied to general non-linear problems although it has been successfully used to solve some time-dependent problems.
5. **The Dual Reciprocity Method.** This is the subject of the present book and constitutes the only general technique other than cell integration.

The Dual Reciprocity Method was introduced by Nardini and Brebbia in 1982 for elastodynamic problems and extended by Wrobel and Brebbia to time-dependent diffusion in 1986. The method was further extended to more general problems by Partridge and Brebbia and Partridge and Wrobel in 1989/90. A DRM bibliography is given at the end of the chapter.

The Dual Reciprocity Method is essentially a generalized way of constructing particular solutions that can be used to solve non-linear and time-dependent problems as well as to represent any internal source distribution. The method can be applied to define sources over the whole domain or only on part of it. The approach will be described in detail in the following chapters and its main advantage, *i.e.* its generality, will be amply demonstrated.

Chapter 2 of this book presents the boundary element method for Poisson-type equations and discusses several approaches for dealing with the non-homogeneous term, including the cell integration technique, the use of particular solutions, the Monte Carlo method, the Galerkin Vector approach and the Multiple Reciprocity technique. A computer code is included which can solve the Poisson equation using internal cells and the Laplace equation as a particular case.

The Dual Reciprocity Method is introduced in chapter 3 for Poisson-type equations in which the non-homogeneous term is a known function of space. The mathematical formulation of the method is explained in detail together with the different approximating functions which may be used to define the particular solution and the corresponding computer code implementation. The mechanism of application of the approach is described including the use of internal points or nodes. Many examples of applications are given using linear and quadratic elements.

In chapter 4 the application of the Dual Reciprocity Method is extended to cases in which the right-hand side of the governing equation is an unknown function of the problem variable as well as a function of space. This includes non-linear cases such as Burger's equation and spontaneous ignition as well as other applications such as convection and the Helmholtz equation, which would otherwise require the use of complex fundamental solutions or the subdivision of the domain into cells. The application of Dual Reciprocity is also extended to three-dimensional analysis. A computer code is given for a two-dimensional equation where the right-hand side is a function of the problem variable and position.

Chapter 5 deals with the application of the Dual Reciprocity Method to linear and non-linear time-dependent cases. This includes diffusion, wave propagation and convection-diffusion problems. A computer code is described for solution of the linear diffusion equation.

In chapter 6 the method is extended to problems for which a fundamental solution different from that of the Laplace equation is used. The Kelvin fundamental solution for two-dimensional elasticity, bi-harmonic fundamental solution for Kirchhoff's thin plate theory and the fundamental solution for the steady-state convection-diffusion equation with constant coefficients are considered, and guidelines are given as to how to use the method for problems which involve other fundamental solutions.

This book presents the state-of-the-art in the application of the Dual Reciprocity Boundary Element Method and describes in detail how the technique can be applied in practice. The state of development of the method is such that it may be used as a general tool for boundary element analysis.

## Selected DRM Bibliography

### 1. Nardini, D. and Brebbia, C. A.

A New Approach to Free Vibration Analysis using Boundary Elements, in *Boundary Element Methods in Engineering*, Computational Mechanics Publications, Southampton, and Springer-Verlag, Berlin and New York, 1982.

### 2. Nardini, D. and Brebbia, C. A.

Transient Dynamic Analysis by the Boundary Element Method, in *Boundary Elements V*, Computational Mechanics Publications, Southampton, and Springer-Verlag, Berlin and New York, 1983.

### 3. Brebbia, C. A. and Nardini, D.

Dynamic Analysis in Solid Mechanics by an Alternative Boundary Element Procedure,

## 6 The Dual Reciprocity

*International Journal of Soil Dynamics and Earthquake Engineering*, Vol. 2, No. 4, pp 228-233, 1983.

4. Brebbia, C. A.  
The Solution of Time Dependent Problems using Boundary Elements, in *The Mathematics of Finite Elements V*, Academic Press, London, 1985.
5. Nardini, D. and Brebbia, C. A.  
Applications of an Alternative Boundary Element Procedure in Elastodynamics, in *Variational Methods in Engineering*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1985.
6. Nardini, D. and Brebbia, C. A.  
Boundary Integral Formulation of Mass Matrices for Dynamic Analysis, in *Topics in Boundary Element Research*, Vol. 2, Springer-Verlag, Berlin and New York, 1985.
7. Nardini, D. and Brebbia, C. A.  
The Solution of Parabolic and Hyperbolic Problems using an Alternative Boundary Element Formulation, in *Boundary Elements VII*, Vol. 1, Computational Mechanics Publications, Southampton, and Springer-Verlag, Berlin and New York, 1985.
8. Wrobel, L. C., Nardini, D. and Brebbia, C. A.  
Analysis of Transient Thermal Problems in the BEASY System, in *BETECH/86*, Computational Mechanics Publications, Southampton, 1986.
9. Nardini, D. and Brebbia, C. A.  
Transient Boundary Element Elastodynamics using the Dual Reciprocity Method and Modal Superposition, in *Boundary Elements VIII*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
10. Brebbia, C. A.  
Boundary Element Solution of Elastodynamic Problems using a New Technique, in *Innovative Numerical Methods in Engineering*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
11. Wrobel, L. C., Brebbia, C. A. and Nardini, D.  
The Dual Reciprocity Boundary Element Formulation for Transient Heat Conduction, in *Finite Elements in Water Resources VI*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
12. Niku, S. M. and Brebbia, C. A.  
Dual Reciprocity Boundary Element Formulation for Potential Problems with Arbitrarily Distributed Forces, Technical Note, *Engineering Analysis*, Vol. 5, No. 1, pp 46-48, 1988.
13. Wrobel, L. C. and Brebbia, C. A.  
The Dual Reciprocity Boundary Element Formulation for Non-Linear Diffusion Problems, *Computer Methods in Applied Mechanics and Engineering*, Vol. 65, No. 2, pp 147-164, 1987.

14. Wrobel, L. C., Telles, J. C. F. and Brebbia, C. A.  
A Dual Reciprocity Boundary Element Formulation for Axisymmetric Diffusion Problems, in *Boundary Elements VIII*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
15. Brebbia, C. A. and Wrobel, L. C.  
Non-linear Transient Thermal Analysis using the Dual Reciprocity Method, in *BETECH/87*, Computational Mechanics Publications, Southampton, 1987.
16. Partridge, P. W. and Brebbia, C. A.  
Computer Implementation of the BEM Dual Reciprocity Method for the Solution of Poisson Type Equations, *Software for Engineering Workstations*, Vol. 5, No. 4, pp 199-206, 1989.
17. Partridge, P. W. and Brebbia, C. A.  
Computer Implementation of the BEM Dual Reciprocity Method for the Solution of General Field Problems, *Communications in Applied Numerical Methods*, Vol. 6, No. 2, pp 83-92, 1990.
18. Partridge, P. W. and Brebbia, C. A.  
The BEM Dual Reciprocity Method for Diffusion Problems, in *Computational Methods in Water Resources VIII*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1990.
19. Partridge, P. W. and Wrobel, L. C.  
The Dual Reciprocity Boundary Element Method for Spontaneous Ignition, *International Journal for Numerical Methods in Engineering*, Vol. 30, 1990.
20. Partridge, P. W. and Brebbia, C. A.  
On Derivatives of the Problem Unknowns in BEM Analysis, Technical Note, *Engineering Analysis*, Vol. 7, No. 1, pp 50-52, 1990.
21. Partridge, P. W. and Brebbia, C. A.  
The Dual Reciprocity Boundary Element Method for the Helmholtz Equation, in *European Boundary Element Symposium*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1990.
22. Brebbia, C. A.  
On the Treatment of Domain Integrals in Boundary Elements, in *BETECH/89*, Computational Mechanics Publications, Southampton, 1989.
23. Brebbia, C. A.  
On Two Different Methods for Transforming Domain Integrals to the Boundary, in *Boundary Elements XI*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.
24. Loeffler, C. F. and Mansur, W. J.  
Dual Reciprocity Boundary Element Formulation for Transient Elastic Wave Propagation in Infinite Domains, in *Boundary Elements XI*, Vol. 2, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.

## 8 The Dual Reciprocity

25. Loeffler, C. F. and Mansur, W. J.  
Dual Reciprocity Boundary Element Formulation for Potential Problems in Infinite Domains, in *Boundary Elements X*, Vol. 2, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.
26. Kamiya, N. and Sawaki, Y.  
The Plate Bending Analysis by the Dual Reciprocity Boundary Elements, *Engineering Analysis*, Vol. 5, No. 1, pp 36-40, 1988.
27. Sawaki, Y., Takeuchi, K. and Kamiya, N.  
Finite Deflection Analysis of Plates by Dual Reciprocity Boundary Elements, in *Boundary Elements XI*, Vol. 3, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.
28. Silva, N. A. and Venturini, W. S.  
Dual Reciprocity Process Applied to Solve Bending Plates on Elastic Foundations, in *Boundary Elements X*, Vol. 3, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.
29. Singh, K.M. and Kalra, M. S.  
A Least Squares Time Integration Scheme for the Dual Reciprocity BEM in Transient Heat Conduction, in *Numerical Methods in Thermal Problems VI*, Vol. 1, Pineridge Press, Swansea, 1989.

## Further Reading

### Boundary Element Methods Conference Proceedings

1. Brebbia, C. A. (Ed)  
Recent Advances in Boundary Element Methods, Pentech Press, London, 1978.
2. Brebbia, C. A. (Ed)  
New Developments in Boundary Element Methods, Computational Mechanics Publications, Southampton, 1980.
3. Brebbia, C. A. (Ed)  
Boundary Element Methods, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1981.
4. Brebbia, C. A. (Ed)  
Boundary Element Methods in Engineering, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1982.
5. Brebbia, C. A., Futagami, T. and Tanaka, M. (Eds)  
Boundary Elements V, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1983.
6. Brebbia, C. A. (Ed)  
Boundary Elements VI, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1984.

7. Brebbia, C. A. and Maier, G. (Eds)  
Boundary Elements VII, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1985.
8. Tanaka, M. and Brebbia, C. A. (Eds)  
Boundary Elements VIII, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
9. Brebbia, C. A., Wendland, W. L. and Kuhn, G. (Eds)  
Boundary Elements IX, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1987.
10. Brebbia, C. A. (Ed)  
Boundary Elements X, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.
11. Brebbia, C. A. and Connor, J.J. (Eds)  
Boundary Elements XI, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.
12. Brebbia, C. A. Tanaka, M. and Honma, T. (Eds)  
Boundary Elements XII, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1990.

## Boundary Element Technology Conference Proceedings

1. Brebbia, C. A. and Noye, B. J. (Eds)  
BETECH/85, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1985.
2. Brebbia, C. A. and Connor, J. J. (Eds)  
BETECH/86, Computational Mechanics Publications, Southampton, 1986.
3. Brebbia, C. A. and Venturini, W. S. (Eds)  
BETECH/87, Computational Mechanics Publications, Southampton, 1987.
4. Brebbia, C. A. and Zamani, N. G. (Eds)  
BETECH/89, Computational Mechanics Publications, Southampton, 1989.
5. Cheng, A. H. D., Brebbia, C. A. and Grilli, S. (Eds)  
BETECH/90, Computational Mechanics Publications, Southampton, 1990.

## Other BEM Meetings

1. Brebbia, C. A. and Chaudouet-Miranda, A. (Eds)  
European Boundary Element Symposium, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1990.
2. Tanaka, M., Brebbia, C. A. and Shaw, R. (Eds)  
USA/Japan BEM Seminar, Computational Mechanics Publications, Southampton, 1990.

## State-of-the-Art Books

1. Brebbia, C. A. (Ed)  
Progress in Boundary Element Methods, Vol. I, Pentech Press, London, 1981.
2. Brebbia, C. A. (Ed)  
Progress in Boundary Element Methods, Vol. II, Pentech Press, London, 1983.
3. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. I: Basic Principles and Applications, Springer-Verlag, Berlin and New York, 1984.
4. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. II: Time-Dependent and Vibration Problems, Springer-Verlag, Berlin and New York, 1985.
5. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. III: Mathematical Aspects, Springer-Verlag, Berlin and New York, 1987.
6. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. IV: Applications in Geomechanics, Springer-Verlag, Berlin and New York, 1987.
7. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. V: Viscous Flow Applications, Springer-Verlag, Berlin and New York, 1989.
8. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. VI: Electromagnetic Applications, Springer-Verlag, Berlin and New York, 1989.
9. Brebbia, C. A. (Ed)  
Topics in Boundary Element Research, Vol. VII: Electrostatics Applications, Springer-Verlag, Berlin and New York, 1990.
10. Wrobel, L. C. and Brebbia, C. A. (Eds)  
Advances in Boundary Element Methods for Heat Transfer, Computational Mechanics Publications, Southampton, 1991.
11. Cisowski, R. D. and Schenck, H. A. (Eds)  
Advances in Boundary Element Methods for Acoustics, Computational Mechanics Publications, Southampton, 1991.
12. Aliabadi, M. (Ed)  
Advances in Boundary Element Methods for Fracture Mechanics, Computational Mechanics Publications, Southampton, 1991.

# Chapter 2

## The Boundary Element Method for the Equations $\nabla^2 u = 0$ and $\nabla^2 u = b$

### 2.1 Introduction

It is generally agreed that the modern definition of Boundary Elements dates from 1978 when the notation used nowadays was accepted, partly due to the publication of the Proceedings of the first international conference on Boundary Elements [1] and partly as a result of the first book on the topic by the Computational Mechanics research group at Southampton [2]. Other books by the Southampton group have been published since, all of them using the same notation [3,4,5] which, with minor changes, will be that used in the present book.

Boundary elements are usually associated with direct formulations in which the problem unknowns are the physical variables, for instance potential or fluxes, what is more attractive to users than the indirect formulations, used mainly before 1978, which involve concepts such as sources or dipoles. The direct formulation is somewhat more general, elegant and simple to apply in computer codes, hence the authors have adopted it in this book. Furthermore, it is easy to deduce the indirect from the direct approach if desired (see [2] and [4]).

The boundary integral equation formulation of potential problems can be traced to Jaswon [6] and Symm [7] who as early as 1963 presented a numerical method to solve Fredholm-type integral equations. They discretized the boundary of the problems into a number of segments or elements and assumed a constant source density upon each of them. A collocation technique was employed to generate a series of integral equations and the influence coefficients were computed using Simpson's rule with the exception of the singular coefficients which were computed analytically or by the summation of the off-diagonal terms. In their pioneering work they even proposed a more general formulation using potentials and derivatives as unknowns (*i.e.* direct approach) and results were reported for some applications [6,7,8]. All the bases for

## 12 The Dual Reciprocity

it deserved, probably due to the simultaneous emergence of the finite element method.

Since 1978 the popularity of the Boundary Element Method has steadily increased. The interpretation of the method in terms of variational or weighted residual type statements helped to clarify its relationship to other numerical techniques such as finite differences and finite elements. At the same time higher-order elements started to be developed following the work on Jacobian-type transformations first used in finite elements. Many researchers also became aware of the importance of accurate numerical integration techniques and investigation started to be carried out in the development of approaches suitable for singular and quasi-singular integrands.

In this chapter the basic formulation of the method for solving the Laplace and Poisson equations is presented. The first part of the chapter, section 2.2, concentrates on developing the basic integral techniques for solving the Laplace equation. Initially, the general integral equation is deduced from the governing partial differential equation and boundary conditions; then, a boundary integral equation is obtained using the fundamental solution and Green's theorem. The properties of the fundamental solutions are demonstrated. The problem is then expressed in matrix form, explaining how the boundary element influence coefficients are obtained and the boundary conditions applied. The cases of linear, quadratic and cubic elements are discussed together with the simpler constant element formulation.

Section 2.3 deals with the solution of Poisson's equation describing some techniques other than the Dual Reciprocity Method. The simplest approach, the cell integration technique, is discussed in section 2.3.2. The method consists of subdividing the region into a series of internal cells, on each of which a numerical integration technique is then applied. Although the method is simple to use, it has the disadvantage of requiring integration over the domain rather than only on the boundary. Because of this, other approaches have been proposed to simplify the evaluation of those integrals. The Monte Carlo method (section 2.3.3) is a particularly interesting approach as it consists of defining a series of random integration points rather than applying a regular integration grid. The most efficient way to simplify the problem however is to reduce the domain integrals to the boundary and this can be easily accomplished in many cases by using particular integrals (section 2.3.4). This technique is however limited to problems in which the particular solution is known, although it can also be seen as the basis for the Dual Reciprocity Method studied in this book, as will be shown in chapter 3. Another interesting possibility is to use a higher-order fundamental solution which transforms the domain sources to the boundary by a repeated integration-by-parts procedure. This is the basis of the so-called Galerkin Vector approach (section 2.3.5) which is valid for the case of harmonic functions only. More recently a generalization of this technique has been developed by Nowak and Brebbia [9,10,11] resulting in a new method called the Multiple Reciprocity Method (section 2.3.6). The MRM employs a set of higher-order fundamental solutions (rather than just one as in the Galerkin Vector method) and this permits the application of Green's identity to each term of the sequence in succession. As a result the method has the important advantage that it can lead in the limit to the exact boundary-only formulation of the problem.

In section 2.4 a computer program is presented which solves Poisson-type prob-

lems using linear boundary elements, with integration over the domain carried out using internal cells. The same program can be used to solve the Laplace equation as a particular case. The program is developed in modular form and many of the subroutines will be used again in the DRM programs to be given in chapters 3, 4 and 5. The results of most of the illustrative examples presented in this chapter may be obtained using this program.

## 2.2 The Case of the Laplace Equation

### 2.2.1 Fundamental Relationships

The starting boundary integral equation required by the method can be deduced in a simple way based on considerations of weighted residuals, Betti's reciprocal theorem, Green's third identity or fundamental principles such as virtual work. The advantage of using a weighted residual technique is its generality; it permits the extension of the method to the solution of more complex partial differential equations and can also be used to relate boundary elements to other numerical techniques.

Consider that we are seeking the solution of a Laplace equation in a two- or three-dimensional domain (figure 2.1)

$$\nabla^2 u = 0 \text{ in } \Omega \quad (2.1)$$

with the following boundary conditions:

- (i) "Essential" conditions of the type  $u = \bar{u}$  on  $\Gamma_1$
  - (ii) "Natural" conditions such as  $q = \partial u / \partial n = \bar{q}$  on  $\Gamma_2$
- (2.2)

where  $n$  is the outward normal to the boundary,  $\Gamma = \Gamma_1 + \Gamma_2$  and the bars indicate known values. More complex boundary conditions such as combinations of the above, *i.e.*

$$\alpha u + \beta q = \gamma \quad (2.3)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are known parameters, can be easily included [4] but they will not be considered here for the sake of simplicity.

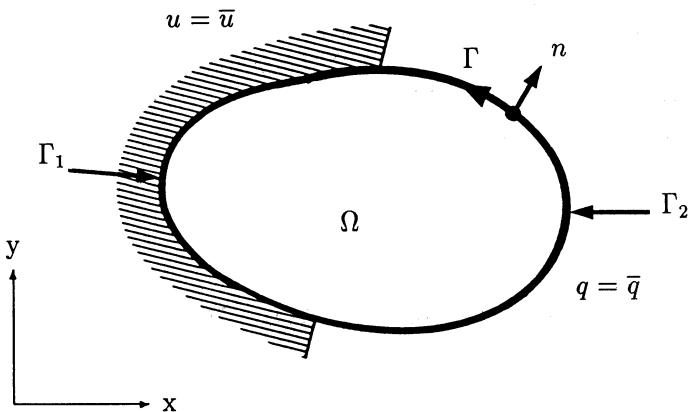


Figure 2.1: Geometric Definitions of the Problem

In principle, the errors introduced in the above equations if the exact (but unknown) values of  $u$  and  $q$  are replaced by an approximate solution, can be minimized by orthogonalizing them with respect to a weighting function  $u^*$ , the normal derivative of which along the boundary is  $q^* = \partial u^*/\partial n$ .

In other words, if  $R$  are the residuals, one can write in general that

$$\begin{aligned} R &= \nabla^2 u \neq 0 \text{ in } \Omega & (2.4) \\ R_1 &= u - \bar{u} \neq 0 \text{ on } \Gamma_1 \\ R_2 &= q - \bar{q} \neq 0 \text{ on } \Gamma_2 \end{aligned}$$

where  $u$  and  $q$  are approximate values (the fact that one or more of the residuals may be identically zero does not detract from the generality of the argument).

The above residuals can be weighted by the functions  $u^*$  and  $q^*$  as follows

$$\int_{\Omega} Ru^* d\Omega = \int_{\Gamma_2} R_2 u^* d\Gamma - \int_{\Gamma_1} R_1 q^* d\Gamma \quad (2.5)$$

or

$$\int_{\Omega} (\nabla^2 u) u^* d\Omega = \int_{\Gamma_2} (q - \bar{q}) u^* d\Gamma - \int_{\Gamma_1} (u - \bar{u}) q^* d\Gamma \quad (2.6)$$

The objective of this procedure is to force the residuals to be zero in an average sense. The sign of the different terms will become clear during the process of integration by parts. Integrating once the Laplacian in (2.6) gives

$$-\int_{\Omega} \frac{\partial u}{\partial x_k} \frac{\partial u^*}{\partial x_k} d\Omega = -\int_{\Gamma_2} \bar{q} u^* d\Gamma - \int_{\Gamma_1} q u^* d\Gamma - \int_{\Gamma_1} u q^* d\Gamma + \int_{\Gamma_1} \bar{u} q^* d\Gamma \quad (2.7)$$

using the so-called Einstein's summation convention for repeated indices, with  $k = 1, 2, 3$ .

Integrating by parts again the terms on the left-hand side one obtains,

$$\int_{\Omega} (\nabla^2 u^*) u d\Omega = - \int_{\Gamma_2} \bar{q} u^* d\Gamma - \int_{\Gamma_1} q u^* d\Gamma + \int_{\Gamma_2} u q^* d\Gamma + \int_{\Gamma_1} \bar{u} q^* d\Gamma \quad (2.8)$$

This important equation is the starting point for the application of the boundary element method. Our aim is now to transform formula (2.8) into a boundary integral equation. This is done by using a special type of weighting function  $u^*$  called the fundamental solution.

### Fundamental Solution

The fundamental solution  $u^*$  represents the field generated by a concentrated unit source acting at a point  $i$ . The effect of this source is propagated from  $i$  to infinity without any consideration of boundary conditions. Because of this,  $u^*$  satisfies the following Poisson equation,

$$\nabla^2 u^* + \Delta_i = 0 \quad (2.9)$$

where  $\Delta_i$  represents a Dirac delta function which goes to infinity at the point  $x = x_i$  and is equal to zero elsewhere. The integral of  $\Delta_i$  over the domain is equal to one. The use of the Dirac delta function is an elegant way of representing unit concentrated sources and forces when dealing with differential equations.

The integral of a Dirac delta function multiplied by any other function is equal to the value of the latter at the point  $i$ . Hence

$$\int_{\Omega} u (\nabla^2 u^*) d\Omega = \int_{\Omega} u (-\Delta_i) d\Omega = -u_i \quad (2.10)$$

Equation (2.8) can now be written as

$$u_i + \int_{\Gamma_2} u q^* d\Gamma + \int_{\Gamma_1} \bar{u} q^* d\Gamma = \int_{\Gamma_2} \bar{q} u^* d\Gamma + \int_{\Gamma_1} q u^* d\Gamma \quad (2.11)$$

It needs to be remembered that equation (2.11) applies to a concentrated source at  $i$  and consequently the values of  $u^*$  and  $q^*$  are those corresponding to that particular position of the source point. For each different position a new integral equation is obtained.

For an isotropic three-dimensional medium the fundamental solution of equation (2.9) is

$$u^* = \frac{1}{4\pi r} \quad (2.12)$$

and for a two-dimensional isotropic medium

$$u^* = \frac{1}{2\pi} \ln \left( \frac{1}{r} \right) \quad (2.13)$$

## 16 The Dual Reciprocity

where  $r$  is the distance from the point  $i$  of application of the concentrated source to any other point under consideration.

It is easy to check that solutions (2.12) and (2.13) satisfy the three- and two-dimensional Laplace equations. In order to demonstrate this, spherical coordinates can be used for simplicity in three-dimensional problems and this allows one to neglect terms which are zero due to the symmetry of the solution, i.e.

$$\nabla^2 u^* \rightarrow \frac{\partial^2 u^*}{\partial r^2} + \frac{2}{r} \frac{\partial u^*}{\partial r} = -\Delta; \quad (2.14)$$

For the case where  $r \equiv 0$  one can prove that

$$\int_{\Omega} (\nabla^2 u^*) d\Omega = \int_{\Omega} -\Delta_i d\Omega = -1 \quad (2.15)$$

This can be demonstrated by integrating around a sphere of radius  $\epsilon$  and then taking the limit as  $\epsilon$  goes to zero. Considering that the sphere has a domain  $\Omega_\epsilon$  it is possible to integrate by parts to express the Laplacian in terms of boundary fluxes  $\partial u^*/\partial n$ , i.e.

$$\int_{\Omega_\epsilon} (\nabla^2 u^*) d\Omega = \int_{\Gamma_\epsilon} \frac{\partial u^*}{\partial n} d\Gamma = \int_{\Gamma_\epsilon} \frac{\partial u^*}{\partial r} d\Gamma \quad (2.16)$$

since  $n \equiv r$  on the surface of the sphere.

Substituting now the fundamental solution (2.12) into (2.16) and making  $r$  (or  $\epsilon$ ) tend to zero gives

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} \frac{\partial u^*}{\partial r} d\Gamma \right\} &= \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} -\frac{1}{4\pi\epsilon^2} d\Gamma \right\} = \\ &= \lim_{\epsilon \rightarrow 0} \left\{ -\frac{4\pi\epsilon^2}{4\pi\epsilon^2} \right\} = -1 \end{aligned} \quad (2.17)$$

Notice that the surface of the sphere has an area  $\Gamma_\epsilon = 4\pi\epsilon^2$ . Similarly, for the two-dimensional case, one can define a small circle of radius  $\epsilon$  and then take the limit when  $\epsilon \rightarrow 0$ , i.e.

$$\begin{aligned} \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} \frac{\partial u^*}{\partial n} d\Gamma \right\} &= \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} -\frac{1}{2\pi\epsilon} d\Gamma \right\} = \\ &= \lim_{\epsilon \rightarrow 0} \left\{ -\frac{2\pi\epsilon}{2\pi\epsilon} \right\} = -1 \end{aligned} \quad (2.18)$$

Here the perimeter of the small circle is  $\Gamma_\epsilon = 2\pi\epsilon$ .

### 2.2.2 Boundary Integral Equations

We have now deduced an integral equation (2.11) which is valid for any point within the domain  $\Omega$ . In boundary elements it is usually preferable for computational reasons to apply equation (2.11) on the boundary and hence it is necessary to find out what happens when the point  $i$  is on  $\Gamma$ . A simple way to do this is to consider that the point

$i$  is on the boundary but the domain itself is augmented by a hemisphere of radius  $\epsilon$  (in 3D) as is shown in figure 2.2 (i) (for 2D the same applies, but a semicircle is considered instead, figure 2.2 (ii)). Point  $i$  is considered to be at the center and then the limit as the radius  $\epsilon$  tends to zero is evaluated. The point will then become again a boundary point and the resulting expression will be the specialization of (2.11) for a point on  $\Gamma$ . At present smooth surfaces as represented in figure 2.2 will be considered, the case of corners will be discussed in section 2.2.6.

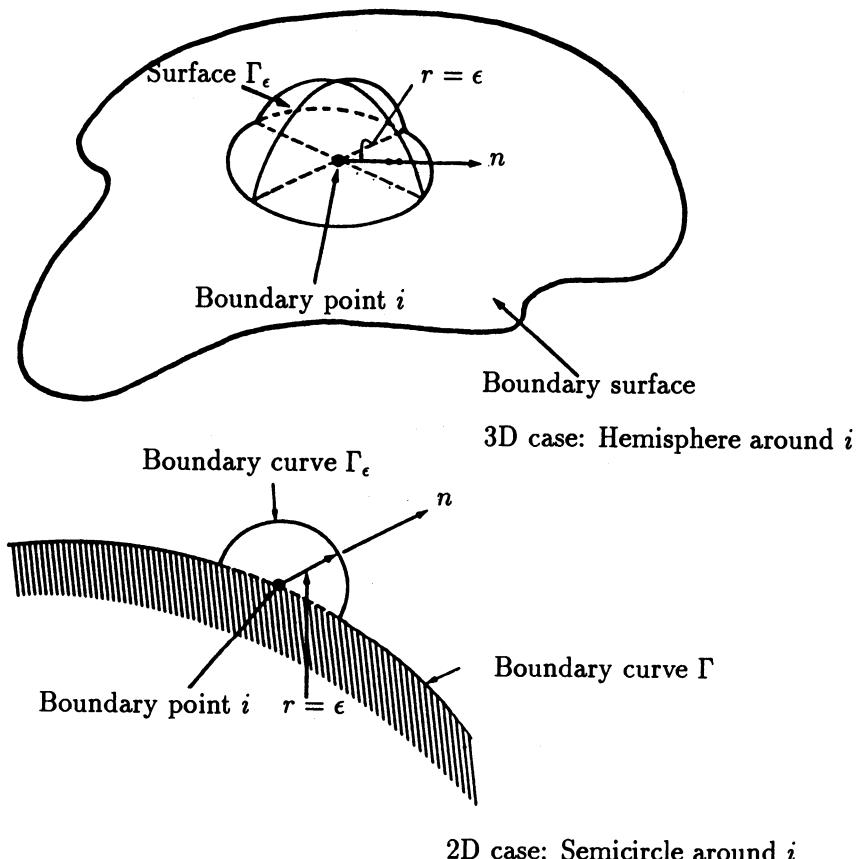


Figure 2.2: Boundary Points for Two- and Three-dimensional Cases, Augmented by a Small Hemisphere or Semicircle.

It is important at this stage to differentiate between two types of boundary integrals in (2.11) as the fundamental solution and its normal derivative behave differently. Consider for the sake of simplicity equation (2.11) before any boundary conditions have been applied, *i.e.*

$$u_i + \int_{\Gamma} u q^* d\Gamma = \int_{\Gamma} u^* q d\Gamma \quad (2.19)$$

## 18 The Dual Reciprocity

Here  $\Gamma = \Gamma_1 + \Gamma_2$  and satisfaction of the boundary conditions will be left for a later stage.

Integrals of the type shown on the right-hand side of (2.19) are easy to deal with as they present a lower order of singularity, *i.e.* for three-dimensional cases the integral around  $\Gamma_\epsilon$  gives:

$$\begin{aligned}\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} q u^* d\Gamma \right\} &= \lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} q \frac{1}{4\pi\epsilon} d\Gamma \right\} = \\ &= \lim_{\epsilon \rightarrow 0} \left\{ q \frac{2\pi\epsilon^2}{4\pi\epsilon} \right\} \equiv 0\end{aligned}\quad (2.20)$$

In other words, the right-hand side integral is continuous when expressions such as (2.11) or (2.19) are taken to the boundary. The left-hand side integral, however, behaves in a different manner. Here we have around  $\Gamma_\epsilon$  the following result,

$$\begin{aligned}\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} u q^* d\Gamma \right\} &= \lim_{\epsilon \rightarrow 0} \left\{ - \int_{\Gamma_\epsilon} u \frac{1}{4\pi\epsilon^2} d\Gamma \right\} = \\ &= \lim_{\epsilon \rightarrow 0} \left\{ -u \frac{2\pi\epsilon^2}{4\pi\epsilon^2} \right\} = -\frac{1}{2} u_i\end{aligned}\quad (2.21)$$

This means that the integral has a discontinuity or jump on the boundary, and produces what is called a free term. It is easy to check that the same will occur for the two-dimensional problem in which case the right-hand side integral around  $\Gamma_\epsilon$  is also identically zero and the left-hand side integral becomes

$$\begin{aligned}\lim_{\epsilon \rightarrow 0} \left\{ \int_{\Gamma_\epsilon} u q^* d\Gamma \right\} &= \lim_{\epsilon \rightarrow 0} \left\{ - \int_{\Gamma_\epsilon} u \frac{1}{2\pi\epsilon} d\Gamma \right\} = \\ &= \lim_{\epsilon \rightarrow 0} \left\{ -u \frac{\pi\epsilon}{2\pi\epsilon} \right\} = -\frac{1}{2} u_i\end{aligned}\quad (2.22)$$

Using (2.20) to (2.22) one can write the following expression for two- or three-dimensional problems when the point  $i$  is on the boundary

$$\frac{1}{2} u_i + \int_{\Gamma} u q^* d\Gamma = \int_{\Gamma} q u^* d\Gamma \quad (2.23)$$

where the integrals are in the sense of Cauchy Principal Values. This is the boundary integral equation generally used as the starting point for the boundary element formulation.

### 2.2.3 The Boundary Element Method for Laplace's Equation

Let us now consider how expression (2.23) can be discretized to find the system of equations from which the boundary values can be found. Assume for simplicity that the body is two-dimensional and its boundary is divided into  $N$  segments or elements as shown in figure 2.3.

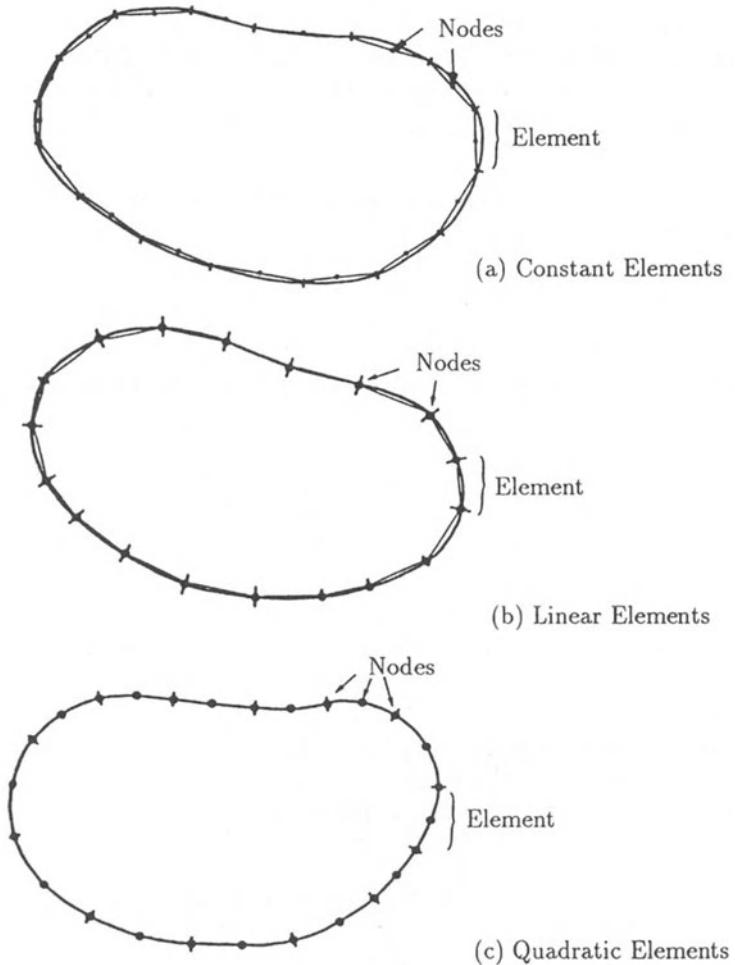


Figure 2.3: Different Types of Boundary Elements

The points where the unknown values are considered are called “nodes” and taken to be in the middle of the element for the so-called constant elements (figure 2.3(a)). These are the elements considered in this section, but later on the case of linear elements, *i.e.* those elements for which the nodes are at the extremes or ends (figure 2.3(b)) and curved elements such as the quadratic ones shown in figure 2.3(c) and for which a further mid-element node is required will be discussed.

The boundary is assumed to be divided into  $N$  elements. Equation (2.23) can be discretized for a given point  $i$  before applying any boundary conditions, as follows

$$\frac{1}{2}u_i + \sum_{j=1}^N \int_{\Gamma_j} u q^* d\Gamma = \sum_{j=1}^N \int_{\Gamma_j} q u^* d\Gamma \quad (2.24)$$

## 20 The Dual Reciprocity

In the case of the constant elements the values of  $u$  and  $q$  are assumed to be constant over each element and equal to the value at the mid-element node. The points at the extremes of the elements are used only for defining the geometry of the problem. Note that for this type of element (*i.e.* constant) the boundary is always “smooth” at the nodes as these are located at the center of the elements, hence the multiplier of  $u_i$  is always  $\frac{1}{2}$ .

The  $u$  and  $q$  values can thus be taken out of the integrals. They will be called  $u_j$  and  $q_j$  for element  $j$ . Hence,

$$\frac{1}{2}u_i + \sum_{j=1}^N u_j \int_{\Gamma_j} q^* d\Gamma = \sum_{j=1}^N q_j \int_{\Gamma_j} u^* d\Gamma \quad (2.25)$$

Notice that there are now two types of integrals to be carried out over the elements, *i.e.*

$$\int_{\Gamma_j} q^* d\Gamma \quad \text{and} \quad \int_{\Gamma_j} u^* d\Gamma$$

These integrals relate the node  $i$  where the fundamental solution is applied to any other node  $j$ . Because of this, their resulting values are sometimes called influence coefficients. We shall call them  $\bar{H}_{ij}$  and  $G_{ij}$ , *i.e.*

$$\bar{H}_{ij} = \int_{\Gamma_j} q^* d\Gamma; \quad G_{ij} = \int_{\Gamma_j} u^* d\Gamma \quad (2.26)$$

Notice that one is assuming throughout that the fundamental solution is applied at a particular node  $i$ , although this is not explicitly indicated in the  $u^*$ ,  $q^*$  notation to avoid proliferation of indices. Hence, for a particular point  $i$ , one can write

$$\frac{1}{2}u_i + \sum_{j=1}^N \bar{H}_{ij}u_j = \sum_{j=1}^N G_{ij}q_j \quad (2.27)$$

Let us now call

$$H_{ij} = \bar{H}_{ij} + \frac{1}{2}\delta_{ij} \quad (2.28)$$

where  $\delta$  is the Kronecker delta, in such a way that the  $\frac{1}{2}$  value is summed to  $\bar{H}$  when  $i = j$ , then equation (2.27) can now be written as

$$\sum_{j=1}^N H_{ij}u_j = \sum_{j=1}^N G_{ij}q_j \quad (2.29)$$

If it is now assumed that the position of node  $i$  also varies from 1 to  $N$ , *i.e.* one assumes that the fundamental solution is applied at each node successively, a system of equations is obtained resulting from the application of (2.29) to each boundary point in turn.

This set of equations can be expressed in matrix form as

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q} \quad (2.30)$$

where  $\mathbf{H}$  and  $\mathbf{G}$  are two  $N \times N$  matrices and  $\mathbf{u}$  and  $\mathbf{q}$  are vectors of length  $N$ .

Notice that  $N_1$  values of  $u$  and  $N_2$  values of  $q$  are known on  $\Gamma_1$  and  $\Gamma_2$  respectively ( $N_1 + N_2 = N$ ), hence there are only  $N$  unknowns in the system of equations (2.30). To introduce these boundary conditions into (2.30) one has to rearrange the system by moving columns of  $\mathbf{H}$  and  $\mathbf{G}$  from one side to the other. Once all unknowns are passed to the left-hand side one can write

$$\mathbf{Ax} = \mathbf{y} \quad (2.31)$$

where  $\mathbf{x}$  is a vector of unknown boundary values of  $u$  and  $q$ , and  $\mathbf{y}$  is found by multiplying the corresponding columns of  $\mathbf{H}$  or  $\mathbf{G}$  by the known values of  $u$  or  $q$ . It is interesting to point out that the unknowns are now a mixture of the potential and its normal derivative, rather than the potential only as in finite elements. This is a consequence of the boundary element method being a “mixed” formulation, and constitutes an important advantage over finite elements.

Equation (2.31) can now be solved and all the boundary values will then be known. Once this is done it is possible to calculate internal values of  $u$  or its derivatives. The values of  $u$  are calculated at any internal point  $i$  using formula (2.11) which can be written in condensed form as

$$u_i = \int_{\Gamma} q u^* d\Gamma - \int_{\Gamma} u q^* d\Gamma \quad (2.32)$$

Notice that now the fundamental solution is considered to be acting on an internal point  $i$  and that all values of  $u$  and  $q$  are already known. The process is then one of direct integration. The same discretization is used for the boundary integrals, i.e.

$$u_i = \sum_{j=1}^N G_{ij} q_j - \sum_{j=1}^N \bar{H}_{ij} u_j \quad (2.33)$$

The coefficients  $G_{ij}$  and  $\bar{H}_{ij}$  have to be calculated anew for each different internal point.

The values of the internal fluxes in the two cartesian directions,  $q_x = \partial u / \partial x$  and  $q_y = \partial u / \partial y$ , are calculated by carrying out derivatives on (2.32), i.e.

$$(q_x)_i = \left( \frac{\partial u}{\partial x} \right)_i = \int_{\Gamma} q \frac{\partial u^*}{\partial x} d\Gamma - \int_{\Gamma} u \frac{\partial q^*}{\partial x} d\Gamma \quad (2.34)$$

$$(q_y)_i = \left( \frac{\partial u}{\partial y} \right)_i = \int_{\Gamma} q \frac{\partial u^*}{\partial y} d\Gamma - \int_{\Gamma} u \frac{\partial q^*}{\partial y} d\Gamma$$

Notice that the derivatives are carried out only on the fundamental solutions  $u^*$  and  $q^*$  as we are computing the variations of flux around the point  $i$  [4].

Computation of integrals for internal points in (2.32) and (2.34) is usually carried out numerically. The procedure is somewhat cumbersome, but in chapter 4 a simpler approach using the Dual Reciprocity Method [12-14] will be given.

### 2.2.4 Evaluation of Integrals

The coefficients  $G_{ij}$  and  $\bar{H}_{ij}$  in the previous expressions can be calculated using numerical integration formulae (such as Gauss quadrature) for the case  $i \neq j$ . In the case where  $i$  and  $j$  are on the same element (*i.e.*  $i = j$ ) the singularity of the fundamental solution requires a more accurate integration scheme. For these integrals it is recommended to use higher-order integration rules or a special formula (such as logarithmic and other transformations which will be discussed later on).

For the particular case of constant elements, the coefficients  $H_{ii}$  and  $G_{ii}$  may be computed analytically. The  $\bar{H}_{ii}$  terms, for instance, are identically zero, as the normal  $n$  and the distance  $r$  from the source point are always perpendicular to each other, *i.e.*

$$\bar{H}_{ii} = \int_{\Gamma_i} q^* d\Gamma = \int_{\Gamma_i} \frac{\partial u^*}{\partial r} \frac{\partial r}{\partial n} d\Gamma \equiv 0 \quad (2.35)$$

The  $G_{ii}$  integrals require special handling. For a two-dimensional element, for instance, they are of the form

$$G_{ii} = \int_{\Gamma_i} u^* d\Gamma = \frac{1}{2\pi} \int_{\Gamma_i} \ln\left(\frac{1}{r}\right) d\Gamma \quad (2.36)$$

In order to integrate the above expression one can perform a change of coordinates (figure 2.4) such that

$$r = \left| \frac{\ell}{2} \xi \right| \quad d\Gamma = dr = \frac{\ell}{2} d\xi \quad (2.37)$$

where  $\ell$  is the element length. Hence, taking into account symmetry, expression (2.36) can be written as

$$\begin{aligned} G_{ii} &= \frac{1}{2\pi} \int_{\text{Point 1}}^{\text{Point 2}} \ln\left(\frac{1}{r}\right) d\Gamma = \frac{1}{\pi} \int_{\text{node } i}^{\text{Point 2}} \ln\left(\frac{1}{r}\right) dr = \\ &= \frac{\ell}{2\pi} \int_0^1 \ln\left(\frac{2}{\ell\xi}\right) d\xi = \\ &= \frac{\ell}{2\pi} \left\{ \ln\left(\frac{2}{\ell}\right) + \int_0^1 \ln\left(\frac{1}{\xi}\right) d\xi \right\} \end{aligned} \quad (2.38)$$

The last integral is equal to 1 so that

$$G_{ii} = \frac{\ell}{2\pi} \left\{ \ln\left(\frac{2}{\ell}\right) + 1 \right\} \quad (2.39)$$

For more complex cases, such as curved elements, special weighted formulae should be used (see Appendix 2). In the two-dimensional codes described in this and later chapters a 4 points Gauss quadrature rule has been used to evaluate the non-singular integrals.

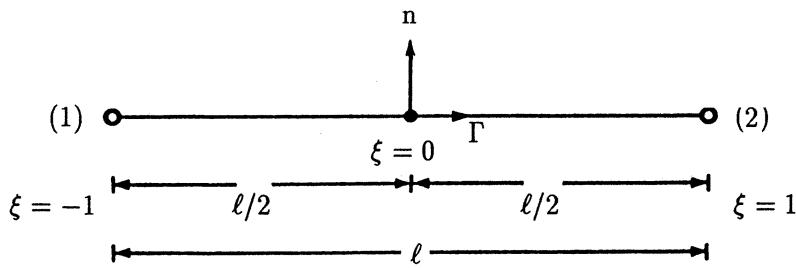
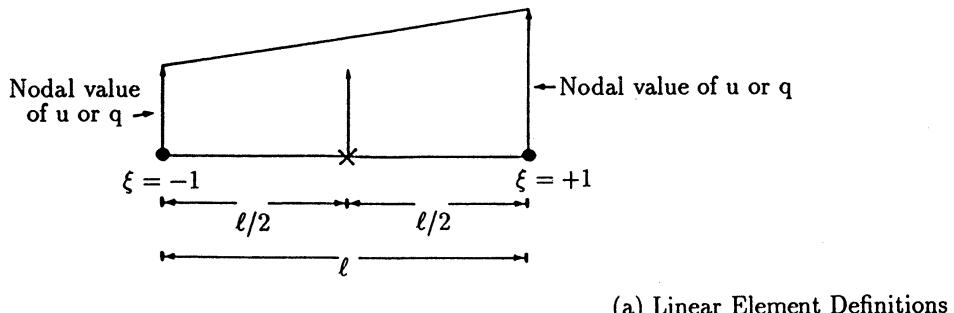
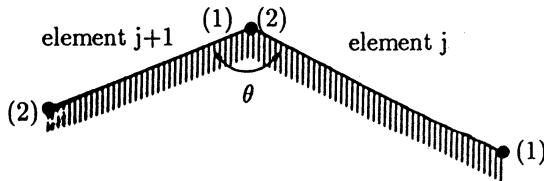


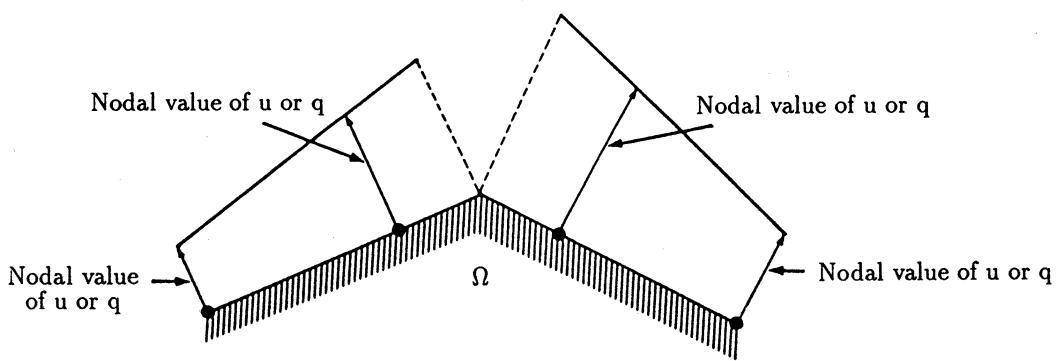
Figure 2.4: Element Coordinate System



### (a) Linear Element Definitions



### (b) Element Intersection



### (c) Discontinuous Elements

Figure 2.5: Linear Element: Basic Definitions and Corner Treatment

## 2.2.5 Linear Elements

Up to this section only the case of constant elements, *i.e.* those with the values of the variables assumed to be the same all over the element, has been discussed. Let us now consider a linear variation of  $u$  and  $q$  for which case the nodes are considered to be at the ends of the element as shown in figure 2.5.

The boundary integral equation (2.23) can now be generalized as

$$c_i u_i + \int_{\Gamma} u q^* d\Gamma = \int_{\Gamma} q u^* d\Gamma \quad (2.40)$$

Notice that the  $\frac{1}{2}$  coefficient of  $u_i$  has been replaced by a known value  $c_i$ . This is because  $c_i = \frac{1}{2}$  applies only for a node located on a smooth part of the boundary. The values of  $c_i$  for any boundary point can be shown to be

$$c_i = \frac{\theta}{2\pi} \quad (2.41)$$

where  $\theta$  is the internal angle at point  $i$  in radians. This result is obtained by defining a small spherical or circular region around the point and then taking its radius to zero (similar to what has been shown in section 2.2.2). Another possibility is to determine the value of  $c_i$  implicitly (see section 2.2.6), and in this case it is not necessary to calculate the angle.

After discretizing the boundary into a series of  $N$  elements, equation (2.40) can be written as in the previous section:

$$c_i u_i + \sum_{j=1}^N \int_{\Gamma_j} u q^* d\Gamma = \sum_{j=1}^N \int_{\Gamma_j} q u^* d\Gamma \quad (2.42)$$

The integrals in this equation are more difficult to evaluate than those for the constant element as  $u$  and  $q$  vary linearly over each element  $\Gamma_j$  and hence it is not possible to take them out of the integrals.

The values of  $u$  and  $q$  at any point on the element can be defined in terms of their nodal values and two linear interpolation functions  $\phi_1$  and  $\phi_2$ , which are given in terms of the homogeneous coordinate  $\xi$  as shown in figure 2.5(a), *i.e.*

$$u(\xi) = \phi_1 u_1 + \phi_2 u_2 = \begin{bmatrix} \phi_1 & \phi_2 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2.43)$$

$$q(\xi) = \phi_1 q_1 + \phi_2 q_2 = \begin{bmatrix} \phi_1 & \phi_2 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix}$$

The dimensionless coordinate  $\xi$  varies from  $-1$  to  $+1$  and the two interpolation functions are

$$\begin{aligned} \phi_1 &= \frac{1}{2}(1 - \xi) \\ \phi_2 &= \frac{1}{2}(1 + \xi) \end{aligned} \quad (2.44)$$

Let us consider the integrals over an element  $j$ . Those on the left-hand side of (2.42) can be written as

$$\int_{\Gamma_j} u q^* d\Gamma = \int_{\Gamma_j} [\phi_1 \quad \phi_2] q^* d\Gamma \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = [\ h_{ij}^1 \quad h_{ij}^2 ] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (2.45)$$

where, for each element  $j$ , we have the two terms

$$h_{ij}^1 = \int_{\Gamma_j} \phi_1 q^* d\Gamma \quad (2.46)$$

and

$$h_{ij}^2 = \int_{\Gamma_j} \phi_2 q^* d\Gamma \quad (2.47)$$

Similarly, the integral on the right-hand side of (2.42) gives

$$\int_{\Gamma_j} q u^* d\Gamma = \int_{\Gamma_j} [\phi_1 \quad \phi_2] u^* d\Gamma \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = [\ g_{ij}^1 \quad g_{ij}^2 ] \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad (2.48)$$

where

$$g_{ij}^1 = \int_{\Gamma_j} \phi_1 u^* d\Gamma \quad (2.49)$$

and

$$g_{ij}^2 = \int_{\Gamma_j} \phi_2 u^* d\Gamma \quad (2.50)$$

## 2.2.6 Treatment of Corners

In general, a boundary element discretization will present a series of points of geometric discontinuity which require special attention as the conditions on both sides may not be the same.

When the boundary of the region is discretized into linear elements, node 2 of element  $j$  is the same point as node 1 of element  $j + 1$  (figure 2.5(b)). While corners with different values of the flux on both sides exist in many practical problems, discontinuous values of the potential are seldom prescribed. Since the potential is unique at any point of the boundary,  $u^2$  of element  $j$  and  $u^1$  of element  $j + 1$  have the same value. However, this argument cannot be applied as a general rule to the flux.

To take into account the possibility that the flux at node 2 of an element may be different from the flux at node 1 of the next element, the fluxes can be arranged in a  $2N$  array.

Substituting equations (2.45) and (2.48) for all elements  $j$  into (2.42) the following equation for node  $i$  is obtained:

$$c_i u_i + \begin{bmatrix} \bar{H}_{i1} & \bar{H}_{i2} & \cdots & \bar{H}_{iN} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_{2N} \end{bmatrix} \quad (2.51)$$

$$\begin{bmatrix} G_{i1} & G_{i2} & \cdots & G_{i2N} \end{bmatrix}$$

where  $\bar{H}_{ij}$  is equal to the  $h_{ij}^1$  term of element  $j$  plus the  $h_{i,j-1}^2$  term of element  $j - 1$ . Hence, formula (2.51) represents the assembled equation for node  $i$ . Note the simplicity of the approach. Equation (2.51) can be written as

$$c_i u_i + \sum_{j=1}^N \bar{H}_{ij} u_j = \sum_{j=1}^{2N} G_{ij} q_j \quad (2.52)$$

Similarly as was previously shown for constant elements (equation (2.29)), this formula can be written as

$$\sum_{j=1}^N H_{ij} u_j = \sum_{j=1}^{2N} G_{ij} q_j \quad (2.53)$$

and the whole set in matrix form becomes

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q} \quad (2.54)$$

where  $\mathbf{G}$  is now a rectangular matrix of size  $N \times 2N$ .

Several situations may occur at a boundary node. First, that the boundary is smooth at the node; in such a case, both fluxes “before” and “after” the node are the same unless they are prescribed as different, but in any case only one variable will be unknown, either the potential or the unique flux. Second, that the node is a corner point. In this case there are four possibilities depending on the boundary conditions:

1. Known values: fluxes “before” and “after” the corner.  
Unknown value: potential.
2. Known values: potential, and flux “before” the corner.  
Unknown value: flux “after” the corner.
3. Known values: potential, and flux “after” the corner.  
Unknown value: flux “before” the corner.
4. Known value: potential.  
Unknown values: flux “before” and “after” the corner.

There is only one unknown per node for the first three cases: the two known values are taken to the right-hand side and the usual system of  $N \times N$  equations obtained, *i.e.*

$$\mathbf{Ax} = \mathbf{y} \quad (2.55)$$

where  $\mathbf{x}$  is a vector of unknowns and  $\mathbf{A}$  is a square matrix, the columns of which contain either columns of the matrix  $\mathbf{H}$ , columns of the matrix  $\mathbf{G}$  after a change of sign or the sum of two consecutive columns of  $\mathbf{G}$  with a change of sign when the unknown is the unique value of the flux at the corresponding node. The known vector  $\mathbf{y}$  is computed from the product of the known boundary conditions and the corresponding coefficients of the matrices  $\mathbf{G}$  or  $\mathbf{H}$ .

When the number of unknowns at a corner node is two (case 4), one extra equation is needed for the node. The problem can also be solved using the idea of “discontinuous” elements [15]. In this case, the second node of element  $j$  and the first node of element  $j + 1$  are shifted inside the two linear elements which meet at the corner and remain as two distinct nodes instead of joining into one at the corner (see figure 2.5(c)). Thus, one equation can be written for each node. The potential and the flux are represented by linear functions along the whole of the element in terms of their nodal values, but they are in principle discontinuous at the corner.

### Numerical Determination of Coefficients $c$

If the boundary is not smooth at point  $i$  the value  $c_i = \frac{1}{2}$  (equation (2.21)) is no longer valid and formula (2.41) needs to be applied. Another possibility is to calculate the diagonal terms of the  $\mathbf{H}$  matrix in equation (2.54) by using the fact that when a uniform potential is applied over a bounded region, all derivatives (including the  $q$  values) must be zero. Hence, equation (2.54) becomes

$$\mathbf{Hu} = 0 \quad (2.56)$$

where  $\mathbf{u}$  is a vector of constant values. Thus, the sum of all elements of any row of  $\mathbf{H}$  ought to be zero, and the values of the diagonal coefficients can be easily calculated once the off-diagonal coefficients are all known, *i.e.*

$$H_{ii} = - \sum_{\substack{j=1 \\ (\text{for } j \neq i)}}^N H_{ij} \quad (2.57)$$

In this way the values of  $c_i$  need not be calculated explicitly.

The above considerations are valid strictly for closed domains. When dealing with infinite regions equation (2.57) must be modified. If a unit potential is prescribed over an unbounded region the integral

$$\int_{\Gamma_\infty} q^* d\Gamma$$

## 28 The Dual Reciprocity

over a fictitious external boundary  $\Gamma_\infty$  at infinity will not be zero, and will give the result [4]

$$\int_{\Gamma_\infty} q^* d\Gamma = -1$$

The diagonal terms in this case are given by the following formula,

$$H_{ii} = 1 - \sum_{\substack{j=1 \\ (\text{for } j \neq i)}}^N H_{ij} \quad (2.58)$$

### 2.2.7 Quadratic and Higher-Order Elements

It is usually more convenient for arbitrary geometries to implement some type of curvilinear element. The simplest of these is the three-node quadratic element which requires working with transformations.

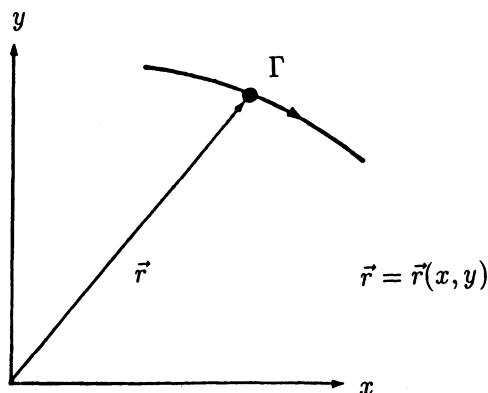


Figure 2.6: Curved Boundary

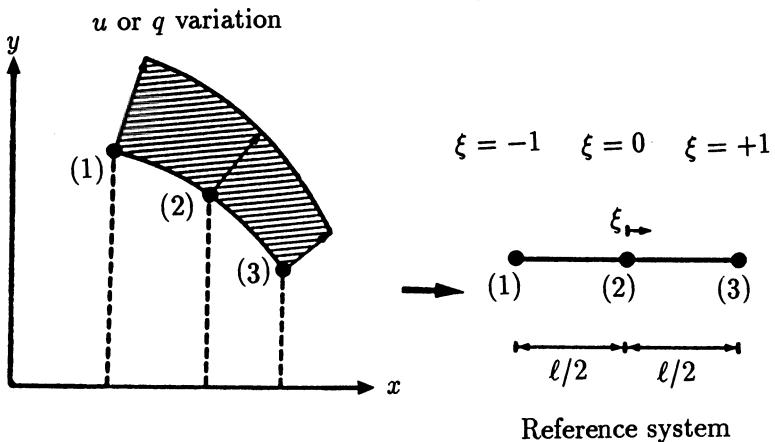


Figure 2.7: Quadratic Element

Consider the curved boundary shown in figure 2.6 where  $\Gamma$  is defined along the boundary and the position vector  $\vec{r}$  is a function of the cartesian system  $(x, y)$ . The variables  $u$  or  $q$  can be written in terms of interpolation functions which depend on the homogeneous coordinate  $\xi$ , i.e.

$$u(\xi) = \phi_1 u_1 + \phi_2 u_2 + \phi_3 u_3 = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (2.59)$$

$$q(\xi) = \phi_1 q_1 + \phi_2 q_2 + \phi_3 q_3 = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (2.60)$$

where the interpolation functions are

$$\begin{aligned} \phi_1 &= \frac{1}{2}\xi(\xi - 1) \\ \phi_2 &= (1 - \xi)(1 + \xi) \\ \phi_3 &= \frac{1}{2}\xi(\xi + 1) \end{aligned} \quad (2.61)$$

These functions are quadratic in  $\xi$ ; expressions (2.59) and (2.60) give the nodal values of the variables  $u$  or  $q$  when specialized for the nodes, i.e. with reference to figure 2.7

Node	$\xi$	$\phi_1$	$\phi_2$	$\phi_3$
1	-1	1	0	0
2	0	0	1	0
3	+1	0	0	1

The integrals along the quadratic elements are similar to those for the linear elements, but there are now three unknown nodal values. Consider, for instance, the integral for the coefficients of matrix  $\mathbf{H}$ , *i.e.*

$$\int_{\Gamma_j} u q^* d\Gamma = \int_{\Gamma_j} \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 \end{bmatrix} q^* d\Gamma \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} h_{ij}^1 & h_{ij}^2 & h_{ij}^3 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (2.62)$$

where

$$\begin{aligned} h_{ij}^1 &= \int_{\Gamma_j} \phi_1 q^* d\Gamma \\ h_{ij}^2 &= \int_{\Gamma_j} \phi_2 q^* d\Gamma \\ h_{ij}^3 &= \int_{\Gamma_j} \phi_3 q^* d\Gamma \end{aligned} \quad (2.63)$$

The evaluation of these terms requires the use of a Jacobian as the functions  $\phi_i$  are expressed in terms of  $\xi$ , but the integrals are functions of  $\Gamma$ . For a curve such as that given in figure 2.6, the transformation is simple,

$$d\Gamma = \sqrt{\left(\frac{dx}{d\xi}\right)^2 + \left(\frac{dy}{d\xi}\right)^2} d\xi = |G| d\xi \quad (2.64)$$

where  $|G|$  is the Jacobian. Hence, one can write

$$h_{ij}^k = \int_{\Gamma_j} \phi_k q^* d\Gamma = \int_{-1}^{+1} \phi_k q^* |G| d\xi \quad (2.65)$$

Formulae such as (2.65) are generally too difficult to integrate analytically and numerical integration must be used in all cases, including those elements with a singularity. For further details, see Appendices 1 and 2.

Notice that in order to calculate the values of the Jacobian  $|G|$  in (2.64) one needs to know the variation of the coordinates  $x$  and  $y$  in terms of  $\xi$ . This can be done by defining the geometrical shape of the element in the same way as the variables  $u$  and  $q$  are defined, *i.e.* using quadratic interpolation,

$$\begin{aligned} x &= \phi_1 x_1 + \phi_2 x_2 + \phi_3 x_3 \\ y &= \phi_1 y_1 + \phi_2 y_2 + \phi_3 y_3 \end{aligned} \quad (2.66)$$

where the subscript indicates the node number. This is a similar concept to that of isoparametric elements commonly used in finite element analysis.

## Cubic Elements

Elements of higher order than quadratic are seldom used in practice, but they may be interesting in some particular applications.

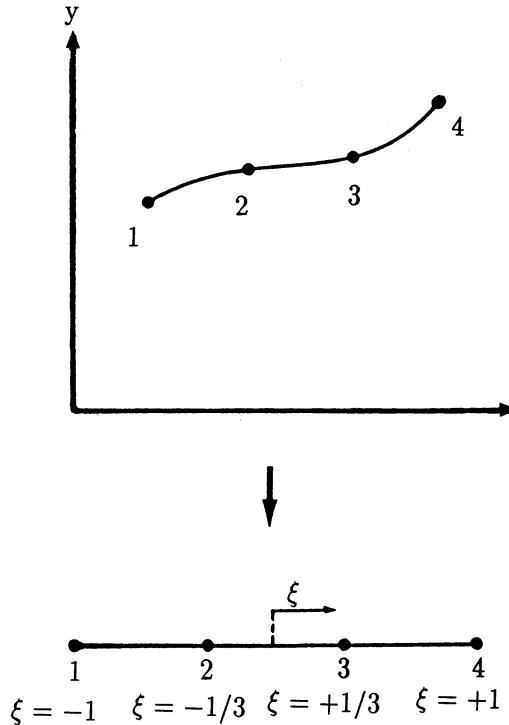


Figure 2.8: Cubic Elements with Four Nodes

Because of this, the case of elements with a cubic variation of geometry and of variables  $u$  or  $q$  will be briefly described. In this case, the functions are defined by taking four nodes over each element (figure 2.8)

$$\begin{aligned} u &= \phi_1 u_1 + \phi_2 u_2 + \phi_3 u_3 + \phi_4 u_4 \\ q &= \phi_1 q_1 + \phi_2 q_2 + \phi_3 q_3 + \phi_4 q_4 \end{aligned} \quad (2.67)$$

and similarly

$$\begin{aligned} x &= \phi_1 x_1 + \phi_2 x_2 + \phi_3 x_3 + \phi_4 x_4 \\ y &= \phi_1 y_1 + \phi_2 y_2 + \phi_3 y_3 + \phi_4 y_4 \end{aligned} \quad (2.68)$$

where the interpolation functions are

$$\begin{aligned}\phi_1 &= \frac{1}{16} (1 - \xi) [-10 + 9 (\xi^2 + 1)] \\ \phi_2 &= \frac{9}{16} (1 - \xi^2) (1 - 3\xi) \\ \phi_3 &= \frac{9}{16} (1 - \xi^2) (1 + 3\xi) \\ \phi_4 &= \frac{1}{16} (1 + \xi) [-10 + 9 (\xi^2 + 1)]\end{aligned}\tag{2.69}$$

The interpolation functions take the following values at the nodes:

Node	$\xi$	$\phi_1$	$\phi_2$	$\phi_3$	$\phi_4$
1	-1	1	0	0	0
2	$-\frac{1}{3}$	0	1	0	0
3	$+\frac{1}{3}$	0	0	1	0
4	+1	0	0	0	1

Another possibility with cubic elements is to define the variation of  $u$  and  $q$  in terms of the function and its derivative along the element at the two extreme points as shown in figure 2.9.

The corresponding function for  $u$  is then given by

$$u = \phi_1 u_1 + \phi_2 \left( \frac{\partial u}{\partial \Gamma} \right)_1 + \phi_3 u_2 + \phi_4 \left( \frac{\partial u}{\partial \Gamma} \right)_2\tag{2.70}$$

with

$$\begin{aligned}\phi_1 &= \frac{1}{4}(\xi - 1)^2 (2 + \xi) \\ \phi_2 &= \frac{1}{8}\ell(\xi - 1)^2 (\xi + 1) \\ \phi_3 &= \frac{1}{4}(\xi + 1)^2 (2 - \xi) \\ \phi_4 &= \frac{1}{8}\ell(\xi + 1)^2 (\xi - 1)\end{aligned}\tag{2.71}$$

where  $\ell$  is the element length. In the case of a curved element  $\ell$  can be calculated integrating the Jacobian  $|G|$ , given by equation (2.64), over the element. The same functions  $\phi$  are used for  $q$ ,  $x$  and  $y$ .

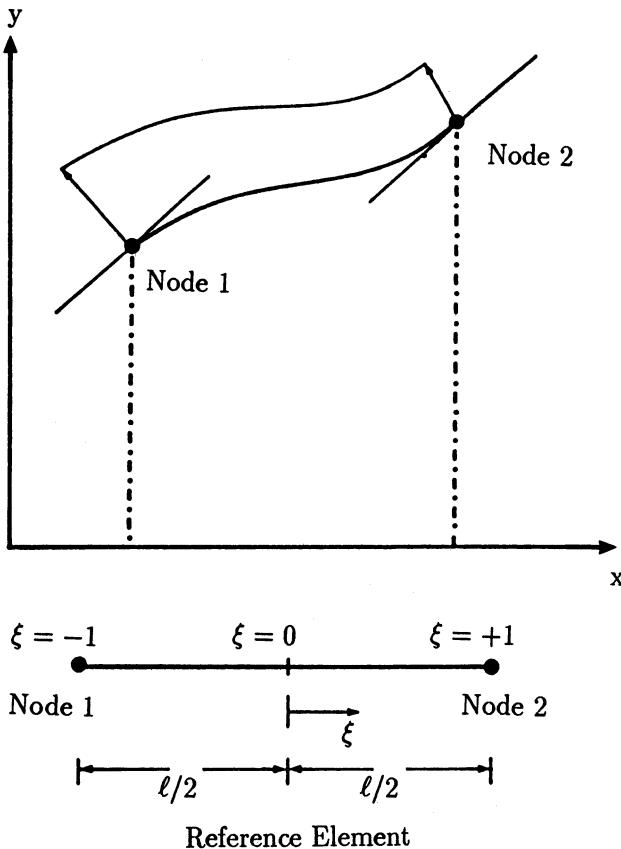


Figure 2.9: Cubic Elements with Only Two Nodes

This type of cubic element could be used in cases where one wishes to have a correct definition of the derivative along  $\Gamma$ , for instance to calculate the fluxes in the tangential direction, or if a reduction in the number of nodes along the element is required. In some cases it may still be better to continue to describe the geometry with four nodes unless this is defined by an analytic function.

### Numerical Results for a Laplace Problem

The Laplace equation,

$$\nabla^2 u = 0 \quad (2.72)$$

is solved in this example for the geometry shown in figure 2.10.

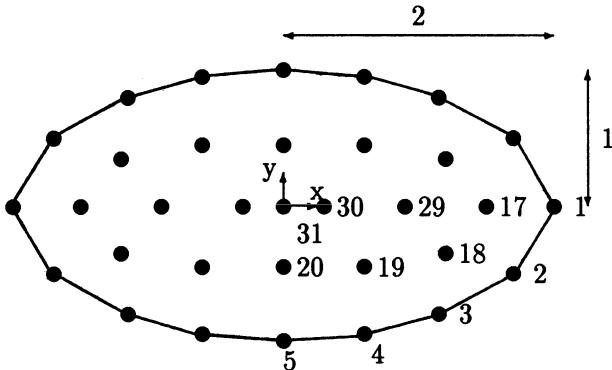


Figure 2.10: Laplace Equation Problem

Here an ellipse of semi-major axis of length 2 and semi-minor axis of length 1 is discretized using 16 linear boundary elements as described in section 2.2.5. 17 internal nodes where the solution is required are also defined.

Homogeneous boundary conditions cannot be prescribed for the Laplace equation as the result will be  $u = q = 0$  at all nodes. As a consequence, a non-homogeneous boundary condition has to be imposed, for example

$$u = \bar{u} = x + y \quad (2.73)$$

The reader may easily verify that (2.73) is a particular solution to the Laplace equation (2.72) which may be used to verify the results which are given in table 2.1.

Node	X	Y	BEM	Exact
17	1.5	0.0	1.507	1.500
18	1.2	-0.35	0.857	0.850
19	0.6	-0.45	0.154	0.150
20	0.0	-0.45	-0.451	-0.450
29	0.9	0.0	0.913	0.900
30	0.3	0.0	0.304	0.300
31	0.0	0.0	0.0	0.0

Table 2.1: BEM Results for the Laplace Equation

To obtain these results, Program 1 of section 2.4 may be used with CONST=0.

The Laplace equation will be used in later chapters as starting point for some iterative procedures, such as for solution of Burger's equation in section 4.4.1, where it enables the first iteration to be carried out producing results which are then used in the second iteration to approximate the domain term.

## 2.3 Formulation for the Poisson Equation

### 2.3.1 Basic Relationships

Domain integrals in boundary elements may arise due to a variety of effects such as body forces, initial states, non-linear terms and others. In what follows, the use of the non-homogeneous Poisson differential equation will be studied. The resulting formulation will later on be extended to cases for which the right-hand side term is, amongst others, a function of space, the potential itself or includes time-dependent effects.

Consider first the case of Poisson's equation, *i.e.*

$$\nabla^2 u = b \quad \text{in } \Omega \quad (2.74)$$

where  $b$  is at present assumed to be a known function. One can start by applying weighted residual formulations in order to deduce the basic integral equations, in a similar way to that done for the Laplace equation in section 2.2, *i.e.*

$$\int_{\Omega} (\nabla^2 u - b) u^* d\Omega = \int_{\Gamma_2} (q - \bar{q}) u^* d\Gamma - \int_{\Gamma_1} (u - \bar{u}) q^* d\Gamma \quad (2.75)$$

which integrated by parts twice produces

$$\begin{aligned} \int_{\Omega} (\nabla^2 u^*) u d\Omega - \int_{\Omega} b u^* d\Omega &= - \int_{\Gamma_2} \bar{q} u^* d\Gamma - \int_{\Gamma_1} q u^* d\Gamma + \\ &+ \int_{\Gamma_2} u q^* d\Gamma + \int_{\Gamma_1} \bar{u} q^* d\Gamma \end{aligned} \quad (2.76)$$

After substituting the fundamental solution  $u^*$  of the Laplace equation into (2.76) and grouping all boundary terms together one obtains

$$c_i u_i + \int_{\Gamma} u q^* d\Gamma + \int_{\Omega} b u^* d\Omega = \int_{\Gamma} q u^* d\Gamma \quad (2.77)$$

Notice that although the  $b$  function is known and consequently the integral in  $\Omega$  do not introduce any new unknowns, the problem has changed in character as we need now to carry out a domain integral as well as the boundary integrals. The constant  $c_i$ , as explained before, depends only on the boundary geometry at the point  $i$  under consideration.

### 2.3.2 Cell Integration Approach

The simplest way of computing the domain term in equation (2.77) is by subdividing the region into a series of internal cells, on each of which a numerical integration scheme such as Gauss quadrature can be applied (figure 2.11).

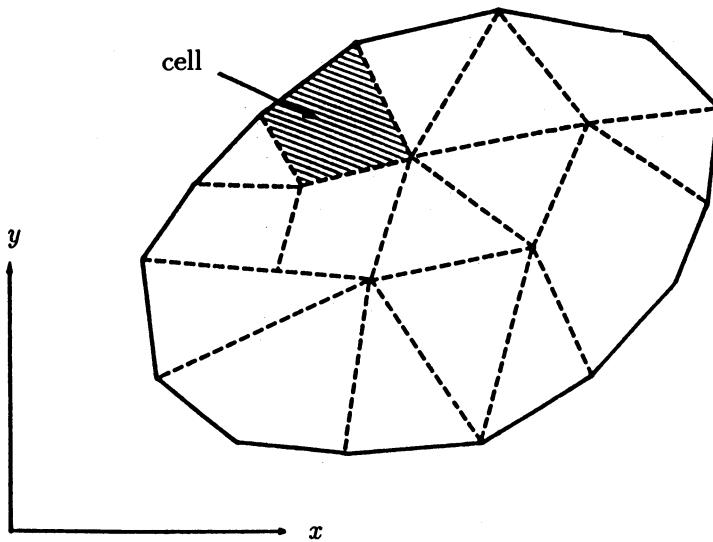


Figure 2.11: Boundary Elements and Internal Cells

In this case, for each boundary point  $i$ , the domain integral in (2.77) can be written as

$$d_i = \int_{\Omega} bu^* d\Omega = \sum_{e=1}^M \left[ \sum_{k=1}^R w_k (bu^*)_k \right] \Omega_e \quad (2.78)$$

where the integral has been approximated by a summation over different cells ( $e$  varies from 1 to  $M$ , where  $M$  is the total number of cells describing the domain  $\Omega$ ),  $w_k$  are the Gauss integration weights, the function  $(bu^*)$  needs to be evaluated at integration points  $k$  on each cell ( $k$  varies from 1 to  $R$ , where  $R$  is the total number of integration points on each cell), and  $\Omega_e$  is the area of cell  $e$ . The term  $d_i$  is the result of the numerical integration and is different for each position  $i$  of the boundary nodes.

Equation (2.77) can now be written as

$$c_i u_i + \sum_{j=1}^N \bar{H}_{ij} u_j + d_i = \sum_{j=1}^N G_{ij} q_j \quad (2.79)$$

or, in matrix form,

$$\mathbf{H}\mathbf{u} + \mathbf{d} = \mathbf{G}\mathbf{q} \quad (2.80)$$

Notice that the domain integrals need to be computed as well when calculating any value of potential or fluxes at internal points. Hence,

$$u_i = \sum_{j=1}^N G_{ij} q_j - \sum_{j=1}^N \bar{H}_{ij} u_j - d_i \quad (2.81)$$

where  $i$  is now an internal point.

Concentrated sources are very simple to handle in boundary elements. They can be seen as a special case for which the function  $b$  at the internal point  $\ell$  becomes

$$b = Q_\ell \Delta_\ell \quad (2.82)$$

where  $Q_\ell$  is the magnitude of the source and  $\Delta_\ell$  is a Dirac delta function. Assuming that a number of these functions exists, they can be written in a summation term as follows,

$$c_i u_i + \int_{\Gamma} u q^* d\Gamma + \int_{\Omega} b u^* d\Omega + \sum_{\ell=1}^P Q_\ell u_\ell^* = \int_{\Gamma} q u^* d\Gamma \quad (2.83)$$

where  $u_\ell^*$  is the value of the fundamental solution at the point  $\ell$  and  $P$  is the number of concentrated sources within the domain.

### Results for the Poisson Equation using Internal Cells

The equation  $\nabla^2 u = -2$  was solved for the geometry shown in figure 2.12 with the homogeneous boundary condition  $\bar{u} = 0$ . This problem is discussed in detail in section 3.4.5 where the DRM results are presented for the same geometry.

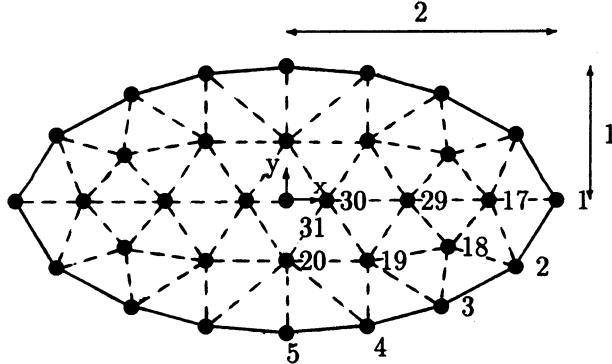


Figure 2.12: Elliptical Section: Boundary Elements and Internal Cells

The same boundary element discretization was used as in the previous problem, however the domain was discretized using 48 internal cells. Results are presented in table 2.2.

These results may be obtained with Program 1 using CONST=-2.

### 2.3.3 The Monte Carlo Method

Another way of integrating the domain terms in boundary elements has been proposed by Gipson [16] and consists of using a system of random integration points rather than applying a regular integration grid as is done in the cell integration method.

Node	X	Y	Cell	Exact
17	1.5	0.0	0.331	0.350
18	1.2	-0.35	0.401	0.414
19	0.6	-0.45	0.557	0.566
20	0.0	-0.45	0.629	0.638
29	0.9	0.0	0.626	0.638
30	0.3	0.0	0.772	0.782
31	0.0	0.0	0.791	0.800

Table 2.2: Cell Results for Poisson Problem

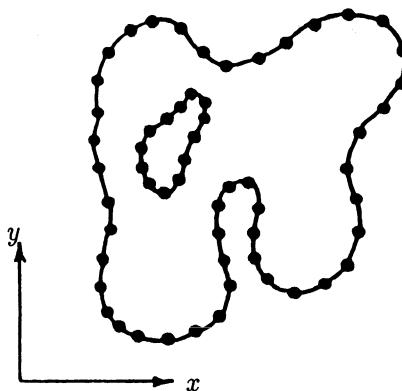


Figure 2.13: Region Geometry

The Monte Carlo technique can be used to generate the random points. Although the concept can be applied to three- as well as two-dimensional problems, only the latter will be considered here for simplicity. Consider a region  $\Omega$  (figure 2.13) of a very general geometry over which one wants to integrate the following term

$$d_i = \int_{\Omega} bu^* d\Omega \quad (2.84)$$

The only requirement for the Monte Carlo technique is that the function  $b$  is integrable. In addition, it will be assumed here that the region is bounded and the limits of integration can be defined within a rectangular area as shown in figure 2.14.

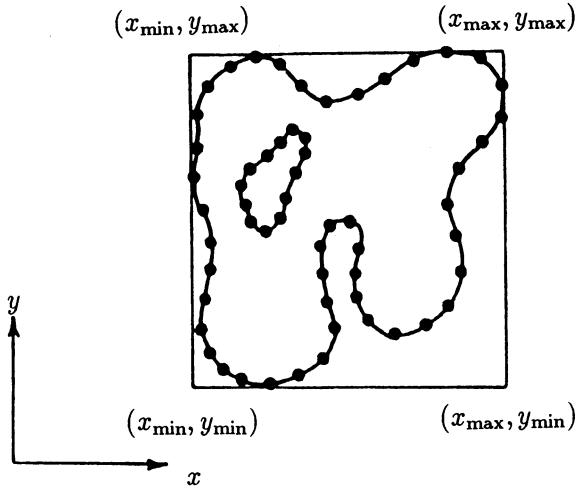


Figure 2.14: Definition of the Minimal Rectangle

Next, a series of uniformly distributed random points of coordinates  $(x_k, y_k)$  is chosen in the rectangle, and each point is tested to check if it belongs to the domain  $\Omega$ , otherwise the point is discarded.

One can consider  $(x_k, y_k)$  as an integration point in which the value of the function  $bu^*$  is computed, *i.e.*

$$u^*b(x_k, y_k) = (u^*b)_k \quad (2.85)$$

and add this to the running total of  $\sum_{k=1}^M (u^*b)_k$ . A similar calculation is done with the square of this function, *i.e.*

$$\sum_{k=1}^M [(u^*b)_k]^2 \quad (2.86)$$

The operation can be repeated until the desired number  $M$  of points has been obtained. The value of the integral (2.84) can then be found, *i.e.*

$$d_i = \frac{\Omega}{M} \sum_{k=1}^M (u^*b)_k \quad (2.87)$$

with the variance given by the following relationship

$$v_i = \frac{\Omega}{M} \sum_{k=1}^M (u^*b)_k^2 - d_i^2 \quad (2.88)$$

The variance can be used to estimate the error in the calculation.

The algorithm has been implemented by Gipson [16] for the solution of a wide variety of Poisson-type problems. The results, although encouraging, tend to be

expensive in computer time as a large number of points is usually required to properly compute the domain term. The idea is nevertheless of interest because it is simple to apply. Improvements to the method could be attempted by giving the random points different weights corresponding to their position relative to other points.

### Results Using the Monte Carlo Method

This example of the use of the Monte Carlo method is taken from Gipson [16]. The problem is of heat generation on an isotropic square plate of side 12. Using the problem symmetry, only one quarter of the plate is modelled. The boundary element discretization used is shown in figure 2.15.

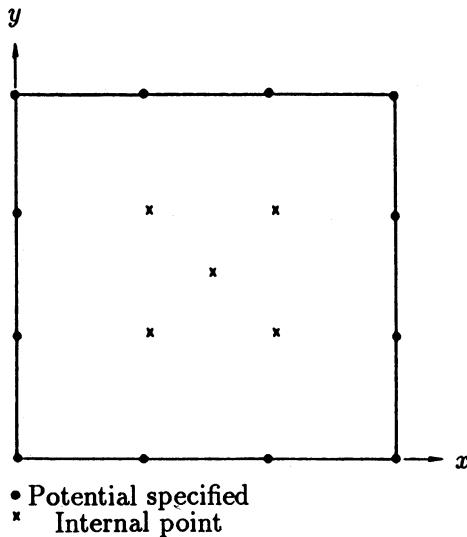


Figure 2.15: One Quarter of Square Plate

Given the use of symmetry, the boundary conditions are  $u = 0$  on the edges  $x = 6$  and  $y = 6$  and  $q = 0$  on the edges  $x = 0$  and  $y = 0$ . 12 linear elements were employed, using the concept of discontinuous elements, discussed in section 2.2.6, at corner points.

The governing equation for this case is

$$\frac{\partial}{\partial x} \left( k_x \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( k_y \frac{\partial u}{\partial y} \right) = -Q \quad (2.89)$$

where  $Q$  is the rate of internal heat generation,  $k_x$  and  $k_y$  are thermal conductivities and  $u$  is the temperature. If  $Q$ ,  $k_x$  and  $k_y$  have unit values then equation (2.89) reduces to the Poisson equation

$$\nabla^2 u = -1 \quad (2.90)$$

The solution was computed using 500, 1000 and 3000 random integration points, with results given in table 2.3. The exact solution is taken from reference [17].

X	Y	$u_{500}$	$u_{1000}$	$u_{3000}$	Exact
2.0	2.0	8.985	8.543	8.537	8.690
4.0	2.0	5.802	5.645	5.736	5.748
3.0	3.0	6.498	6.362	6.477	6.522
2.0	4.0	5.718	5.634	5.633	5.748
4.0	4.0	3.961	3.987	3.981	3.928
2.0	0.0	9.761	9.607	9.718	9.588
4.0	0.0	6.628	6.161	6.234	6.286

Table 2.3: Results of Monte Carlo Method for Square Plate

### 2.3.4 The Use of Particular Solutions

A simple way of solving equation (2.77) without having to compute any domain integrals is by changing the variables in such a manner that these integrals disappear [18]. This can be attempted by splitting the function  $u$  into a particular solution and the solution of the associated homogeneous equation.

To illustrate this procedure, consider the Poisson equation (2.74) with boundary conditions as given by (2.2). Assume now that the potential function  $u$  can be written as

$$u = \tilde{u} + \hat{u} \quad (2.91)$$

where  $\tilde{u}$  is the solution of the homogeneous equation and  $\hat{u}$  is a particular solution of the Poisson equation such that

$$\nabla^2 \hat{u} = b \quad (2.92)$$

One can now write the domain integral in (2.77) as

$$\int_{\Omega} bu^* d\Omega = \int_{\Omega} (\nabla^2 \hat{u}) u^* d\Omega \quad (2.93)$$

Integrating by parts this equation and taking into consideration the special character of the fundamental solution - see equation (2.13) - one finds the following expressions

$$\begin{aligned} \int_{\Omega} bu^* d\Omega &= \int_{\Omega} (\nabla^2 \hat{u}) u^* d\Omega = \\ &= \int_{\Omega} \hat{u} (\nabla^2 u^*) d\Omega + \int_{\Gamma} u^* \hat{q} d\Gamma - \int_{\Gamma} q^* \hat{u} d\Gamma = \\ &= -c_i \hat{u}_i + \int_{\Gamma} u^* \hat{q} d\Gamma - \int_{\Gamma} q^* \hat{u} d\Gamma \end{aligned} \quad (2.94)$$

where  $\hat{q} = \partial \hat{u} / \partial n$ .

Substituting (2.94) into (2.77) one finds the following expression,

$$\begin{aligned} & c_i u_i + \int_{\Gamma} u q^* d\Gamma - \int_{\Gamma} q u^* d\Gamma \\ &= c_i \hat{u}_i + \int_{\Gamma} q^* \hat{u} d\Gamma - \int_{\Gamma} u^* \hat{q} d\Gamma \end{aligned} \quad (2.95)$$

Notice that now all integrals are computed only along the boundary. Equation (2.95) can be written in a more compact form as a function of the new variable  $\tilde{u}$  as follows,

$$c_i \tilde{u}_i + \int_{\Gamma} q^* \tilde{u} d\Gamma = \int_{\Gamma} u^* \tilde{q} d\Gamma \quad (2.96)$$

where  $\tilde{u} = u - \hat{u}$  defines the new variable as given in equation (2.91).

Formula (2.95) can be written after the boundary discretization as

$$\begin{aligned} & c_i u_i + \sum_{j=1}^N \bar{H}_{ij} u_j - \sum_{j=1}^N G_{ij} q_j = \\ &= c_i \hat{u}_i + \sum_{j=1}^N \bar{H}_{ij} \hat{u}_j - \sum_{j=1}^N G_{ij} \hat{q}_j \end{aligned} \quad (2.97)$$

Applying the above to all boundary points produce the following system,

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{H}\hat{\mathbf{u}} - \mathbf{G}\hat{\mathbf{q}} \quad (2.98)$$

or simply

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{d} \quad (2.99)$$

where the vector  $\mathbf{d}$  is given by

$$\mathbf{d} = \mathbf{H}\hat{\mathbf{u}} - \mathbf{G}\hat{\mathbf{q}} \quad (2.100)$$

The main disadvantage of this approach is the need to describe the behaviour of the function  $b$  in an analytical form which in some cases may be difficult or impossible to do. In chapters 3 and 4 the technique will be generalized to account for arbitrary or unknown types of  $b$  functions by using a summation of localized particular solutions with coefficients to be determined. This idea gave origin to the Dual Reciprocity Method first proposed by Nardini and Brebbia in 1982 [12].

## Results Using Particular Solutions

Here the Poisson equation  $\nabla^2 u = -2$  is solved using the solution procedure described above, i.e.

$$u = \tilde{u} + \hat{u} \quad (2.101)$$

which means that the complete solution  $u$  will be expressed as the sum of the solution to the homogeneous Laplace equation,  $\tilde{u}$ , plus a particular solution to the Poisson equation,  $\hat{u}$ .

The reader can easily verify that

$$\hat{u} = -\frac{1}{2} (x^2 + y^2) \quad (2.102)$$

is such a solution. Results may now be obtained for the Poisson equation solving the Laplace equation, using Program 1 with CONST=0. The boundary conditions are defined by (2.102) with a change of sign, in order that when the sum (2.101) is carried out, the net result will be the imposition of a homogeneous boundary condition on  $\Gamma$ .

The problem geometry is the same as that used in section 2.3.2, figure 2.12, but without the internal cells, with which the results of this solution procedure may be compared.

The solution procedure may easily be understood by examining table 2.4.

Node	X	Y	$\tilde{u}$	$\hat{u}$	$u = \tilde{u} + \hat{u}$	Cell	Exact
17	1.5	0.0	1.473	-1.125	0.348	0.331	0.350
18	1.2	-0.35	1.200	-0.781	0.419	0.401	0.414
19	0.6	-0.45	0.855	-0.281	0.574	0.557	0.566
20	0.0	-0.45	0.747	-0.101	0.646	0.629	0.638
29	0.9	0.0	1.049	-0.405	0.644	0.626	0.638
30	0.3	0.0	0.835	-0.045	0.790	0.772	0.782
31	0.0	0.0	0.808	0.000	0.808	0.791	0.800

Table 2.4: Results for Poisson Problem using Particular Solutions

The column  $\tilde{u}$  in table 2.4 is the solution of the Laplace equation with essential boundary conditions defined by equation (2.102) with a change of sign. Column  $\hat{u}$  is the result of evaluating (2.102) for the coordinates of the nodes. Column  $u = \tilde{u} + \hat{u}$  is the sum of the two, thus the problem solution. Note that here a problem with a domain integral has been solved without the use of internal cells.

The Dual Reciprocity Method can be interpreted as a superposition of localized particular solutions. An instance in which the DRM is equivalent to the particular solution method is discussed in section 3.2.3.

### 2.3.5 The Galerkin Vector Approach

Another way of dealing with volume sources is to transform the resulting domain integral into equivalent boundary integrals. In principle this is possible when the function  $b$  is harmonic (although in the next section it will be shown how this process can be generalized to deal with other functions). Harmonic means that  $b$  obeys the Laplace equation

$$\nabla^2 b = 0 \quad (2.103)$$

To effect the transformation - which is basically accomplished by integrating by parts - one makes use of a new function  $w^*$  which is such that

$$\nabla^2 w^* = u^* \quad (2.104)$$

where  $u^*$  is the fundamental solution defined in equation (2.9).

Next, Green's identity is written in the form

$$\int_{\Omega} (b \nabla^2 w^* - w^* \nabla^2 b) d\Omega = \int_{\Gamma} \left( b \frac{\partial w^*}{\partial n} - w^* \frac{\partial b}{\partial n} \right) d\Gamma \quad (2.105)$$

If  $\nabla^2 b = 0$  and (2.104) is taken into account, equation (2.105) reduces to

$$\int_{\Omega} b u^* d\Omega = \int_{\Gamma} b \frac{\partial w^*}{\partial n} d\Gamma - \int_{\Gamma} w^* \frac{\partial b}{\partial n} d\Gamma \quad (2.106)$$

Hence, the domain integral has been effectively reduced to two different boundary integrals.

The function  $w^*$  required in this case is simply the fundamental solution of the bi-harmonic equation, i.e.

$$\nabla^2 u^* = \nabla^2 (\nabla^2 w^*) = \nabla^4 w^* = -\Delta u^* \quad (2.107)$$

which for two dimensions is well known,

$$w^* = \frac{r^2}{8\pi} \left[ \ln \left( \frac{1}{r} \right) + 1 \right] \quad (2.108)$$

and for three dimensions is given by

$$w^* = \frac{1}{8\pi} r \quad (2.109)$$

Notice that for two dimensions  $w^*$  satisfies (2.104) as follows,

$$\nabla^2 w^* = \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial w^*}{\partial r} \right) = \frac{1}{2\pi} \ln \left( \frac{1}{r} \right) = u^* \quad (2.110)$$

and for three dimensions one finds that

$$\nabla^2 w^* = \frac{1}{r^2} \frac{\partial}{\partial r} \left( r^2 \frac{\partial w^*}{\partial r} \right) = \frac{1}{4\pi r} = u^* \quad (2.111)$$

### 2.3.6 The Multiple Reciprocity Method

The Multiple Reciprocity Method (MRM) is a new technique of transforming domain integrals to the boundary which can be seen as a generalization of the Galerkin Vector approach previously described. It employs a set of higher-order fundamental solutions which permits the application of Green's identity to each term of the sequence in succession. As a result the method can lead in the limit to the exact boundary-only formulation of the problem.

The MRM was introduced by Nowak [9] and extended to a series of applications by Nowak and Brebbia [10,11], including transient problems and the Helmholtz equation. The name MRM was proposed by Nowak and Brebbia in reference [11].

Consider again the case of Poisson's equation but now with the subscript 0 on the right-hand side of the equation to differentiate from similar functions which will be generated during the solution, *i.e.*

$$\nabla^2 u = b_0 \quad \text{in } \Omega \quad (2.112)$$

where  $u$  and  $b_0$  are the usual potential and source functions, respectively.

The fundamental solution of Laplace's equation as defined by (2.9) will also have a subscript 0 for the reasons explained above.

In applying reciprocity principles one obtains the same expression (2.77) as before, which is written below for completeness

$$c_i u_i + \int_{\Gamma} u q_0^* d\Gamma + \int_{\Omega} b_0 u_0^* d\Omega = \int_{\Gamma} q u_0^* d\Gamma \quad (2.113)$$

The domain integral in (2.113) can be transformed into a series of equivalent boundary integrals. In order to do this, one can start as in the previous section by introducing a function  $u_1^*$  related to the fundamental solution  $u_0^*$  by formula (2.104), *i.e.*

$$u_0^* = \nabla^2 u_1^* \quad (2.114)$$

Thus, the domain integral in (2.113) can be expanded as before, equation (2.105), but without assuming that  $\nabla^2 b_0 \equiv 0$ ,

$$\int_{\Omega} b_0 u_0^* d\Omega = \int_{\Omega} b_0 \nabla^2 u_1^* d\Omega = \int_{\Gamma} b_0 q_1^* d\Gamma - \int_{\Gamma} u_1^* \frac{\partial b_0}{\partial n} d\Gamma + \int_{\Omega} u_1^* \nabla^2 b_0 d\Omega \quad (2.115)$$

As the source function  $b_0$  is assumed to be a known function of space the value of  $\nabla^2 b_0$  can be obtained analytically; hence, a new function  $b_1$  can be defined such that

$$b_1 = \nabla^2 b_0 \quad (2.116)$$

The domain integral on the right-hand side of (2.115) can then be written in a similar form as for the previous integral and expanded in the same way,

$$\int_{\Omega} b_1 u_1^* d\Omega = \int_{\Gamma} b_1 q_2^* d\Gamma - \int_{\Gamma} u_2^* \frac{\partial b_1}{\partial n} d\Gamma + \int_{\Omega} u_2^* \nabla^2 b_1 d\Omega \quad (2.117)$$

Note that now a function  $b_2$  can be computed such that

$$b_2 = \nabla^2 b_1 \quad (2.118)$$

and this procedure continued as many times as desired.

The approach can be generalized by introducing two sequences of functions defined by the following recurrence formulae,

$$\begin{aligned} b_{j+1} &= \nabla^2 b_j \\ \nabla^2 u_{j+1}^* &= u_j^* \quad \text{for } j = 0, 1, 2 \dots \end{aligned} \quad (2.119)$$

The domain integral can finally be expressed as,

$$\int_{\Omega} b_0 u_0^* d\Omega = \sum_{j=0}^{\infty} \int_{\Gamma} \left( b_j q_{j+1}^* - u_{j+1}^* \frac{\partial b_j}{\partial n} \right) d\Gamma \quad (2.120)$$

where  $q_{j+1}^* = \partial u_{j+1}^* / \partial n$ .

Introducing expression (2.120) into the original boundary integral equation, the exact boundary-only formulation of the problem is obtained, *i.e.*

$$c_i u_i + \int_{\Gamma} u q_0^* d\Gamma - \int_{\Gamma} q u_0^* d\Gamma = - \sum_{j=0}^{\infty} \int_{\Gamma} \left( b_j q_{j+1}^* - u_{j+1}^* \frac{\partial b_j}{\partial n} \right) d\Gamma \quad (2.121)$$

The integrals can be evaluated numerically by subdividing the boundary  $\Gamma$  into elements as usual. As the functions  $b_j$  are at present assumed to be known functions of space, the integrals in the summation can be calculated directly. The same type of interpolation used for  $u$  and  $q$  can be applied to  $b_j$  and  $\partial b_j / \partial n$ . Then, equation (2.121) can be expressed in terms of the usual boundary element influence matrices  $\mathbf{H}$  and  $\mathbf{G}$  (which will now be called  $\mathbf{H}_0$  and  $\mathbf{G}_0$  to differentiate from the others) plus those matrices resulting from the use of higher-order fundamental solutions, which are defined as  $\mathbf{H}_{j+1}$  and  $\mathbf{G}_{j+1}$  ( $j=0,1,2,\dots$ ), *i.e.*

$$\mathbf{H}_0 \mathbf{u} - \mathbf{G}_0 \mathbf{q} = - \sum_{j=0}^{\infty} (\mathbf{H}_{j+1} \mathbf{p}_j - \mathbf{G}_{j+1} \mathbf{r}_j) \quad (2.122)$$

where the vectors  $\mathbf{p}_j$  and  $\mathbf{r}_j$  contain the function  $b_j$  and its normal derivative  $\partial b_j / \partial n$ , respectively, evaluated at the boundary nodes.

The terms of the series on the right-hand side of equation (2.122) vanish rapidly provided that the problem has been properly scaled (*i.e.* all dimensions are divided by the maximum dimension of the problem). It has been shown that the convergence of the series is very fast in a variety of practical cases. Furthermore this convergence can be easily controlled since all the terms are known and the contribution of each of them can be evaluated. It should also be pointed out that the functions  $u_j^*$  and  $q_j^*$  have no singularities for  $j = 1, 2, \dots$  and thus their integration does not require any special technique.

Finally, equation (2.122) can be solved using standard boundary element subroutines after taking into consideration the boundary conditions.

## Higher-Order Fundamental Solutions

The higher-order fundamental solutions required here are defined by the recurrence formula (2.119). This equation can easily be solved analytically when the Laplace operator is written in terms of a cylindrical (for 2-D problems) or spherical (for 3-D cases) coordinate system. For example, for 2-D problems the general form of the function  $u_j^*$  is given by the following expression

$$u_j^* = \frac{1}{2\pi} r^{2j} (A_j \ln r - B_j) \quad (2.123)$$

The coefficients  $A_j$  and  $B_j$  are obtained from the following recurrence relationships

$$A_{j+1} = \frac{A_j}{4(j+1)^2} \quad (2.124)$$

$$B_{j+1} = \frac{1}{4(j+1)^2} \left( \frac{A_j}{j+1} + B_j \right) \quad (2.125)$$

For  $j = 0$  one obtains the classical fundamental solution, i.e.  $A_0 = 1$  and  $B_0 = 0$ . Notice that formulae (2.124) and (2.125) introduce factorials into the denominators of coefficients  $A_j$  and  $B_j$  and hence guarantee their rapid convergence.

## 2.4 Computer Program 1

A complete listing of Program 1 will be given in what follows, together with a description of each subroutine. Data and output for a test problem are presented at the end of the section. This program solves the Poisson equation,  $\nabla^2 u = b$ , using linear boundary elements and up to a maximum of 200 nodes. The function  $b$  to be used in this case is a constant called CONST in the program and read as data. If  $\text{CONST} \neq 0$ , the resulting domain integral is evaluated using numerical integration over internal cells. The program may be used for equations of the type discussed in chapter 3, the right-hand side of which is a known function of  $(x, y)$ , altering only one line of the routine NECMOD which is indicated in the listing. This code is unsuitable for equations of the types discussed from chapter 4 onwards, in which the problem variable appears in the domain integral. In any case, as the far more powerful and convenient DRM method is described from the next chapter onwards, it is hoped that the reader will not need to resort further to internal cells.

If  $b = 0$  then CONST is read as zero, the Laplace equation is modelled, and the cell data omitted as in this case there is no domain integral.

Four-point Gauss quadrature is used for integration over the boundary elements. A modified Gaussian quadrature due to Hammer et al.[19] is used for integration over

## 48 The Dual Reciprocity

the internal cells. Weight factors and integration point coordinates are given as DATA in subroutine INPUT1.

The variable names in the program follow as closely as possible those used in the text. The program is in modular form and many of the subroutines will be used again in the DRM programs given in later chapters.

Since the objective of this book is a clear explanation of the dual reciprocity method and its applications, program refinements which have no bearing on the method have been avoided in the interests of simplicity.

### 2.4.1 MAINP1

Data is read in INPUT1. In routine ASSEM2 the matrices  $\mathbf{H}$  and  $\mathbf{G}$  are calculated and stored. Boundary conditions are then applied to the equation

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q} \quad (2.126)$$

to reduce it to the form

$$\mathbf{A}\mathbf{x} = \mathbf{y} \quad (2.127)$$

as explained in section 2.2.3. If  $b \neq 0$  routine NECMOD carries out the integration over internal cells to calculate the additional vector  $\mathbf{d}$ . Vectors  $\mathbf{d}$  and  $\mathbf{y}$  are added in the main program. Vector  $\mathbf{x}$  is calculated in SOLVER, producing the unknown values of  $u$  and  $q$ . INTERM calculates interior values using the now known boundary values of  $u$  and  $q$  plus the parts of the stored  $\mathbf{H}$  and  $\mathbf{G}$  matrices which relate to internal nodes. Results are printed by routine OUTPUT.

This process is summarized in figure 2.16, which shows the modules of Program 1.

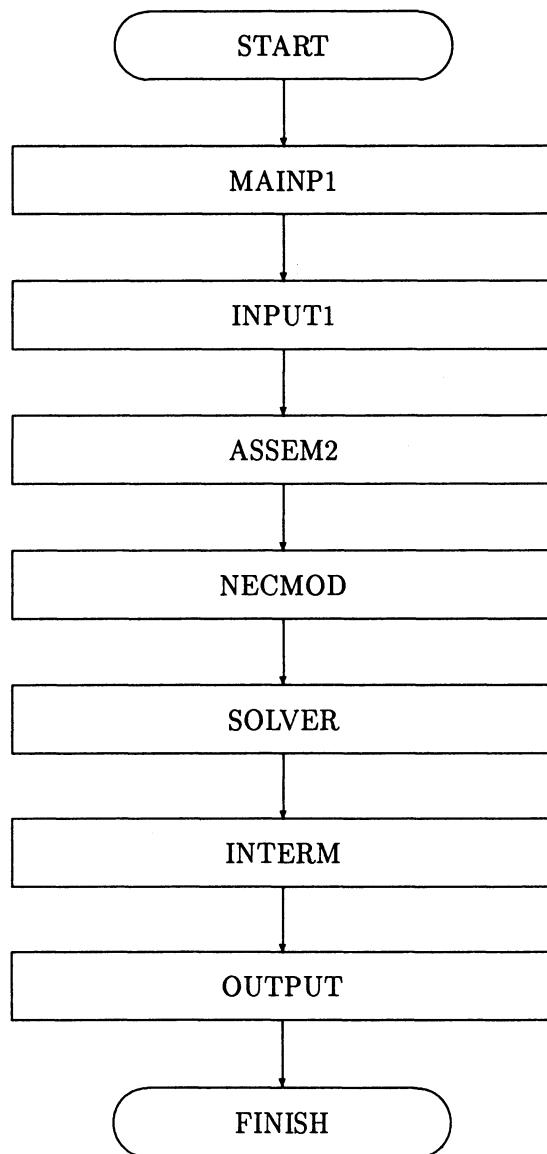


Figure 2.16: Modular Structure for Program 1

## 50 The Dual Reciprocity

```
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
COMMON/CELL/NCI,MKJ(200,3),WW(7),CP(7,3),DA(200),CONST,NPI
COMMON/FIVE/XY(200),A(200,200)
REAL LE,LJ
INTEGER CON
C
C  MASTER ROUTINE1
C
C  DATA INPUT
C
CALL INPUT1
C
C CALCULATION OF H AND G MATRICES: APPLY BOUNDARY CONDITIONS
C MATRIX IN A, RIGHT-HAND SIDE VECTOR IN XY
C
CALL ASSEM2
C
C CALCULATION OF ADDITIONAL RHS VECTOR D: POISSON EQUATION ONLY
C
IF(CONST.NE.0.) THEN
CALL NECMOD
C
C SUM XY AND D
C
DO 5 I = 1,NN
XY(I)=XY(I)+D(I)
5 CONTINUE
END IF
C
C SOLVE FOR BOUNDARY VALUES
C
CALL SOLVER
C
C PUT BOUNDARY VALUES IN APPROPRIATE ARRAY, U OR Q
C
DO 76 I=1,NN
KK=KODE(I)
IF(KK.EQ.0.OR.KK.EQ.2)THEN
U(I)=XY(I)
ELSE
Q(I)=XY(I)
END IF
76 CONTINUE
C
C CALCULATE VALUES AT INTERNAL NODES
C
IF(L.GT.0)CALL INTERM
C
C WRITE RESULTS
C
CALL OUTPUT
C
```

```
STOP
END
```

## 2.4.2 Subroutine INPUT1

This subroutine differs from that which will be used in the subsequent DRM programs in that, in the present case, cell data needs to be read. All of the material in COMMON/CELL refers to the internal cells. The integration points and weight factors are defined in DATA: POEI, FDEP respectively for integration over elements and CP, WW for integration over cells. First, the constant  $b$  or CONST is read. If zero, the Laplace equation is being modelled, otherwise the Poisson equation. Next, global data is read: NN Number of boundary nodes, NE Number of boundary elements, L Number of internal nodes. If  $b \neq 0$  NCI Number of internal cells is also needed.

Next  $u$  and  $q$  are initialized. If  $b \neq 0$  the cell nodes are defined, the numbers being stored in MKJ to avoid confusion with CON which is used for the nodes of the boundary elements. For the triangular cells used here MKJ(I,1), MKJ(I,2) and MKJ(I,3) will contain the numbers of the three nodes of cell I. These three points should be given in anticlockwise order starting from any one of them. The coordinates of the boundary nodes are then given in clockwise order so that the program can store the numbers of the nodes of each element in CON correctly, followed by the coordinates of all the internal cell vertices. Next, the program calculates the area of the internal cells which are stored in array DA.

Then boundary conditions are dealt with. Boundary condition type is stored in KODE as follows: 1  $u$  known,  $q$  unknown; 2  $q$  known,  $u$  unknown. Following KODE is the parameter VAL, which is the known value of  $u$  or  $q$  respectively. At internal nodes,  $u$  is unknown and  $q$  not defined. There is no need to read KODE for internal nodes, the routine will automatically assign these to zero.

All input data is then printed.

```
SUBROUTINE INPUT1
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
COMMON/FIVE/XY(200),A(200,200)
COMMON/CELL/NCI,MKJ(200,3),WW(7),CP(7,3),DA(200),CONST,NPI
C
C COORDINATES OF NUMERICAL INTEGRATION POINTS FOR BOUNDARY ELEMENTS IN POEI.
C WEIGHT FACTORS FOR BOUNDARY ELEMENTS IN FDEP. 4-POINT GAUSS INTEGRATION USED.
C
DATA POEI/0.86113631,-0.86113631,0.33998104,-0.33998104/
DATA FDEP/0.34785485,0.34785485,0.65214515,0.65214515/
C
C COORDINATES OF NUMERICAL INTEGRATION POINTS FOR TRIANGULAR CELLS IN CP.
C WEIGHT FACTORS FOR CELLS IN WW. 7-POINT MODIFIED GAUSS SCHEME USED.
C
DATA WW/0.22500000,0.12592918,0.12592918,0.12592918,
* 0.13239416,0.13239416,0.13239416/
DATA CP/0.33333333,0.79742699,0.10128651,0.10128651,
* 0.05971587,0.47014206,0.47014206,0.33333333,0.10128651,
```

## 52 The Dual Reciprocity

```
* 0.79742699,0.10128651,0.47014206,0.05971587,0.47014206,
* 0.33333333,0.10128651,0.10128651,0.79742699,
* 0.47014206,0.47014206,0.05971587/

      REAL LE,LJ
      INTEGER CON

C
C NUMBER OF INTEGRATION POINTS PER BOUNDARY ELEMENT IN NI,
C NUMBER OF INTEGRATION POINTS PER INTERNAL CELL IN NPI
C
NI=4
NPI=7

C
C DEFINE RIGHT-HAND SIDE OF EQUATION
C
C CONST = 0; LAPLACE EQUATION
C CONST /= 0; POISSON EQUATION
C
      READ(5,132) CONST
132   FORMAT(F11.5)
C
C WRITE HEADING
C
      WRITE(6,133) CONST
133   FORMAT(//'EQUATION NABLA2U=',F5.2,//)
C
C NN  NUMBER OF BOUNDARY NODES
C NE  NUMBER OF BOUNDARY ELEMENTS
C L   NUMBER OF INTERNAL NODES
C
      READ(5,1234) NN,NE,L
1234  FORMAT(3I4)
C
C NCI NUMBER OF INTERNAL CELLS
C
      IF(CONST.NE.0.)THEN
      READ(5,134) NCI
134   FORMAT(I4)
      ELSE
      NCI=0
      END IF
C
C CON CONNECTIVITY OF BOUNDARY ELEMENTS
C
      DO 1007 I = 1,NN
      CON(I,1) = I
      CON(I,2) = I + 1
1007   CONTINUE
      CON(NN,2) = 1
C
C INITIALIZE VECTORS FOR INTERNAL NODES
C
      DO 1008 I = NN+1,NN+L
```

```

KODE(I) = 0
U(I) = 0.
Q(I) = 0.
1008    CONTINUE
C
C READ CONNECTIVITY OF TRIANGULAR INTERNAL CELLS
C
IF(CONST.NE.0.)THEN
READ(5,1014)(J,(MKJ(J,I),I=1,3),JJ=1,NCI)
1014    FORMAT(16I4)
END IF
C
C READ COORDINATES OF NODES
C
DO 2 I=1,NN+L
READ(5,3) X(I),Y(I)
3    FORMAT(2F13.6)
2    CONTINUE
C
C CALCULATE AREA OF INTERNAL CELLS
C
DO 49 J = 1,NCI
I1 = MKJ(J,1)
I2 = MKJ(J,2)
I3 = MKJ(J,3)
AB1 = Y(I2) - Y(I3)
AB2 = Y(I3) - Y(I1)
AB4 = X(I3) - X(I2)
AB5 = X(I1) - X(I3)
DA(J)= ABS((AB1*AB5-AB2*AB4)/2.0)
49    CONTINUE
C
C READ TYPE AND VALUE OF BOUNDARY CONDITION
C VAL CONTAINS THE VALUE OF THE BOUNDARY CONDITION
C
C VALUES OF KODE(I)
C KODE(I) = 1 U KNOWN AT POINT I
C KODE(I) = 2 Q KNOWN AT POINT I
C
DO 4 I=1,NN
READ(5,5) KODE(I),VAL
IF(KODE(I).EQ.1) U(I) = VAL
IF(KODE(I).EQ.2) Q(I) = VAL
4    CONTINUE
5    FORMAT(I3,F11.5)
C
C PRINT INPUT DATA
C
WRITE(6,8) NN
8    FORMAT(' NUMBER OF BOUNDARY NODES = ',I3/)
WRITE(6,9) NE
9    FORMAT(' NUMBER OF BOUNDARY ELEMENTS = ',I3/)
WRITE(6,10) L

```

## 54 The Dual Reciprocity

```

10      FORMAT(' NUMBER OF INTERNAL NODES = ',I3)
      WRITE(6,432) NCI
432    FORMAT(' NUMBER OF INTERNAL CELLS = ',I3//)
      WRITE(6,11)
11      FORMAT(' DATA FOR BOUNDARY NODES')
      WRITE(6,12)
12      FORMAT(' NODE   X       Y     TYPE U     Q  ')
      WRITE(6,1012) (I,X(I),Y(I),KODE(I),U(I),Q(I),I=1,NN)
1012   FORMAT(I3,2F10.6,I3,2F6.3)
      WRITE(6,1187)
1187   FORMAT(' COORDINATES OF INTERNAL NODES')
      WRITE(6,1288)
1288   FORMAT(' NODE   X       Y  ')
      WRITE(6,1112) (I,X(I),Y(I),I=NN+1,NN+L)
1112   FORMAT(I3,2F10.6)
      WRITE(6,6)
6       FORMAT(' ELEMENT DATA')
      WRITE(6,7)
7       FORMAT('   NO   NODE 1 NODE 2      LENGTH')
C
C CALCULATE ELEMENT LENGTH IN LE
C
      DO 33 K = 1,NE
      LE(K) = SQRT((X(CON(K,2))-X(CON(K,1)))**2 +
1        (Y(CON(K,2)) - Y(CON(K,1)))**2)
      WRITE(6,48) K,CON(K,1),CON(K,2),LE(K)
48      FORMAT(3I6,F16.6)
33      CONTINUE
C
C PRINT CELL DATA
C
      IF(CONST.NE.0.)THEN
      WRITE(6,567)
567    FORMAT(' CELL DATA')
      WRITE(6,678)
678    FORMAT('   CELL NODE1 NODE2 NODE3  AREA')
      WRITE(6,1713)(I,(MKJ(I,J),J=1,3),DA(I),I=1,NCI)
1713   FORMAT(4I7,F10.4)
      END IF
C
      RETURN
END

```

### 2.4.3 Subroutine ASSEM2

The name of this subroutine comes from ASSEMble 2-node linear boundary elements, and will also be used later on in programs 2 and 3. First, the matrices  $\mathbf{H}$  and  $\mathbf{G}$  are calculated and stored. Boundary conditions are applied to the equation  $\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q}$  to reduce it to the form  $\mathbf{A}\mathbf{x} = \mathbf{y}$ . The DO loop control variable J1 will be the total number of the source points, i.e.  $NN+L$ , from which integration is carried out over the entire boundary. XX, YY are the cartesian coordinates of each integration point. When the

extreme point of an element concides with node J1, the corresponding coefficients of H for the element are zero, and those of G are calculated using exact integration. The exact value of G is stored in GE. Diagonal coefficients of H are calculated by adding the off-diagonal terms in the same row and the result is stored in CC. H is stored assembled but G is stored unassembled on account of the possible discontinuity of the outward normals at boundary nodes (see section 2.6). Boundary conditions are then applied. Known values of  $u$  are multiplied by the respective coefficient of H and subtracted from XY and known values of  $q$  are multiplied by the respective coefficient of G and added to XY. After solution this vector will contain the unknowns. Coefficients of H and G which correspond to unknown values of  $u$  and  $q$  are stored in the appropriate position in A. Note that, in forming matrix A and the corresponding vector XY, only the first NN lines of H and G are used. The last L lines are used in routine INTERM.

```

SUBROUTINE ASSEM2
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
COMMON/FIVE/XY(200),A(200,200)
COMMON/DRM/HH(200,200),GG(200,400)
REAL LE,LJ
C
C ASSEMBLE H AND G MATRICES FOR LINEAR BOUNDARY ELEMENTS
C APPLY BOUNDARY CONDITIONS TO FORM AX=Y
C
      INTEGER CON
      PI = 3.141592654
C
      DO 1 J1 = 1,NN+L
      XI = X(J1)
      YI = Y(J1)
      XY(J1) = 0.
      DO 21 J = 1,NN
      HH(J1,J)= 0.
      A(J1,J) = 0.
21    CC=0.
      DO 2 J2 = 1,NE
      LJ = LE(J2)
      N1 = CON(J2,1)
      N2 = CON(J2,2)
      X1 = X(N1)
      X2 = X(N2)
      Y1 = Y(N1)
      Y2 = Y(N2)
      H1 = 0
      H2 = 0
      G1 = 0
      G2 = 0
      IF(J1.EQ.(N1.OR.N2))GO TO 4
C
C NON-SINGULAR INTEGRALS USING NUMERICAL INTEGRATION
C
      DO 3 J3 = 1,NI

```

## 56 The Dual Reciprocity

```
E = POEI(J3)
W = FDEP(J3)
XX = X1 + (1.+E)*(X2-X1)/2.
YY = Y1 + (1.+E)*(Y2-Y1)/2.
R = SQRT((XI-XX)**2 + (YI-YY)**2)
PP = ((XI-XX)*(Y1-Y2)+(YI-YY)*(X2-X1))/(R*R*4.*PI)*W
C
C ELEMENTS OF H AND G CALCULATED USING FORMULAS 2.46-47, 2.49-50
C
H1 = H1 + (1.-E)*PP/2.0
H2 = H2 + (1.+E)*PP/2.0
PP = ALOG(1./R)/(4.*PI)*LJ*W
G1 = G1 + (1.-E)*PP/2.
G2 = G2 + (1.+E)*PP/2.
3      CONTINUE
C
C DIAGONAL TERMS CALCULATED SUMMING OFF-DIAGONAL COEFFICIENTS
C
CC = CC - H1 - H2
4      CONTINUE
C
C EXACT INTEGRATION FOR G WHEN I AND J ARE ON THE SAME ELEMENT
C
GE = LJ*(3./2.-ALOG(LJ))/(4.*PI)
C
C STORE FULL G AND H MATRICES FOR USE WITH DRM
C H ASSEMBLED, G UNASSEMBLED
C
IF(N1.EQ.J1) G1=GE
IF(N2.EQ.J1) G2=GE
HH(J1,N1)=HH(J1,N1)+H1
HH(J1,N2)=HH(J1,N2)+H2
GG(J1,2*J2-1) = G1
GG(J1,2*J2)    = G2
C
C APPLY BOUNDARY CONDITIONS: HU=GQ BECOMES AX=Y
C FOR TYPES OF BOUNDARY CONDITIONS SEE ROUTINE INPUT1
C KNOWN TERMS ARE MULTIPLIED AND PUT IN VECTOR XY
C
KK=KODE(N1)
IF(KK.EQ.0.OR.KK.EQ.2)THEN
XY(J1) = XY(J1) + Q(N1)*G1
A(J1,N1) = A(J1,N1) + H1
ELSE
XY(J1) = XY(J1) - U(N1)*H1
A(J1,N1) = A(J1,N1) - G1
END IF
KK=KODE(N2)
IF(KK.EQ.0.OR.KK.EQ.2)THEN
XY(J1) = XY(J1) + Q(N2)*G2
A(J1,N2) = A(J1,N2) + H2
ELSE
XY(J1) = XY(J1) - U(N2)*H2
```

```

A(J1,N2) = A(J1,N2) - G2
END IF
2      CONTINUE
C
C APPLY BOUNDARY CONDITIONS TO DIAGONAL TERM
C
HH(J1,J1) = CC
KK=KODE(J1)
IF(KK.EQ.0.OR.KK.EQ.2)THEN
A(J1,J1) = CC
ELSE
XY(J1) = XY(J1) - U(J1)*CC
END IF
C
1      CONTINUE
RETURN
END

```

## 2.4.4 Subroutine NECMOD

This routine is called only if  $b \neq 0$  to evaluate the domain integrals using internal cells. In program 2, this will be substituted by a routine RHSVEC which calculates the same integral using the DRM. For each source point I, integration is carried out over the whole domain, i.e. over all the internal cells. The domain integral is calculated numerically using formula (2.78) and the result of the integration is stored in array D.

```

SUBROUTINE NECMOD
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 COM(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
COMMON/CELL/MCI,MKJ(200,3),WW(7),CP(7,3),DA(200),CONST,NPI
COMMON/FIVE/XY(200),A(200,200)
REAL LE,LJ
INTEGER COM
PI = 3.141592654
DO 1 I = 1,NN+L
C
C INTEGRATION OVER WHOLE DOMAIN FROM POINT I
C
XI = X(I)
YI = Y(I)
FF = 0.
DO 20 J1 = 1,MCI
C
C AR AREA OF CELL; N1,N2,N3 CONNECTIVITY OF CELL
C X1,X2,X3,Y1,Y2,Y3 COORDINATES OF CELL VERTICES
C
AR = DA(J1)
N1 = MKJ(J1,1)
N2 = MKJ(J1,2)
N3 = MKJ(J1,3)
X1 = X(N1)
X2 = X(N2)

```

```

X3 = X(N3)
Y1 = Y(N1)
Y2 = Y(N2)
Y3 = Y(N3)
DO 30 J3 = 1,NPI
C1 = CP(J3,1)
C2 = CP(J3,2)
C3 = CP(J3,3)

C
C XP,YP COORDINATES OF INTEGRATION POINT
C
XP = X1*C1 + X2*C2 + X3*C3
YP = Y1*C1 + Y2*C2 + Y3*C3

C
C DX, DY COMPONENTS OF R
C
DX = XI-XP
DY = YI-YP
R = SQRT(DX*DX + DY*DY)

C
C NEXT LINE IS TO BE ALTERED IF EQUATIONS OF THE TYPE NABLA2U=B(X,Y)
C          OF CHAPTER 3 ARE BEING PROGRAMMED
C
C SUMMATION OVER CELLS DONE USING FORMULA (2.78)
C
FF = FF - WW(J3)*AR*ALOG(1./R)*CONST/(2.*PI)
30      CONTINUE
20      CONTINUE
C
C ON COMPLETION OF INTEGRATION OVER WHOLE DOMAIN FROM POINT I
C THE RESULT IS PUT IN THE APPROPRIATE POSITION IN VECTOR D
C
D(I) =FF
1      CONTINUE
RETURN
END

```

## 2.4.5 Subroutine SOLVER

This is a standard routine with the characteristics described in the comment statements. A discussion of the mechanism of Gauss elimination is outside the scope of this text, but the reader is referred to [20] where other types of solver may also be found. The results are stored in XY.

```

SUBROUTINE SOLVER
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/XY(200),A(200,200)
REAL LE,LJ
INTEGER CON
C
C NON-PIVOTAL GAUSS ELIMINATION SOLVER FOR AX=Y

```

```

C MATRIX A STARTS IN A, KNOWN VECTOR Y STARTS IN XY
C RESULTS FOR X RETURNED IN XY.
C SYSTEM MUST BE OF SIZE NN BY NN
C
C TRIANGULARIZE NONSYMMETRIC FULL MATRIX
C
      DO 1 M = 1,NN-1
      AM = 1.0/A(M,M)
      DO 21 J = M+1,NN
      CD = A(M,J)
      A(M,J) = CD*AM
21     DO 2 N = M+1,NN
      ALM = A(N,M)
      DO 3 KK = M+1,NN
3       A(N,KK) = A(N,KK) - A(M,KK)*ALM
2       CONTINUE
1       CONTINUE
C
C BACK SUBSTITUTION
C
      DO 10 M = 1,NN-1
      AM = 1.0/A(M,M)
      XY(M) = XY(M)*AM
      DO 20 N = M+1,NN
      ALM = A(N,M)
      XY(N) = XY(N) - XY(M)*ALM
20     CONTINUE
10    CONTINUE
      XY(NN) = XY(NN)/A(NN,NN)
      NI = NN
300   NI = NI - 1
      IF(NI.NE.0)THEN
          DO 6 J = NI + 1,NN
          XY(NI) = XY(NI) - A(NI,J)*XY(J)
6       CONTINUE
      GOTO 300
      END IF
      RETURN
      END

```

## 2.4.6 Subroutine INTERM

This routine calculates results at internal points once the boundary values are available. Since H and G have been stored and array D contains the results of integration over the cells, the process is of simple array multiplication. Note the difference in treatment of H and G due to the latter being unassembled.

```

SUBROUTINE INTERM
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
COMMON/DRM/HH(200,200),GG(200,400)
REAL LE,LJ

```

```

      INTEGER COM
C
C CALCULATES VALUES OF POTENTIAL AT INTERNAL POINTS ONCE
C BOUNDARY SOLUTION IS KNOWN
C
      DO 1 I = MN+1,MN+L
         U(I) = 0.
      DO 2 K = 1,MN-1
         U(I) = U(I)+(GG(I,2*K)+GG(I,(2*K+1)))*Q(K+1)
2      CONTINUE
         U(I) = U(I)+(GG(I,1)+GG(I,(2*NE)))*Q(1)
      DO 3 K = 1,MN
         U(I) = U(I)-HH(I,K)*U(K)
3      CONTINUE
         U(I) = U(I) + D(I)
1      CONTINUE
      RETURN
END

```

### 2.4.7 Subroutine OUTPUT

This routine prints all results. It is the final routine to be called in the main program.

```

SUBROUTINE OUTPUT
COMMON/ONE/MN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
      INTEGER COM
      REAL LE
C
C WRITE RESULTS
C
      WRITE (6,1010)
1010 FORMAT('' BOUNDARY RESULTS '')
      WRITE (6,1020)
1020 FORMAT(' NODE      FUNCTION      DERIVATIVE')
      WRITE(6,101)(J1,U(J1),Q(J1),J1=1,MN)
101  FORMAT(1X,I4,2F15.6)
      WRITE (6,1018)
1018 FORMAT('' RESULTS AT INTERIOR NODES')
      WRITE (6,1028)
1028 FORMAT(' NODE      FUNCTION      ')
      WRITE(6,102)(J,U(J),J=MN+1,MN+L)
102  FORMAT(I5,F15.6)
      RETURN
END

```

### 2.4.8 Results of a Test Problem

The data for Program 1 is listed below. The data sets (iii) and (iv) are omitted if  $b = 0$ , i.e. if the Laplace equation is being modelled.

i) 1 line to define  $b$  in  $\nabla^2 u = b$ .

Internal name CONST  
FORMAT(F11.5)

ii) 1 line of global data, Number of Boundary Nodes, Number of Boundary Elements, Number of Internal Nodes.

Internal names NN, NE, L  
FORMAT(3I4)

(iii) 1 line to define Number of Internal Cells.

Internal name NCI  
FORMAT(I4)

(iv) NCI/4 lines (4 cells per line) with the following data: Cell Number, Node 1, Node 2, Node 3.

Internal name MKJ  
FORMAT(16I4)

v) NN+L lines to define coordinates of each node. Two coordinates (one node) per line. Boundary nodes in clockwise order.

Internal names X,Y  
FORMAT(2F13.6)

vi) NN lines to define boundary conditions. One boundary condition type and one value per line.

Internal names KODE, VAL  
FORMAT(I3,2F11.5)

## Data for Poisson Problem

The data and output files listed below are for the case of the Poisson problem considered in section 2.3.2 (figure 2.12).

```
-2.
16 16 17
48
 1  2   1 17   2 16 17   1   3   9   8 23   4   9 23 10
 5 28 17 16   6 18   2 17   7 24 10 23   8 22 23   8
 9 15 28 16 10   3   2 18 11   7 22   8 12 11 10 24
```



1 0.  
1 0.  
1 0.  
1 0.  
1 0.

## Output of Poisson Problem

EQUATION NABLA2U=-2.00

NUMBER OF BOUNDARY NODES = 16

NUMBER OF BOUNDARY ELEMENTS = 16

NUMBER OF INTERNAL NODES = 17

NUMBER OF INTERNAL CELLS = 48

### DATA FOR BOUNDARY NODES

NODE	X	Y	TYPE	U	Q
1	2.000000	0.000000	1	0.000	0.000
2	1.705706	-0.522150	1	0.000	0.000
3	1.178800	-0.807841	1	0.000	0.000
4	0.597614	-0.954310	1	0.000	0.000
5	0.000000	-1.000000	1	0.000	0.000
6	-0.597614	-0.954310	1	0.000	0.000
7	-1.178800	-0.807841	1	0.000	0.000
8	-1.705706	-0.522150	1	0.000	0.000
9	-2.000000	0.000000	1	0.000	0.000
10	-1.705706	0.522150	1	0.000	0.000
11	-1.178800	0.807841	1	0.000	0.000
12	-0.597614	0.954310	1	0.000	0.000
13	0.000000	1.000000	1	0.000	0.000
14	0.597614	0.954310	1	0.000	0.000
15	1.178800	0.807841	1	0.000	0.000
16	1.705706	0.522150	1	0.000	0.000

### COORDINATES OF INTERNAL NODES

NODE	X	Y
17	1.500000	0.000000
18	1.200000	-0.350000
19	0.600000	-0.450000
20	0.000000	-0.450000
21	-0.600000	-0.450000
22	-1.200000	-0.350000

## 64 The Dual Reciprocity

23	-1.500000	0.000000
24	-1.200000	0.350000
25	-0.600000	0.450000
26	0.000000	0.450000
27	0.600000	0.450000
28	1.200000	0.350000
29	0.900000	0.000000
30	0.300000	0.000000
31	0.000000	0.000000
32	-0.300000	0.000000
33	-0.900000	0.000000

### ELEMENT DATA

NO	NODE 1	NODE 2	LENGTH
1	1	2	0.599374
2	2	3	0.599374
3	3	4	0.599358
4	4	5	0.599358
5	5	6	0.599358
6	6	7	0.599358
7	7	8	0.599374
8	8	9	0.599374
9	9	10	0.599374
10	10	11	0.599374
11	11	12	0.599358
12	12	13	0.599358
13	13	14	0.599358
14	14	15	0.599358
15	15	16	0.599374
16	16	1	0.599374

### CELL DATA

CELL	NODE1	NODE2	NODE3	AREA
1	2	1	17	0.1305
2	16	17	1	0.1305
3	9	8	23	0.1305
4	9	23	10	0.1305
5	28	17	16	0.1143
6	18	2	17	0.1143
7	24	10	23	0.1143
8	22	23	8	0.1143
9	15	28	16	0.1176
10	3	2	18	0.1176
11	7	22	8	0.1176
12	11	10	24	0.1176
13	29	17	28	0.1050

14	29	18	17	0.1050
15	23	22	33	0.1050
16	23	33	24	0.1050
17	27	28	15	0.1363
18	19	3	18	0.1363
19	22	7	21	0.1363
20	11	24	25	0.1363
21	27	15	14	0.1464
22	19	4	3	0.1464
23	21	7	6	0.1464
24	25	12	11	0.1464
25	27	29	28	0.1200
26	19	18	29	0.1200
27	22	21	33	0.1200
28	33	25	24	0.1200
29	27	30	29	0.1350
30	30	19	29	0.1350
31	25	33	32	0.1350
32	33	21	32	0.1350
33	26	27	14	0.1513
34	20	4	19	0.1513
35	21	6	20	0.1513
36	12	25	26	0.1513
37	26	30	27	0.1350
38	20	19	30	0.1350
39	21	20	32	0.1350
40	32	26	25	0.1350
41	13	26	14	0.1643
42	5	4	20	0.1643
43	5	20	6	0.1643
44	26	13	12	0.1643
45	31	30	26	0.0675
46	31	20	30	0.0675
47	31	32	20	0.0675
48	31	26	32	0.0675

## BOUNDARY RESULTS

NODE	FUNCTION	DERIVATIVE
1	0.000000	-0.733859
2	0.000000	-1.046938
3	0.000000	-1.378876
4	0.000000	-1.549611
5	0.000000	-1.611697
6	0.000000	-1.549612
7	0.000000	-1.378875
8	0.000000	-1.046941
9	0.000000	-0.733854
10	0.000000	-1.046941
11	0.000000	-1.378876
12	0.000000	-1.549613

## 66 The Dual Reciprocity

13	0.000000	-1.611696
14	0.000000	-1.549611
15	0.000000	-1.378877
16	0.000000	-1.046939

### RESULTS AT INTERIOR NODES

#### NODE      FUNCTION

17	0.331468
18	0.401583
19	0.557093
20	0.629051
21	0.557093
22	0.401583
23	0.331468
24	0.401583
25	0.557093
26	0.629051
27	0.557093
28	0.401583
29	0.626685
30	0.772933
31	0.791648
32	0.772933
33	0.626685

## 2.5 References

1. Brebbia, C. A. (Ed)  
Recent Advances in Boundary Element Methods, Pentech Press, London, 1978.
2. Brebbia, C. A.  
The Boundary Element Method for Engineers, Pentech Press, London, 1978.
3. Brebbia, C. A. and Walker, S.  
Boundary Element Techniques in Engineering, Newnes-Butterworths, London, 1979.
4. Brebbia, C. A., Telles, J. F. C. and Wrobel, L. C.  
Boundary Element Techniques, Springer-Verlag, Berlin and New York, 1984.
5. Brebbia, C. A. and Dominguez, J.  
Boundary Elements: An Introductory Course, Computational Mechanics Publications, Southampton and McGraw-Hill, New York, 1989.
6. Jaswon, M. A.  
Integral Equation Methods in Potential Theory I, *Proceedings of the Royal Society*, Series A, Vol. 275, pp 23-32, 1963.
7. Symm, G. T.  
Integral Equation Methods in Potential Theory II, *Proceedings of the Royal Society*, Series A, Vol. 275, pp 33-46, 1963.

8. Jaswon, M. A. and Ponter, A. R. S.  
An Integral Equation Solution of the Torsion Problem, *Proceedings of the Royal Society, Series A*, Vol. 273, pp 237-246, 1963.
9. Nowak, A. J.  
The Multiple Reciprocity Method of Solving Transient Heat Conduction Problems, in *Boundary Elements XI*, Vol. 2, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.
10. Brebbia, C. A. and Nowak, A. J.  
A New Approach for Transforming Domain Integrals to the Boundary, in *Numerical Methods in Engineering*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.
11. Nowak, A. J. and Brebbia, C. A.  
The Multiple Reciprocity Method: A New Approach for Transforming BEM Domain Integrals to the Boundary, *Engineering Analysis*, Vol. 6, No. 3, pp 164-167, 1989.
12. Nardini, D. and Brebbia, C. A.  
A New Approach to Free Vibration Analysis Using Boundary Elements, in *Boundary Element Methods in Engineering*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1982.
13. Brebbia, C. A. and Nardini, D.  
Dynamic Analysis in Solid Mechanics by an Alternative Boundary Element Procedure, *International Journal of Soil Dynamics and Earthquake Engineering*, Vol. 2, No. 4, pp 228-233, 1983.
14. Nardini, D. and Brebbia, C. A.  
Boundary Integral Formulation of Mass Matrices for Dynamic Analysis, in *Topics in Boundary Element Research*, Vol. 2, Springer-Verlag, Berlin and New York, 1985.
15. Danson, D. J., Brebbia, C. A. and Adey, R. A.  
BEASY, A Boundary Element Analysis System, in *Finite Element Systems Handbook*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1985.
16. Gipson, G. S.  
Boundary Element Fundamentals - Basic Concepts and Recent Developments in the Poisson Equation, Computational Mechanics Publications, Southampton, 1987.
17. Lebedev, N. N., Skalskaya, I. P. and Ulfyans, Y. S.  
Worked Problems in Applied Mathematics, translated by R. A. Silverman, Dover Publications, New York, 1965.
18. Azevedo, J. P. S. and Brebbia, C. A.  
An Efficient Technique for Reducing Domain Integrals to the Boundary, in *Boundary Elements X*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.

## 68 The Dual Reciprocity

19. Hammer, P. C., Marlowe, O. J. and Stroud, A. H.  
Numerical Integration over Simplexes and Cones, *Mathematics of Computation*, Vol. 10, pp 130-139, 1956.
20. Press, W. H., Flannery, B. P., Teukolsky, S. A. and Vetterling, W. T.  
Numerical Recipes: The Art of Scientific Computation, Cambridge University Press, Cambridge, 1986.

# Chapter 3

## The Dual Reciprocity Method for Equations of the Type $\nabla^2 u = b(x, y)$

### 3.1 Equation Development

#### 3.1.1 Preliminary Considerations

In early BEM applications there was always a requirement that a fundamental solution for the problem under consideration had to be available. This fundamental solution had to take into account all the terms in the governing equation in order to avoid domain integrals in the formulation of the boundary integral equation, otherwise internal cells had to be defined [1].

With the recent developments in the method, fundamental solutions to many different equations have been found [2]; however, this does not ensure that one is available for any given case. Also, in many instances, it is inconvenient to alter a program by incorporating a new fundamental solution simply because the user wishes to study a slightly different differential equation. The use of cells to evaluate domain integrals implies an internal discretization which considerably increases the amount of data needed to run the program and hence the method loses some of its attraction in relation to the Finite Element Method or other domain techniques. The use of the Boundary Element Method with domain integrals evaluated by integration over internal cells has been explained in chapter 2.

In order to avoid the problems mentioned above, it is desirable to have available a Boundary Element technique that:

- Enables a “boundary-only” solution to be obtained, *i.e.* without discretizing the domain into cells;
- Does not depend on obtaining a new fundamental solution for each case;
- Can be applied using a similar approach in each case like in Finite Element analysis.

There are at present at least five techniques which aim to satisfy the above conditions:

1. Analytical integration of the domain integral;
2. The use of Fourier expansions;
3. The Galerkin Vector technique;
4. The Multiple Reciprocity Method;
5. The Dual Reciprocity Method.

The use of analytical integration is comparatively recent, and although it is an accurate method, its application is restricted to very simple cases [3]. The Fourier transform method is not so straightforward in most cases as the calculation of the coefficients of the Fourier expansions is computationally cumbersome, however the method has been applied with some success [4]. The Galerkin Vector technique is able to transform certain types of domain integral into boundary integrals but also has restricted application [2]. The Multiple Reciprocity Method is in a sense a recent, more powerful extension of this [5].

The Dual Reciprocity Method was first proposed in 1982 [6] and it can be said to satisfy all of the conditions cited above: *i.e.* it may be used with any type of fundamental solution and does not need internal cells, however it permits the definition of internal nodes if the user so desires. In addition, it is by far the easiest of the above methods to use in practice.

In this chapter, the Dual Reciprocity Method will be developed for the Poisson equation, which is the simplest type of non-homogeneous equation. Several different source terms are considered. A computer program will be presented with which the results for the examples analysed in this chapter can be obtained. This program is written in modular form as was Program 1 in section 2.4.

### 3.1.2 Mathematical Development of the DRM for the Poisson Equation

The DRM is explained in this section with reference to the Poisson equation

$$\nabla^2 u = b \quad (3.1)$$

In the present chapter  $b = b(x, y)$ , that is,  $b$  is considered to be a known function of position. In chapter 4,  $b$  will be considered as a function of the potential as well as position, *i.e.*  $b = b(x, y, u)$  and in chapter 5,  $b$  will also be allowed to be a function of time, *i.e.*  $b = b(x, y, u, t)$ , thus extending the method to non-linear, transient and other problems.

The solution to equation (3.1) can be expressed as the sum of the solution of a homogeneous, Laplace's equation and a particular solution  $\hat{u}$ , as seen in section 2.3.4, such that

$$\nabla^2 \hat{u} = b \quad (3.2)$$

It is generally difficult to find a solution  $\hat{u}$  that satisfies the above, particularly in the case of non-linear or time-dependent problems. The Dual Reciprocity Method proposes the use of a series of particular solutions  $\hat{u}_j$  instead of a single function  $\hat{u}$ . The number of  $\hat{u}_j$  used is equal to the total number of nodes in the problem. If there are  $N$  boundary nodes and  $L$  internal nodes, there will be  $N + L$  values of  $\hat{u}_j$ , see figure 3.1.

The following approximation for  $b$  is then proposed:

$$b \simeq \sum_{j=1}^{N+L} \alpha_j f_j \quad (3.3)$$

where the  $\alpha_j$  are a set of initially unknown coefficients and the  $f_j$  are approximating functions. The particular solutions  $\hat{u}_j$ , and the approximating functions  $f_j$ , are linked through the relation

$$\nabla^2 \hat{u}_j = f_j \quad (3.4)$$

The functions  $f_j$  in (3.3) can be compared to the usual interpolation functions  $\phi_i$  in expansions such as,

$$u = \sum \phi_i u_i \quad (3.5)$$

which are used on the boundary elements themselves and in Finite Element analysis. Equation (3.3) is exact at the node points, as is equation (3.5).

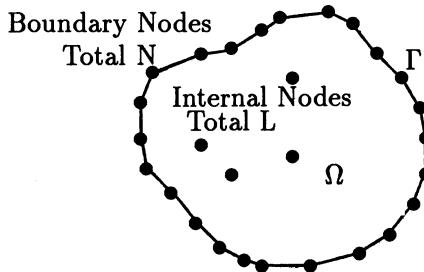


Figure 3.1: Boundary and Internal Nodes

The expansion (3.3) may then be considered as valid over the whole problem domain as in the case of one large superelement. The functions  $f_j$  are geometry-dependent, as are the  $\phi_i$  in equation (3.5). At present, no restriction will be placed on these functions, and in fact many different types may be used, each of which results

in a different function  $\hat{u}_j$  as determined from (3.4). The question of which type of function  $f_j$  to use will be considered in detail in section 3.2.

Substituting equation (3.4) into (3.3) gives

$$b = \sum_{j=1}^{N+L} \alpha_j (\nabla^2 \hat{u}_j) \quad (3.6)$$

Equation (3.6) can be substituted into the original equation (3.1) to give the following expression:

$$\nabla^2 u = \sum_{j=1}^{N+L} \alpha_j (\nabla^2 \hat{u}_j) \quad (3.7)$$

The source term  $b$  of (3.1) has been replaced in equation (3.7) by a summation of products of coefficients  $\alpha_j$  and the Laplacian operating on the particular solutions  $\hat{u}_j$ .

The procedure seen in chapter 2 for developing the boundary element method for the Laplace equation will now be applied. Equation (3.7) can be multiplied by the fundamental solution and integrated over the domain, producing

$$\int_{\Omega} (\nabla^2 u) u^* d\Omega = \sum_{j=1}^{N+L} \alpha_j \int_{\Omega} (\nabla^2 \hat{u}_j) u^* d\Omega \quad (3.8)$$

Note that the same result may be obtained starting from equation

$$\int_{\Omega} (\nabla^2 u) u^* d\Omega = \int_{\Omega} b u^* d\Omega \quad (3.9)$$

and substituting for  $b$  (equation (3.6)).

Integrating by parts the Laplacian terms in (3.8), as shown in chapter 2, produces the following integral equation for each source node  $i$ ,

$$c_i u_i + \int_{\Gamma} q^* u d\Gamma - \int_{\Gamma} u^* q d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left( c_i \hat{u}_{ij} + \int_{\Gamma} q^* \hat{u}_j d\Gamma - \int_{\Gamma} u^* \hat{q}_j d\Gamma \right) \quad (3.10)$$

The term  $\hat{q}_j$  in equation (3.10) is defined as  $\hat{q}_j = \partial \hat{u}_j / \partial n$ , where  $n$  is the unit outward normal to  $\Gamma$ , and can be expanded to

$$\hat{q}_j = \frac{\partial \hat{u}_j}{\partial x} \frac{\partial x}{\partial n} + \frac{\partial \hat{u}_j}{\partial y} \frac{\partial y}{\partial n} \quad (3.11)$$

Note that equation (3.10) involves no domain integrals. The source term  $b$  in (3.1) has been substituted by equivalent boundary integrals. This was done by first approximating  $b$  using equation (3.6), and then expressing both right and left-hand sides of the resulting expression as boundary integrals using a weighted residual technique. The same result may be achieved using Green's second identity or a reciprocity principle. It is this operation which gives the name to the method: reciprocity has been applied to both sides of (3.8) to take all terms to the boundary, hence Dual Reciprocity Method.

The next step, as explained in chapter 2, is to write equation (3.10) in discretized form, with summations over the boundary elements replacing the integrals. This gives for a source node  $i$  the expression

$$c_i u_i + \sum_{k=1}^N \int_{\Gamma_k} q^* u d\Gamma - \sum_{k=1}^N \int_{\Gamma_k} u^* q d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left( c_i \hat{u}_{ij} + \sum_{k=1}^N \int_{\Gamma_k} q^* \hat{u}_j d\Gamma - \sum_{k=1}^N \int_{\Gamma_k} u^* \hat{q}_j d\Gamma \right) \quad (3.12)$$

It may be noted that, since  $\hat{u}$  and  $\hat{q}$  are known functions once  $f$  is defined, there is no need to approximate their variation within each boundary element by using interpolation functions and nodal values as done for  $u$  and  $q$ . However, to do so implies that the same matrices  $H$  and  $G$  defined in chapter 2 may be used on both sides of the equation. This procedure introduces an approximation in the evaluation of the terms on the right-hand side of equation (3.12); however, the error has been shown to be small and the efficiency of the method to be considerably increased.

After introducing the interpolation functions and integrating over each boundary element, the above equation can be written in terms of nodal values as

$$c_i u_i + \sum_{k=1}^N H_{ik} u_k - \sum_{k=1}^N G_{ik} q_k = \sum_{j=1}^{N+L} \alpha_j \left( c_i \hat{u}_{ij} + \sum_{k=1}^N H_{ik} \hat{u}_{kj} - \sum_{k=1}^N G_{ik} \hat{q}_{kj} \right) \quad (3.13)$$

The index  $k$  is used for the boundary nodes which are the field points. After application to all boundary nodes using a collocation technique, equation (3.13) can be expressed in matrix form as

$$Hu - Gq = \sum_{j=1}^{N+L} \alpha_j (H\hat{u}_j - G\hat{q}_j) \quad (3.14)$$

In equation (3.14), the terms  $c_i$  have been incorporated onto the principal diagonal of  $H$ . Note that for internal nodes a different procedure will be adopted (see section 3.3.1).

If each of the vectors  $\hat{u}_j$  and  $\hat{q}_j$  is considered to be one column of the matrices  $\hat{U}$  and  $\hat{Q}$  respectively, then equation (3.14) may be written without the summation to produce

$$Hu - Gq = (H\hat{U} - G\hat{Q})\alpha \quad (3.15)$$

Equation (3.15) is the basis for the application of the Boundary Element Dual Reciprocity Method and involves discretization of the boundary only. Internal nodes may be defined in the number and at the locations desired by the user; this is generally done at points where it is desirable to know the interior solution. If the exact locations of the interior nodes are not important another possibility is for the computer code to choose them according to the variation of the source terms.

## Interior Nodes

The definition of interior nodes is not normally a necessary condition to obtain a boundary solution; however, the solution will usually be more accurate if a number of such nodes is used. Comparisons of results for different numbers of interior nodes are presented for the Poisson equation in section 3.4.5, for the Helmholtz equation in chapter 4 and for a time-dependent diffusion problem in chapter 5.

An obvious situation where interior nodes are necessary in order to obtain a solution arises if a homogeneous boundary condition  $u = 0$  is applied at all boundary nodes.

When interior nodes are defined, each one is independently placed, and they do not form part of any element or cell, thus only the coordinates are needed as input data. Hence, these nodes may be defined in any order.

## The $\alpha$ Vector

The  $\alpha$  vector in equation (3.15) will now be considered. It was seen in equation (3.3) that

$$b = \sum_{j=1}^{N+L} f_j \alpha_j \quad (3.16)$$

By taking the value of  $b$  at  $(N + L)$  different points, a set of equations like (3.16) is obtained; this may be expressed in matrix form as

$$\mathbf{b} = \mathbf{F}\boldsymbol{\alpha} \quad (3.17)$$

where each column of  $\mathbf{F}$  consists of a vector  $\mathbf{f}_j$  containing the values of the function  $f_j$  at the  $(N + L)$  DRM collocation points. In the case of the problems considered in this chapter, the function  $b$  in (3.1) and (3.16) is a known function of position. Thus (3.17) may be inverted to obtain  $\boldsymbol{\alpha}$ , i.e.

$$\boldsymbol{\alpha} = \mathbf{F}^{-1}\mathbf{b} \quad (3.18)$$

The right-hand side of equation (3.15) is thus a known vector. Writing (3.15) as

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{d} \quad (3.19)$$

where

$$\mathbf{d} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\boldsymbol{\alpha} \quad (3.20)$$

it is seen that vector  $\mathbf{d}$  can be obtained directly by multiplying known matrices and vectors. Equation (3.20) may be compared with (2.78) which obtains the same result by integrating over internal cells.

Applying boundary conditions to (3.19) as explained in chapter 2 ( $N$  values of  $u$  and  $q$  are known on  $\Gamma$ ), this equation reduces to the form

$$\mathbf{Ax} = \mathbf{y} \quad (3.21)$$

where  $\mathbf{x}$  contains  $N$  unknown boundary values of  $u$  or  $q$ . A full discussion of the implementation of different boundary conditions in BEM analysis can be found in [2].

### Interior Solution

After the solution of (3.21) is obtained using standard techniques, the values at any internal node can be calculated from equation (3.13), each one involving a separate multiplication of known vectors and matrices. In the case of internal nodes, as was explained in chapter 2,  $c_i = 1$  and equation (3.13) becomes

$$u_i = - \sum_{k=1}^N H_{ik} u_k + \sum_{k=1}^N G_{ik} q_k + \sum_{j=1}^{N+L} \alpha_j \left( \hat{u}_{ij} + \sum_{k=1}^N H_{ik} \hat{u}_{kj} - \sum_{k=1}^N G_{ik} \hat{q}_{kj} \right) \quad (3.22)$$

The development of DRM for Poisson-type equations is now complete. In the next section, the different approximating functions  $f$  and the respective expressions  $\hat{u}$  and  $\hat{q}$  will be considered and in section 3.3 the computer implementation will be discussed.

## 3.2 Different $f$ Expansions

The particular solution  $\hat{u}$ , its normal derivative  $\hat{q}$  and the corresponding approximating functions  $f$  used in DRM analysis are not limited by the formulation except that the resulting  $\mathbf{F}$  matrix, equation (3.17), should be non-singular.

In order to define these functions it is customary to propose an expansion for  $f$  and then compute  $\hat{u}$  and  $\hat{q}$  using equations (3.4) and (3.11), respectively.

Nardini and Brebbia, the originators of the method, have proposed the following types of functions for  $f$  [6,7]:

1. Elements of the Pascal triangle;
2. Trigonometric series;
3. The distance function  $r$  used in the definition of the fundamental solution.

Many other types of function may be proposed. The  $r$  function was adopted first by Nardini and Brebbia and then by most researchers as the simplest and most accurate alternative.

If  $f = r$ , then it can easily be shown that the corresponding  $\hat{u}$  function is  $r^3/9$ , in the two-dimensional case, remembering the definition of  $r$

$$r^2 = r_x^2 + r_y^2 \quad (3.23)$$

where  $r_x$  and  $r_y$  are the components of  $r$  in the direction of the  $x$  and  $y$  axes.

The function  $\hat{q}$  will be given by

$$\hat{q} = \frac{r}{3}[r_x \cos(n, x) + r_y \cos(n, y)] \quad (3.24)$$

In the above, the direction cosines refer to the outward normal at the boundary with respect to the  $x$  and  $y$  axes. Formula (3.24) may be easily obtained using (3.11) and remembering that  $\partial r / \partial x = r_x / r$  and  $\partial r / \partial y = r_y / r$ .

Recent work [8-14] suggests that  $f = r$  is in fact one component of the series

$$f = 1 + r + r^2 + \dots + r^m \quad (3.25)$$

The  $\hat{u}$  and  $\hat{q}$  functions corresponding to (3.25) are:

$$\hat{u} = \frac{r^2}{4} + \frac{r^3}{9} + \dots + \frac{r^{m+2}}{(m+2)^2} \quad (3.26)$$

$$\hat{q} = \left( r_x \frac{\partial x}{\partial n} + r_y \frac{\partial y}{\partial n} \right) \left( \frac{1}{2} + \frac{r}{3} + \dots + \frac{r^m}{(m+2)} \right) \quad (3.27)$$

In principle, any combination of terms may be selected from (3.25). To illustrate this, for Poisson-type equations, three cases will be considered:

1.  $f = r$
2.  $f = 1 + r$
3.  $f = 1$  at one node and  $f = r$  at remaining nodes

Case (2) is generally recommended, and used in the program of section 3.4. Alternatives (1) and (3) require some modifications, but produce similar results. Two additional cases,

4.  $f = 1 + r + r^2$
5.  $f = 1 + r + r^2 + r^3$

are considered in the next chapter for convective and non-linear problems where the governing equations are more complex. In all cases, however, results are found to differ little from those obtained using  $1 + r$  which is, as will be explained, the simplest alternative.

### 3.2.1 Case $f = r$

In this case the matrix  $\mathbf{F}$  will contain zeros on the leading diagonal, but the matrix is non-singular if no double nodes are used (if two nodes have the same coordinates two rows and columns of  $\mathbf{F}$  will be identical).

The solver used to obtain  $\boldsymbol{\alpha}$  will have to be capable of row and column interchange. Once  $\boldsymbol{\alpha}$  is accurately determined good results may be expected. Results for the case of a Poisson equation analysed using  $f = r$  are given in table 3.1.

### 3.2.2 Case $f = 1 + r$

The presence of the constant guarantees the “completeness” of the expansion, and also implies that the leading diagonal of  $\mathbf{F}$  is no longer zero. Equation (3.17) may be solved for  $\boldsymbol{\alpha}$  using standard Gauss elimination. This is the simplest alternative to program, and that which has been adopted in Program 2, section 3.4. It has already produced excellent results for a wide range of engineering problems as will be seen in this and in subsequent chapters. Results for a Poisson example analysed using  $f = 1 + r$  are given in table 3.1.

Note that in this case  $\hat{u} = r^2/4 + r^3/9$  and  $\hat{q} = (r_x \partial x / \partial n + r_y \partial y / \partial n)(1/2 + r/3)$ .

### 3.2.3 Case $f = 1$ at One Node and $f = r$ at Remaining Nodes

Another set of approximating functions which has been suggested is a combination of  $f = r$  and  $f = 1$ . The idea of including a function  $f = \text{constant}$  is to simulate better the effects of a constant source. Note, however, that  $f = 1$  cannot be used at more than one node as otherwise matrix  $\mathbf{F}$  becomes singular.

Using the notation  $f_{\ell j}$ , where both indices refer to DRM collocation points, let the value of  $j$  at which  $f = 1$  be  $k$ . At all other values of  $j$ ,  $f = r$ . In this case  $f$  can be expressed by the formula

$$f_{\ell j} = \delta_{jk} + (1 - \delta_{jk})r_{\ell j} \quad (3.28)$$

where  $\delta$  is the Kronecker delta and  $k$  has a fixed value equal to the number of the node at which the constant is applied, usually the central node of the problem. In the case of the problem shown in figure 3.5, for example, the central node is numbered 31, so  $k$  in equation (3.28) must have the value 31.

So, for the Poisson equation  $\nabla^2 u = -2$ , equation (3.17) will now have the form:

$$\begin{array}{c|c|c} & j=k & \\ \begin{matrix} r_{\ell j} & \dots & 1 & r_{\ell j} & \dots \\ \vdots & & 1 & \vdots & \\ & & \vdots & & \end{matrix} & \alpha = & \begin{matrix} -2 \\ -2 \\ \vdots \end{matrix} \end{array} \quad (3.29)$$

Solving this system produces a vector  $\boldsymbol{\alpha}$  which consists entirely of zeros except at the position  $k$  where  $\alpha_k = -2$ .

The above is equivalent to an expansion of  $b$ , equation (3.3), in the form

$$b = \sum_{j=1}^{N+L} \alpha_j f_j = \alpha_k = -2 \quad (3.30)$$

that is, there is no summation involved and no approximation for the case of the constant source term; in fact, for a constant  $b$ ,

$$\int_{\Omega} bu^* d\Omega = b \int_{\Omega} u^* d\Omega = b \int_{\Omega} (\nabla^2 \hat{u}) u^* d\Omega \quad (3.31)$$

since  $\nabla^2 \hat{u} = 1$  for  $\hat{u} = r^2/4$ . A reciprocity principle can now be applied to the last term in the above; thus, in this case the Dual Reciprocity Method is equivalent to the technique of particular solutions described in section 2.3.4.

### 3.3 Computer Implementation

The computer implementation of the DRM is explained in this section. First, equations (3.15) and (3.22) are written in a schematized form, and then another schematized equation is written which includes them both. Finally, the sign of the components of the  $r$  function and its derivatives are discussed.

#### 3.3.1 Schematized Matrix Equations

Equation (3.15), from which the solution at boundary nodes is obtained, can be written as follows:

$$\begin{matrix} N \times N & | \\ \boxed{\mathbf{H}} & \boxed{u} \end{matrix} - \begin{matrix} N \times N & | \\ \boxed{\mathbf{G}} & \boxed{q} \end{matrix} = \begin{matrix} | \\ N \end{matrix} \quad (3.32)$$

$$\left[ \begin{matrix} N \times N & | \\ \boxed{\mathbf{H}} & \boxed{\hat{U}} \end{matrix} - \begin{matrix} N \times N & | \\ \boxed{\mathbf{G}} & \boxed{\hat{Q}} \end{matrix} \right] \begin{matrix} | \\ N+L \\ \alpha \end{matrix}$$

Notice that there is a known vector of size  $N$  on the right-hand side of (3.32), called  $\mathbf{d}$  in equation (3.19). After applying boundary conditions equation (3.21) is obtained. The coefficients of matrices  $\mathbf{H}$  and  $\mathbf{G}$  in the above equation are calculated in the usual way as shown in chapter 2. These matrices are the same on both sides of the equation.

Each coefficient of the matrices  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  is a function of the distance between two nodes. Calling the coefficients of  $\hat{\mathbf{U}}$  by  $\hat{u}_{kj}$ , and those of  $\hat{\mathbf{Q}}$  by  $\hat{q}_{kj}$ , then the  $k$

points are the boundary nodes, and the  $j$  points refer to all nodes, boundary and internal. Thus, there are  $N k$  points and  $(N + L) j$  points, generating matrices of size  $N \times (N + L)$ .

At this stage it is important to remember that, to facilitate the treatment of points of geometric discontinuity, Program 1 of chapter 2 considered matrix  $\mathbf{G}$  in unassembled form, i.e. as a rectangular matrix of dimensions  $N \times 2N$  (see section 2.2.6). Since the same program structure will be used for the DRM implementation, matrix  $\hat{\mathbf{Q}}$  will also have to be considered in unassembled form. Thus,  $\hat{\mathbf{Q}}$  will in fact have the dimensions  $2N \times (N + L)$ , with two values of  $\hat{q}_{kj}$  for each boundary node  $k$ , obtained by applying expression (3.11) for the two elements joining at that node.

The discontinuity of the outward normal at boundary nodes can be seen in figure 3.2.

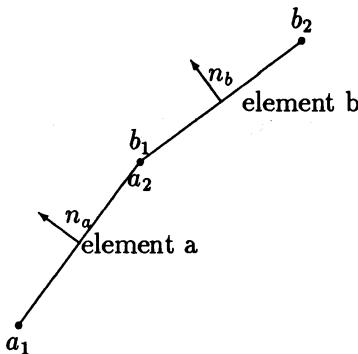


Figure 3.2: Junction of Two Boundary Elements

In order to obtain the vector  $\alpha$ , matrix  $\mathbf{F}$  is constructed of size  $(N + L) \times (N + L)$ . Each of its terms  $f_{\ell j}$  is a function of the distance between points  $\ell$  and  $j$ , both of which take all the values from 1 to  $(N + L)$ .  $\mathbf{F}$  is thus a symmetric matrix. Vector  $\mathbf{b}$  is also constructed, its coefficients being the nodal values of function  $b$ . For the present,  $b$  is assumed to be a known function, thus  $\alpha$  can be found through equation (3.17) using standard Gauss elimination.

Having obtained the values of the unknowns on the boundary, the next step is to calculate values at internal nodes. For this, equation (3.22) is used in the following schematized form

$$\begin{aligned}
 & \boxed{L \times L} \quad \boxed{L} = \boxed{L \times N} \quad \boxed{N} - \boxed{L \times N} \quad \boxed{N} \\
 & \mathbf{I} \quad \mathbf{u} \qquad \mathbf{G} \qquad \mathbf{q} \qquad \mathbf{H} \qquad \mathbf{u} \\
 & + \left[ \begin{array}{c|c} \boxed{L \times N} & \boxed{N \times (N+L)} \\ \mathbf{H} & \end{array} \right] - \left[ \begin{array}{c|c} \boxed{L \times N} & \boxed{N \times (N+L)} \\ \mathbf{G} & \end{array} \right] \boxed{N+L} \\
 & \qquad \hat{\mathbf{U}} \qquad \hat{\mathbf{Q}} \qquad \alpha \\
 & + \boxed{L \times L} \quad \boxed{L \times (N+L)} \quad \boxed{N+L} \\
 & \mathbf{I} \qquad \hat{\mathbf{U}} \qquad \alpha \\
 & \qquad \qquad \qquad \alpha
 \end{aligned} \tag{3.33}$$

Equation (3.33) will be considered, term by term. On the left-hand side an identity matrix of size  $L \times L$  multiplies the  $L$  values of  $u$  at internal nodes. This is due to the  $c_i$  terms, which are all unity at internal nodes. The same matrix appears in the last term on the right-hand side. The  $\mathbf{H}$  and  $\mathbf{G}$  matrices partitions of size  $L \times N$  which appear on both sides of (3.33) are produced by integrating over the boundary from each internal node and are not, therefore, the same partitions of  $\mathbf{H}$  and  $\mathbf{G}$  given in (3.32). In the first and second terms on the right-hand side,  $\mathbf{G}$  and  $\mathbf{H}$  multiply the now known boundary values of  $q$  and  $u$ , respectively.

The matrices  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  of size  $N \times (N+L)$  which appear in the third and fourth terms are exactly the same as given in (3.32).

The last term in (3.33) is extremely important. It is, in a sense, an “extra” term since it does not appear in (3.32), having been incorporated onto the main diagonal of  $\mathbf{H}$ . It is generated by the term  $\alpha_j \hat{u}_{ij}$  of equation (3.22). Its contribution to the internal solution is important, and should on no account be omitted. The  $\hat{\mathbf{U}}$  matrix in this term is different from that of the third term. Considering a generic coefficient to be  $\hat{u}_{ij}$ , then  $i$  will have the range 1,  $L$  and  $j$  will have the range 1,  $(N+L)$ . Vector  $\alpha$  is as described before for equation (3.32).

The relationship between equations (3.32) and (3.33) can be easily seen if a third schematic representation is drawn in which both are shown in one global scheme, now valid for both internal and boundary nodes. This is done as follows:

$$\begin{array}{c|c} BS & 0 \\ \hline N \times N & N \times L \end{array} \quad - \quad \begin{array}{c|c} BS & 0 \\ \hline IS & 0 \end{array} = \begin{array}{c|c} BS & 0 \\ \hline IS & 0 \end{array} \quad (3.34)$$

$$\left[ \begin{array}{c} \boxed{\begin{array}{|c|c|} \hline BS & 0 \\ \hline N \times N & N \times L \\ \hline \end{array}} & \boxed{\begin{array}{|c|} \hline (BS + IS) \\ \hline N \times (N + L) \\ \hline \end{array}} & - & \boxed{\begin{array}{|c|c|} \hline BS & 0 \\ \hline IS & 0 \\ \hline \end{array}} & \boxed{\begin{array}{|c|} \hline (BS + IS) \\ \hline 0 \\ \hline \end{array}} & \boxed{\begin{array}{|c|} \hline BS \\ \hline + \\ \hline IS \\ \hline \end{array}} \end{array} \right] \alpha$$

In the above equation, matrix partitions marked  $BS$  are used exclusively in the Boundary Solution, while those marked  $IS$  are used exclusively in the Internal nodes Solution; the ones marked  $(BS+IS)$  are used in both. Empty partitions are indicated as 0 and the identity matrices which appear in equation (3.33) are marked  $I$ .

This representation with all the matrices in an  $(N + L) \times (N + L)$  form simplifies the understanding of the method and will be used in the next chapter for problems for which  $b$  is an unknown function.

Given the above, both equations (3.32) and (3.33) may be represented together by the matrix equation

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\boldsymbol{\alpha} \quad (3.35)$$

It can be seen from the schematized equations that it is necessary to use the  $(N + L)$  DRM collocation points in order to obtain improved results both at boundary and interior nodes. Although the solutions are uncoupled, the number and position of internal nodes will influence the boundary solution through the term  $b$ , according to equation (3.3).

A computer program written for Poisson-type equations will be presented in section 3.4. In examining it, and the subsequent computer programs in chapters 4 and 5, the above schematic representation should be borne in mind.

### 3.3.2 Sign of the Components of $r$ and its Derivatives

The distance function  $r$  used in two-dimensional BEM analysis is defined as  $r^2 = r_x^2 + r_y^2$ , where  $r_x$  and  $r_y$  are the projections of  $r$  on the  $x$  and  $y$  axes, respectively.

The derivatives of  $r$  with respect to  $x$  and  $y$  are given by  $r_x/r$  and  $r_y/r$ . Similar expressions hold in three dimensions.

In calculating the coefficients of each row of the  $\mathbf{H}$  and  $\mathbf{G}$  matrices a fixed point  $i$  (source point) is defined and integration is carried out over the boundary. Thus,  $r$  may be considered as the modulus of a vector  $\vec{r}_{ik}$  where  $i$  is a fixed point and  $k$  variable along the boundary.

In calculating the coefficients of the DRM matrices  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$ , it has been shown in equation (3.13) and in the previous schematic representations that the indices are different from those used in the case of the matrices  $\mathbf{H}$  and  $\mathbf{G}$ . For the DRM matrices  $r$  may be considered as the modulus of a vector  $\vec{r}_{kj}$  where, for each boundary point  $k$ ,  $j$  represents each of the other nodes, boundary and internal. To calculate a given line of  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$ , point  $k$  is thus fixed and  $j$  is considered to vary, as shown in figure 3.3.

The  $i$  points are source points, the  $k$  points are boundary nodes and the  $j$  points are DRM collocation points. The BEM matrices  $\mathbf{H}$  and  $\mathbf{G}$  use the source points  $i$ , while the DRM matrices, which are geometric relationships, do not.

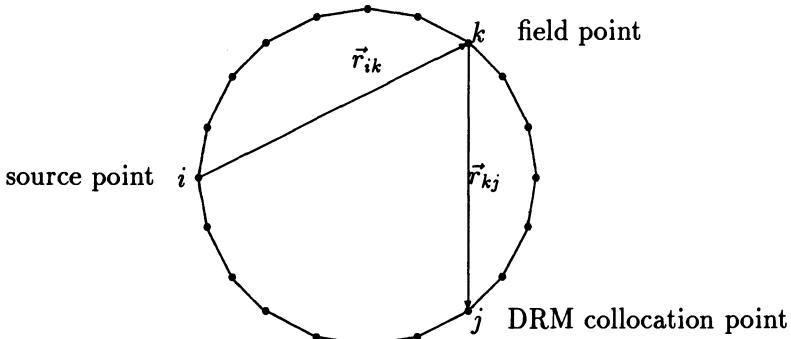


Figure 3.3: Vectors  $\vec{r}_{ik}$  and  $\vec{r}_{kj}$

Thus

$$r_{ik}^2 = (x_k - x_i)^2 + (y_k - y_i)^2$$

$$r_{kj}^2 = (x_j - x_k)^2 + (y_j - y_k)^2$$

Rows and columns of matrices  $\mathbf{H}$  and  $\mathbf{G}$  are given by  $i$  and  $k$  respectively, while that of  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  are given by  $k$  and  $j$ , respectively. Letting  $j = i$  it can be seen that  $r_x$  and  $r_y$  change signs in the second case.

The above is not important in the calculation of  $\hat{\mathbf{U}}$  as this matrix does not contain derivatives; however, if the correct sign of  $r_x$  and  $r_y$  is not used, the transpose of the  $\hat{\mathbf{Q}}$  matrix will be obtained, resulting in an erroneous solution.

The DRM matrix  $\mathbf{F}$  is also generated considering DRM collocation points as it also represents geometric relationships and does not include the source nodes  $i$ . The change of sign referred to above also occurs when calculating its derivatives, which will be needed in section 4.2.

### 3.4 Computer Program 2

In this section a computer program will be given for the solution of Poisson-type equations by the DRM. The program adopts the expansion  $f = 1+r$  and is specialized for the case of function  $b$  in equation (3.1) being a known function of position. Either a single function or a summation of known functions (*e.g.*  $b = 1 + x^2 + \sin x$ ) may be employed. The type of function to be used must be supplied by the user and incorporated into the subroutine ALFAF2 in the position indicated in section 3.4.3.

If  $b$  is a function of the problem variable, Program 3 (chapter 4) should be used; if  $b$  is a time-dependent function, Program 4 (chapter 5) must be employed.

Linear elements are used for simplicity, although any element type can be used with the DRM. Another type of element would involve the complete revision of routine ASSEM2 and some small changes to INPUT2, INTERM and RHSVEC. If curved elements are to be used special integration, as described in Appendix 2, must be employed.

The computer program to be presented is in modular form, using some of the subroutines which have been given in Program 1 (chapter 2), which solves the Poisson equation using internal cells. Again the program is dimensioned for up to 200 nodes, and its structure is shown in figure 3.4.

The program modules:

- ASSEM2
- SOLVER
- INTERM
- OUTPUT

were described and listed in chapter 2 and will be used in the form already given. In this section the following new routines will be described and listed:

- MAINP2
- INPUT2
- ALFAF2
- RHSVEC

To construct Program 2 starting from Program 1, the following operations are carried out:

1. Substitution of MAINP2 for MAINP1
2. Substitution of INPUT2 for INPUT1
3. Substitution of RHSVEC for NECMOD
4. Inclusion of ALFAF2

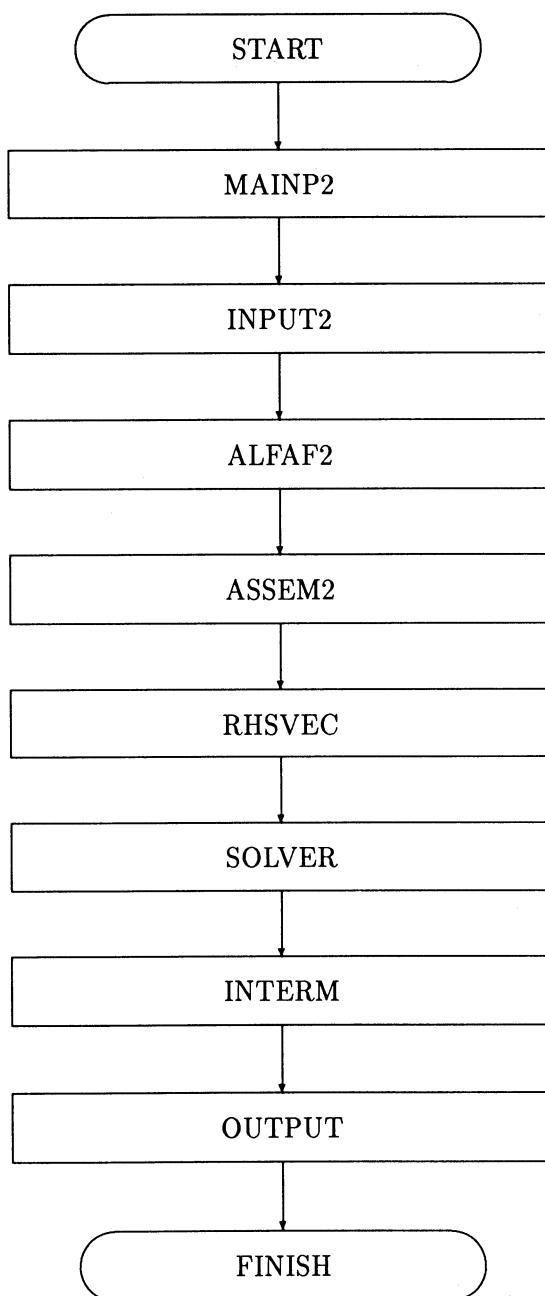


Figure 3.4: Modular structure for Program 2

The variable names used in the program are the same as those used in Program 1. Where new variables are introduced, these will have symbols as close as possible to those used in the text. At the end of the section, results and data for a test problem are given.

### 3.4.1 MAINP2

The structure of MAINP2 is very similar to that of MAINP1. Data is read with a new routine, INPUT2, from which all reference to cells has been purged.

Next, vector  $\alpha$  is calculated based on the known values of the  $b(x, y)$  function, from the following relationship

$$\alpha = F^{-1}b$$

and is explained in section 3.4.3 on subroutine ALFAF2.

The remainder of the routine follows MAINP1 except that the DRM routine RHSVEC is now called to calculate the  $d$  vector, i.e.

$$d = (\hat{H}U - \hat{G}\hat{Q})\alpha$$

instead of the cell routine NECMOD.

```

COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/XY(200),A(200,200)
REAL LE,LJ
INTEGER CON
C
C  MASTER ROUTINE2
C
C DATA INPUT
C
      CALL INPUT2
C
C CALCULATION OF ALPHA VECTOR
C
      CALL ALFAF2
C
C CALCULATION OF G AND H MATRICES: APPLY BOUNDARY CONDITIONS
C RESULTING MATRIX IN A: RESULTING VECTOR IN XY
C
      CALL ASSEM2
C
C CALCULATION OF ADDITIONAL VECTOR D
C
      CALL RHSVEC
C
C SUM XY AND D
C
      DO 5 I = 1,NN

```

```

XY(I)=XY(I)+D(I)
5      CONTINUE
C
C SOLVE FOR BOUNDARY VALUES
C
CALL SOLVER
C
C PUT BOUNDARY VALUES IN APPROPRIATE ARRAY U OR Q
C
DO 76 I=1,NN
KK=KODE(I)
IF(KK.EQ.0.OR.KK.EQ.2)THEN
U(I) = XY(I)
ELSE
Q(I) = XY(I)
END IF
76      CONTINUE
C
C CALCULATE VALUES AT INTERNAL NODES
C
CALL INTERM
C
C WRITE RESULTS
C
CALL OUTPUT
STOP
END

```

### 3.4.2 Subroutine INPUT2

Comparing INPUT2 and INPUT1 the reader can already see some of the advantages of the DRM. Subroutine INPUT2 is much shorter since integration points, weight factors, connectivity and other data relating to cells have been eliminated. The listing of INPUT2 is as follows:

```

SUBROUTINE INPUT2
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200),NI
COMMON/FIVE/XY(200),A(200,200)
COMMON/CELL/NCI,MKJ(200,3),WW(7),CP(7,3),DA(200),CONST,NPI
C
C COORDINATES OF NUMERICAL INTEGRATION POINTS FOR BOUNDARY ELEMENTS IN POEI
C WEIGHT FACTORS FOR BOUNDARY ELEMENTS IN FDEP. 4-POINT GAUSS INTEGRATION USED.
C
DATA POEI/0.86113631,-0.86113631,0.33998104,-0.33998104/
DATA FDEP/0.34785485,0.34785485,0.65214515,0.65214515/
REAL LE,LJ
INTEGER CON
C
C NUMBER OF INTEGRATION POINTS PER BOUNDARY ELEMENT IN NI
C
NI=4

```

```

C
C NN  NUMBER OF BOUNDARY NODES
C NE  NUMBER OF BOUNDARY ELEMENTS
C L   NUMBER OF INTERNAL NODES
C
      READ(5,1234) NN,NE,L
1234  FORMAT(3I4)
C
C CON CONNECTIVITY OF BOUNDARY ELEMENTS
C
      DO 1007 I = 1,NN
      CON(I,1) = I
      CON(I,2) = I + 1
1007  CONTINUE
      CON(NN,2) = 1
C
C INITIALIZE VECTORS FOR INTERNAL NODES
C
      DO 1008 I = NN+1,NN+L
      KODE(I) = 0
      U(I) = 0.
      Q(I) = 0.
1008  CONTINUE
C
C READ COORDINATES OF NODES
C
      DO 2 I=1,NN+L
      READ(5,3) X(I),Y(I)
3      FORMAT(2F13.6)
2      CONTINUE
C
C READ TYPE AND VALUE OF BOUNDARY CONDITION
C VAL CONTAINS THE VALUE OF THE BOUNDARY CONDITION
C
C VALUES OF KODE(I)
C KODE(I) = 1 U KNOWN AT POINT I
C KODE(I) = 2 Q KNOWN AT POINT I
C
      DO 4 I=1,NN
      READ(5,5) KODE(I),VAL
5      FORMAT(I3,F11.5)
      IF(KODE(I).EQ.1) U(I)=VAL
      IF(KODE(I).EQ.2) Q(I) = VAL
4      CONTINUE
C
C PRINT INPUT DATA
C
      WRITE(6,8) NN
8      FORMAT(' NUMBER OF BOUNDARY NODES = ',I3/)
      WRITE(6,9) NE
9      FORMAT(' NUMBER OF BOUNDARY ELEMENTS = ',I3/)
      WRITE(6,10) L
10     FORMAT(' NUMBER OF INTERNAL NODES = ',I3/)

```

```

        WRITE(6,11)
11      FORMAT(' DATA FOR BOUNDARY NODES')
        WRITE(6,12)
12      FORMAT(' NODE   X           Y     TYPE U     Q   ')
        WRITE(6,1012) (I,X(I),Y(I),KODE(I),U(I),Q(I),I=1,NN)
1012    FORMAT(I3,2F10.6,I3,2F6.3)
        WRITE(6,1187)
1187    FORMAT(' COORDINATES OF INTERNAL NODES')
        WRITE(6,1288)
1288    FORMAT(' NODE   X           Y   ')
        WRITE(6,1112) (I,X(I),Y(I),I=NN+1,NN+L)
1112    FORMAT(I3,2F10.6)
        WRITE(6,6)
6       FORMAT(//' ELEMENT DATA'//)
        WRITE(6,7)
7       FORMAT('      NO   NODE 1 NODE 2      LENGTH')
C
C CALCULATE ELEMENT LENGTH IN LE
C
        DO 33 K = 1,NE
        LE(K) = SQRT((X(CON(K,2))-X(CON(K,1)))**2 +
1      (Y(CON(K,2)) - Y(CON(K,1)))**2)
        WRITE(6,48) K,CON(K,1),CON(K,2),LE(K)
48      FORMAT(3I6,F16.6)
33      CONTINUE
C
        RETURN
END

```

### 3.4.3 Subroutine ALFAF2

This is a very simple routine. The user defines the type of  $b$  function to be used at the line marked by the comment statement. The  $F$  matrix is then constructed, and  $\alpha$  is obtained by Gauss elimination using the routine SOLVER. Vector  $\alpha$  is stored in array ALPHA.

```

SUBROUTINE ALFAF2
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/B(200),F(200,200)
REAL LE,LJ
INTEGER CON
C
NH = NN
NN = NN + L
DO 5 I = 1,NN
XI = X(I)
YI = Y(I)
C PUT THE PROBLEM SOURCE TERM ON NEXT LINE
C B=B(X,Y) ONLY
C

```

```

B(I) =-2.
C
      DO 2 J = 1,NN
      XJ = X(J)
      YJ = Y(J)
      R = SQRT((XI-XJ)**2 + (YI-YJ)**2)
      F(I,J) = R+1
2       CONTINUE
5       CONTINUE
      CALL SOLVER
      DO 3 I = 1,NN
C
C THE ALPHA VECTOR IS IN ALPHA
C
      ALPHA(I) = B(I)
      B(I) = 0.
      DO 3 J = 1,NN
      F(I,J) = 0.
3       CONTINUE
      NN = NH
      RETURN
      END

```

### 3.4.4 Subroutine RHSVEC

This DRM routine evaluates the vector  $\mathbf{d} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\boldsymbol{\alpha}$ . Matrices  $\mathbf{G}$  and  $\mathbf{H}$  are already available, since they were stored when subroutine ASSEM2 was called. ASSEM2 is listed in section 2.4.

The routine has five stages:

1. Calculate  $\hat{\mathbf{Q}}\boldsymbol{\alpha}$ .

The result is a vector which is stored in  $\mathbf{QH}$ . The coefficients of  $\hat{q}$  are stored in  $\mathbf{QH1}$  and  $\mathbf{QH2}$  for the nodes at the beginning and end of each element, in order to take into account the possible discontinuity in the outward normal at boundary nodes as discussed in section 3.3.

2. Calculate  $\mathbf{G}\hat{\mathbf{Q}}\boldsymbol{\alpha}$ .

The resulting vector is stored in  $\mathbf{D}$ . The multiplication is carried out taking into consideration that  $\mathbf{G}$  has been stored unassembled.

3. Calculate  $\hat{\mathbf{U}}\boldsymbol{\alpha}$ .

The resulting vector is stored in  $\mathbf{UH}$ . The value of  $\hat{u}$  for a given node is put in  $\mathbf{UH1}$ . This process is more straightforward than stage (1) as there is no discontinuity in  $\hat{u}$  at nodes, so no reference need be made to elements in the calculation.

4. Calculate  $\mathbf{H}\hat{\mathbf{U}}\boldsymbol{\alpha}$ .

Here the result for  $\hat{\mathbf{U}}\boldsymbol{\alpha}$  is multiplied by  $\mathbf{H}$ .

5. Calculate  $c_i \hat{u}_{ij} \alpha_j$ .

The contribution of these terms needs to be taken into account for interior nodes.

Note that as  $c_i = 1$  this process is equivalent to adding  $\hat{\mathbf{U}}\alpha$  to D for the interior nodes as shown in the schematic representations given in equations 3.32 - 3.34.

The listing of RHSVEC is as follows:

```

SUBROUTINE RHSVEC
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/DRM/HH(200,200),GG(200,400)
DIMENSION UH(200),QH(200)
REAL LE,LEK
INTEGER CON
C
C ROUTINE USES F=1+R
C
C D IS THE RHS VECTOR CALCULATED IN THIS ROUTINE
C INITIALIZE ARRAY D
C
      DO 1 I=1,NN+L
1      D(I) = 0.
C
      DO 59 K = 1,2*NE
59      QH(K) = 0.
      DO 60 K = 1,NE
60      N1 = CON(K,1)
      N2 = CON(K,2)
      X1 = X(N1)
      X2 = X(N2)
      Y1 = Y(N1)
      Y2 = Y(N2)
      LEK = LE(K)
      K1 = 2*K - 1
      K2 = 2*K
      DO 50 J = 1,NN+L
C
C ALPHA VECTOR
C
      ALPHAJ = ALPHA(J)
C
C QHAT CALCULATED AT BEGINNING AND END OF EACH ELEMENT
C
      XP = X(J)
      YP = Y(J)
      R1 = SQRT((X1-XP)**2 + (Y1-YP)**2)
      R2 = SQRT((X2-XP)**2 + (Y2-YP)**2)
      QH1 = (0.5+R1/3.)*( (X1-XP)*(Y1-Y2)+(Y1-YP)*(X2-X1))/LEK
      QH2 = (0.5+R2/3.)*( (X2-XP)*(Y1-Y2)+(Y2-YP)*(X2-X1))/LEK
C
C MULTIPLY QHAT BY ALPHA; RESULT IN QH
C
      QH(K1) = QH(K1) + QH1*ALPHAJ

```

```

        QH(K2) = QH(K2) + QH2*ALPHAJ
50      CONTINUE
60      CONTINUE
C
C PUT RESULT IN D
C
DO 70 I = 1,NN+L
    DO 80 K = 1,2*NE
        D(I) = D(I) - GG(I,K)*QH(K)
80      CONTINUE
70      CONTINUE
DO 159 K=1,NN
    UH(K)=0.
DO 160 K=1,NN
    XK=X(K)
    YK=Y(K)
    DO 150 J=1,NN+L
        ALPHAJ = ALPHA(J)
        XJ = X(J)
        YJ = Y(J)
        R = SQRT((XK-XJ)**2+(YK-YJ)**2)
        UH1 = (R**3)/9. + (R**2)/4.

C
C MULTIPLY UHAT BY ALPHA; RESULT IN UH
C
        UH(K) = UH(K) + UH1*ALPHAJ
150      CONTINUE
160      CONTINUE
C
C ADD RESULT TO D
C
DO 170 I=1,NN+L
    DO 180 K=1,NN
        D(I) = D(I) + HH(I,K)*UH(K)
180      CONTINUE
170      CONTINUE
C
C CONTRIBUTION OF CI*UHAT*ALPHA TERMS
C
DO 1160 I=NN+1,NN+L
    XI=X(I)
    YI=Y(I)
    DO 1150 J=1,NN+L
        ALPHAJ = ALPHA(J)
        XP = X(J)
        YP = Y(J)
        R = SQRT((XI-XP)**2+(YI-YP)**2)
        UH1 = (R**3)/9. + (R**2)/4.
        D(I) = D(I) + UH1*ALPHAJ
1150      CONTINUE
1160      CONTINUE
RETURN
END

```

### 3.4.5 Comparison of Results for a Torsion Problem using Different Approximating Functions

Results will be presented for the problem of torsion of an elliptical section [15] already discussed in chapter 2.

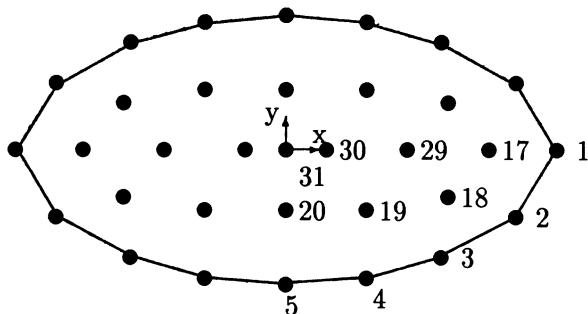


Figure 3.5: Ellipse Problem Discretization

## Formulation

The problem of Saint-Venant torsion of a member of constant cross-section is governed by the equation

$$\frac{\partial}{\partial x} \left( \frac{1}{G} \frac{\partial v}{\partial x} \right) + \frac{\partial}{\partial y} \left( \frac{1}{G} \frac{\partial v}{\partial y} \right) = -2\theta \quad (3.36)$$

where  $G$  is the shear modulus and  $\theta$  the angle of twist. The problem variable is the stress function  $v$ , such that

$$\tau_{xz} = \frac{\partial v}{\partial y}, \quad \tau_{yz} = -\frac{\partial v}{\partial x} \quad (3.37)$$

Defining  $u = v/G\theta$  equation (3.36) can be written as

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2 \quad (3.38)$$

which is the same as equation (3.1) with  $b = -2$  and the Poisson equation studied in section 2.3.2.

The elliptical section shown in figure 3.5 has a semi-major axis  $a = 2$  and a semi-minor axis  $b = 1$ . The equation of the ellipse is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \quad (3.39)$$

Equation (3.39) not only enables the coordinates of the boundary nodes in figure 3.5 to be generated, but also appears in the exact solution which for  $G = \theta = 1$  is given by:

$$u = -0.8 \left( \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 \right) \quad (3.40)$$

Thus it can be seen that (3.40) satisfies the boundary condition  $u = 0$  on  $\Gamma$ . The equation may be verified by substitution into (3.38). The solution for  $q$  is obtained starting from

$$q = \frac{\partial u}{\partial x} \frac{\partial x}{\partial n} + \frac{\partial u}{\partial y} \frac{\partial y}{\partial n} \quad (3.41)$$

In the case of an ellipse the direction cosines of the outward normal at any point on the boundary are given by  $\partial x/\partial n = x/a$  and  $\partial y/\partial n = y/b$  such that

$$q = -0.2(x^2 + 8y^2) \quad (3.42)$$

## DRM Results

For the DRM solution 16 linear boundary elements were used and 17 internal nodes defined. Results are given in table 3.1 for all the  $f$  functions considered in sections 3.2.1-3.2.3. The results at the twelve points in the table describe the complete solution due to symmetry. Boundary results are rather poor at node 1 due to the coarse discretization there. Results obtained in section 2.3.2 using cell integration are also included for comparison.

Variable	Node	X	Y	$f = r$	$f = 1 + r$	$f = 1 \text{ or } r$	Cell	Exact
$q$	1	2.0	0.0	-0.680	-0.680	-0.682	-0.733	-0.8
	2	1.706	-0.522	-1.019	-1.020	-1.024	-1.046	-1.018
	3	1.179	-0.808	-1.357	-1.359	-1.363	-1.378	-1.322
	4	0.598	-0.954	-1.531	-1.532	-1.536	-1.549	-1.528
	5	0.0	-1.0	-1.587	-1.588	-1.592	-1.611	-1.6
$u$	17	1.5	0.0	0.349	0.349	0.350	0.331	0.350
	18	1.2	-0.35	0.418	0.418	0.418	0.401	0.414
	19	0.6	-0.45	0.574	0.573	0.573	0.557	0.566
	20	0.0	-0.45	0.646	0.646	0.646	0.629	0.638
	29	0.9	0.0	0.643	0.643	0.643	0.626	0.638
	30	0.3	0.0	0.789	0.789	0.789	0.772	0.782
	31	0.0	0.0	0.807	0.807	0.807	0.791	0.800

Table 3.1: Results for Torsion of Ellipse for Different Functions  $f$

For the linear elements employed in the discretization, results using the three different  $f$  functions are seen to be very similar, thus the use of  $f = 1 + r$  is recommended

as this is the simplest to use, requiring no special considerations. Note that the cell integration results are much less accurate. In table 3.2 DRM results are presented for the same problem with  $f = 1 + r$  considering different numbers of internal nodes.

Variable	Node	X	Y	$L = 17$	$L = 13$	$L = 9$	$L = 5$	$L = 1$	Exact
$q$	1	2.0	0.0	-0.680	-0.678	-0.677	-0.676	-0.666	-0.8
	2	1.706	-0.522	-1.020	-1.017	-1.016	-1.013	-0.995	-1.018
	3	1.179	-0.808	-1.359	-1.357	-1.354	-1.349	-1.325	-1.322
	4	0.598	-0.954	-1.532	-1.530	-1.525	-1.517	-1.499	-1.528
	5	0.0	-1.0	-1.588	-1.585	-1.580	-1.573	-1.557	-1.6
$u$	31	0.0	0.0	0.807	0.806	0.803	0.798	0.788	0.800

Table 3.2: Results for Torsion Problem for Different Values of  $L$

It can be seen that, in this case, the sensitivity of the results to the number of internal nodes is small. The total variation of results from  $L = 1$  to  $L = 17$  is approximately 2%.

### 3.4.6 Data and Output for Program 2

The data file for the torsion problem is given below. This data is less than that needed for the same problem using Program 1, section 2.4, because no cells are used.

The total data input consists of :

i) 1 line of global data, Number of Boundary Nodes, Number of Boundary Elements, Number of Internal Nodes.

Internal names NN; NE; L

FORMAT(3I4)

ii)  $NN+L$  lines to define coordinates of each node. Boundary nodes are given in clockwise order.

Internal names X; Y

FORMAT(2F13.6)

iii)  $NN$  lines to define boundary conditions. One boundary condition type and one value per line.

Internal names KODE; VAL

FORMAT(I3,F11.5)

## Data for Torsion Problem

**Output for the Torsion Problem**(These are the results for  $f = 1 + r$  presented in table 3.1)**NUMBER OF BOUNDARY NODES = 16****NUMBER OF BOUNDARY ELEMENTS = 16****NUMBER OF INTERNAL NODES = 17****DATA FOR BOUNDARY NODES**

NODE	X	Y	TYPE	U	Q
1	2.000000	0.000000	1	0.000	0.000
2	1.705706	-0.522150	1	0.000	0.000
3	1.178800	-0.807841	1	0.000	0.000
4	0.597614	-0.954310	1	0.000	0.000
5	0.000000	-1.000000	1	0.000	0.000
6	-0.597614	-0.954310	1	0.000	0.000
7	-1.178800	-0.807841	1	0.000	0.000
8	-1.705706	-0.522150	1	0.000	0.000
9	-2.000000	0.000000	1	0.000	0.000
10	-1.705706	0.522150	1	0.000	0.000
11	-1.178800	0.807841	1	0.000	0.000
12	-0.597614	0.954310	1	0.000	0.000
13	0.000000	1.000000	1	0.000	0.000
14	0.597614	0.954310	1	0.000	0.000
15	1.178800	0.807841	1	0.000	0.000
16	1.705706	0.522150	1	0.000	0.000

**COORDINATES OF INTERNAL NODES**

NODE	X	Y
17	1.500000	0.000000
18	1.200000	-0.350000
19	0.600000	-0.450000
20	0.000000	-0.450000
21	-0.600000	-0.450000
22	-1.200000	-0.350000
23	-1.500000	0.000000
24	-1.200000	0.350000
25	-0.600000	0.450000
26	0.000000	0.450000
27	0.600000	0.450000
28	1.200000	0.350000
29	0.900000	0.000000
30	0.300000	0.000000
31	0.000000	0.000000
32	-0.300000	0.000000
33	-0.900000	0.000000

## ELEMENT DATA

NO	NODE 1	NODE 2	LENGTH
1	1	2	0.599374
2	2	3	0.599374
3	3	4	0.599358
4	4	5	0.599358
5	5	6	0.599358
6	6	7	0.599358
7	7	8	0.599374
8	8	9	0.599374
9	9	10	0.599374
10	10	11	0.599374
11	11	12	0.599358
12	12	13	0.599358
13	13	14	0.599358
14	14	15	0.599358
15	15	16	0.599374
16	16	1	0.599374

## BOUNDARY RESULTS

NODE	FUNCTION	DERIVATIVE
1	0.000000	-0.680788
2	0.000000	-1.020366
3	0.000000	-1.359183
4	0.000000	-1.532477
5	0.000000	-1.588502
6	0.000000	-1.532475
7	0.000000	-1.359174
8	0.000000	-1.020376
9	0.000000	-0.680807
10	0.000000	-1.020356
11	0.000000	-1.359191
12	0.000000	-1.532487
13	0.000000	-1.588514
14	0.000000	-1.532477
15	0.000000	-1.359191
16	0.000000	-1.020369

## RESULTS AT INTERIOR NODES

NODE	FUNCTION
17	0.349104
18	0.418270
19	0.573598
20	0.646316
21	0.573597

22	0.418268
23	0.349101
24	0.418267
25	0.573594
26	0.646314
27	0.573596
28	0.418268
29	0.643356
30	0.789432
31	0.807675
32	0.789432
33	0.643355

### A Benchmark Problem

In order to evaluate the performance of numerical methods, a series of standard benchmark problems with known analytical solutions have been adopted. The following is from NAFEMS Thermal Analysis Benchmarks #9(ii) ref 2D/PO/CNSR [16]. The details of the problem are shown in figure 3.6.

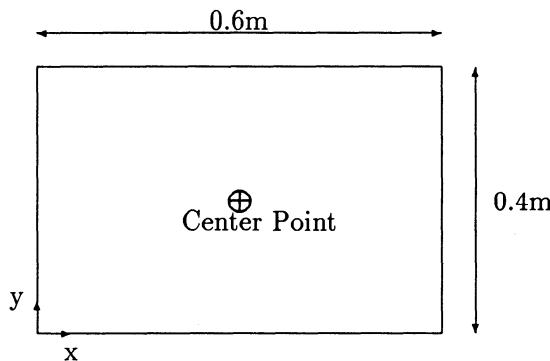


Figure 3.6: The NAFEMS Benchmark Problem

The temperature  $u$  on the boundary is maintained at  $0^\circ C$ . There is a distributed heat source of  $1000000W/m^3$  (assuming unit thickness). The temperature at the center point should be  $310.1^\circ C$  for  $K_x = K_y = 52$ .

This problem is governed by the Poisson equation

$$\nabla^2 u = -\frac{1000000}{52} \quad (3.43)$$

Using the techniques already explained, the DRM gives a value of  $314.6^\circ$  at this point, an error of 1.5%.

Two discretizations were tested, one using one linear element per 0.1 m of the boundary and one node at each corner point, and another using the same elements but with two nodes at each corner to model the discontinuity in  $q$  at these points.

In this case the corner nodes were slightly distanced from the corner point along the edges of the problem. This is necessary to avoid producing a singular  $\mathbf{F}$  matrix as was explained in section 3.2.1.

Both discretizations produced the same result at the center node.

### 3.5 Results for Different Functions $b = b(x, y)$

Examples of the use of the program given in section 3.4 will now be presented for different known functions  $b(x, y)$ . In all applications the same problem geometry will be used, that already given in figure 3.5 with the homogeneous boundary condition  $u = 0$ .

As far as the governing equation is concerned, it is necessary to modify one line of the routine ALFAF2 to take each new  $b$  function into consideration. This change will be explained in each case.

#### 3.5.1 The Case $\nabla^2 u = -x$

This case and all the others in this section will be modelled using the element geometry shown in figure 3.5. The governing equation is

$$\nabla^2 u = -x \quad (3.44)$$

Variable	Node	X	Y	Exact	$f = 1 + r$	Cell Solution
$q$	1	2.0	0.0	-0.571	-0.497	-0.493
	2	1.706	-0.522	-0.620	-0.635	-0.623
	3	1.179	-0.808	-0.556	-0.586	-0.578
	4	0.598	-0.954	-0.326	-0.334	-0.330
	5	0.0	-1.0	0.0	0.0	0.0
$u$	17	1.5	0.0	0.187	0.188	0.176
	18	1.2	-0.35	0.177	0.181	0.171
	19	0.6	-0.45	0.121	0.124	0.118
	20	0.0	-0.45	0.0	0.0	0.0
	29	0.9	0.0	0.205	0.208	0.200
	30	0.3	0.0	0.083	0.088	0.082
	31	0.0	0.0	0.0	0.0	0.0

Table 3.3: DRM and Cell Solutions of  $\nabla^2 u = -x$

The exact solution is given by

$$u = -\frac{2x}{7} \left( \frac{x^2}{4} + y^2 - 1 \right) \quad (3.45)$$

which satisfies the boundary condition  $u = 0$  on  $\Gamma$  and produces

$$q = -\frac{x}{14} \left( \frac{3x^2}{2} + 2y^2 - 2 \right) - \frac{4xy^2}{7} \quad (3.46)$$

Results for both the DRM with  $f = 1 + r$  and internal cells (Program 1) are given in table 3.3. The poor results at node 1 are attributed to the coarse discretization near this point.

Notice that to use Program 2 to obtain these results line `B(I)=-2.` should be changed to `B(I)=-X(I)` in `ALFAF2.`

### 3.5.2 The Case $\nabla^2 u = -x^2$

The problem governing equation is

$$\nabla^2 u = -x^2 \quad (3.47)$$

with geometry and boundary conditions as before. The exact solution in this case is given by

$$u = -\frac{1}{246} (50x^2 - 8y^2 + 33.6) \left( \frac{x^2}{4} + y^2 - 1 \right) \quad (3.48)$$

which again satisfies  $u = 0$  on  $\Gamma$ . The expression for the normal flux along the boundary is

$$q = \frac{1}{246} (-50x^3 - 96xy^2 + 83.2x) \frac{x}{2} + \frac{1}{246} (-96x^2y + 32y^3 - 83.2y) y \quad (3.49)$$

Table 3.4 gives the results of the DRM analysis using both  $f = 1 + r$  and  $f = r$ . To use Program 2 to obtain the results in the column  $f = 1 + r$ , line `B(I)=-2.` in subroutine `ALFAF2` should be changed to `B(I)=-X(I)**2.`

Variable	Node	X	Y	Exact	$f = r$	$f = 1 + r$
$q$	1	2.0	0.0	-0.949	-0.825	-0.827
	2	1.706	-0.522	-0.915	-0.948	-0.952
	3	1.179	-0.808	-0.657	-0.698	-0.702
	4	0.598	-0.954	-0.343	-0.339	-0.343
	5	0.0	-1.0	-0.208	-0.194	-0.197
$u$	17	1.5	0.0	0.259	0.262	0.263
	18	1.2	-0.35	0.220	0.220	0.220
	19	0.6	-0.45	0.143	0.136	0.135
	20	0.0	-0.45	0.103	0.092	0.092
	29	0.9	0.0	0.240	0.236	0.236
	30	0.3	0.0	0.151	0.142	0.142
	31	0.0	0.0	0.136	0.127	0.127

Table 3.4: DRM Solutions of  $\nabla^2 u = -x^2$ 

### 3.5.3 The Case $\nabla^2 u = a^2 - x^2$

The geometric data and boundary conditions are as before. Since the semi-major axis  $a$  of the ellipse is of length 2, the governing equation becomes

$$\nabla^2 u = 4 - x^2 \quad (3.50)$$

This problem can be viewed as a combination of a constant source  $b = 4$  and a source  $b = -x^2$ . Thus, its analytical solution is obtained by the sum of expressions of the types (3.40) and (3.48), *i.e.*

$$u = \left[ 1.6 - \frac{1}{246} (50x^2 - 8y^2 + 33.6) \right] \left( \frac{x^2}{4} + y^2 - 1 \right) \quad (3.51)$$

with

$$q = 0.4 (x^2 + 8y^2) + \frac{1}{246} (-50x^3 - 96xy^2 + 83.2x) \frac{x}{2} + \frac{1}{246} (-96x^2y + 32y^3 - 83.2y) y \quad (3.52)$$

For the DRM implementation, the two source terms can be approximated separately, and in theory each can use a different  $f$  expansion. However, in practice, it is more efficient to take the terms together and use the same  $f$  expansion as before. In Program 2, this means that line `B(I)==2.` in subroutine `ALFAF2` should be changed to `B(I)=4.-X(I)**2.`

Variable	Node	X	Y	$f = 1 + r$	Exact
$q$	1	2.0	0.0	0.533	0.650
	2	1.706	-0.522	1.088	1.121
	3	1.179	-0.808	2.016	1.986
	4	0.598	-0.954	2.721	2.713
	5	0.0	-1.0	2.979	2.991
$u$	17	1.5	0.0	-0.434	-0.440
	18	1.2	-0.35	-0.616	-0.607
	19	0.6	-0.45	-1.011	-0.988
	20	0.0	-0.45	-1.199	-1.172
	29	0.9	0.0	-1.049	-1.035
	30	0.3	0.0	-1.436	-1.412
	31	0.0	0.0	-1.488	-1.463

Table 3.5: DRM Solution of  $\nabla^2 u = 4 - x^2$ 

### 3.5.4 Results using Quadratic Elements

In the previous sections results were presented for the problem shown in figure 3.5 for a series of different  $f$  representations using a linear boundary element discretization. In this section, results will be given using quadratic elements, but using only  $f = 1 + r$ . In the discretization shown in figure 3.5, 16 linear boundary elements were used. Here, results will be obtained using 16 curved quadratic elements, such that the problem now has 32 boundary nodes. The same internal nodes were used. Although the nodes were re-numbered for programming, the original numbers will be maintained in the tables shown below for comparison.

Table 3.6 shows results obtained for  $\nabla^2 u = -x$ , with both linear and quadratic boundary elements.

Table 3.7 gives results for the same discretizations for the equation  $\nabla^2 u = -x^2$  and table 3.8 for the equation  $\nabla^2 u = 4 - x^2$ .

The implementation of quadratic elements in the program given in section 3.4 can be done in standard form [15], and does not affect the parts of the program relating to the DRM. Reference should also be made to Appendix 2 where special numerical integration formulae for curved elements are discussed.

## 3.6 Problems with Different Domain Integrals on Different Regions

### 3.6.1 The Subregion Technique

If the problem under consideration presents regions with distinct  $b$  functions, the sub-regions technique may be applied. This technique is explained in standard Boundary

Variable	Node	X	Y	Linear BE	Quadratic BE	Exact
$q$	1	2.0	0.0	-0.497	-0.543	-0.571
	2	1.706	-0.522	-0.635	-0.677	-0.620
	3	1.179	-0.808	-0.586	-0.583	-0.556
	4	0.598	-0.954	-0.334	-0.331	-0.326
	5	0.0	-1.0	0.0	0.0	0.0
$u$	17	1.5	0.0	0.188	0.188	0.187
	18	1.2	-0.35	0.181	0.178	0.177
	19	0.6	-0.45	0.124	0.121	0.121
	20	0.0	-0.45	0.0	0.0	0.0
	29	0.9	0.0	0.208	0.206	0.205
	30	0.3	0.0	0.088	0.084	0.083
	31	0.0	0.0	0.0	0.0	0.0

Table 3.6: Results for Linear and Quadratic Boundary Elements for  $\nabla^2 u = -x$ 

Variable	Node	X	Y	Linear BE	Quadratic BE	Exact
$q$	1	2.0	0.0	-0.827	-0.957	-0.949
	2	1.706	-0.522	-0.952	-0.986	-0.915
	3	1.179	-0.808	-0.702	-0.684	-0.657
	4	0.598	-0.954	-0.343	-0.336	-0.343
	5	0.0	-1.0	-0.197	-0.193	-0.208
$u$	17	1.5	0.0	0.263	0.263	0.259
	18	1.2	-0.35	0.220	0.221	0.220
	19	0.6	-0.45	0.135	0.142	0.143
	20	0.0	-0.45	0.092	0.101	0.103
	29	0.9	0.0	0.236	0.240	0.240
	30	0.3	0.0	0.142	0.149	0.151
	31	0.0	0.0	0.127	0.134	0.136

Table 3.7: Results for Linear and Quadratic Boundary Elements for  $\nabla^2 u = -x^2$

Variable	Node	X	Y	Linear BE	Quadratic BE	Exact
$q$	1	2.0	0.0	0.533	0.591	0.650
	2	1.706	-0.522	1.088	1.181	1.121
	3	1.179	-0.808	2.016	2.063	1.986
	4	0.598	-0.954	2.721	2.748	2.713
	5	0.0	-1.0	2.979	3.000	2.991
$u$	17	1.5	0.0	-0.434	-0.433	-0.440
	18	1.2	-0.35	-0.616	-0.603	-0.607
	19	0.6	-0.45	-1.011	-0.986	-0.988
	20	0.0	-0.45	-1.199	-1.172	-1.172
	29	0.9	0.0	-1.049	-1.032	-1.035
	30	0.3	0.0	-1.436	-1.411	-1.412
	31	0.0	0.0	-1.488	-1.462	-1.463

Table 3.8: Results for Linear and Quadratic Boundary Elements for  $\nabla^2 u = 4 - x^2$ 

Element text books such as [1], [2] and [15], and may easily be combined with DRM analysis.

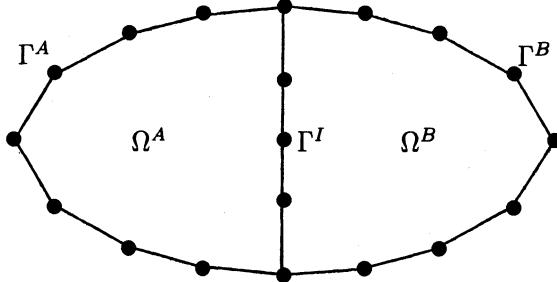


Figure 3.7: Problem with Two Subregions

In figure 3.7, a domain composed of two subregions is shown. Consider that there are  $N^A$ ,  $N^I$  and  $L^A$  nodes on the external boundary  $\Gamma^A$ , interface  $\Gamma^I$  and domain  $\Omega^A$ , respectively, with similar definitions for  $N^B$  and  $L^B$ . Also, consider that in region  $A$

$$\nabla^2 u = b^A \quad (3.53)$$

and in region  $B$

$$\nabla^2 u = b^B \quad (3.54)$$

where  $b^A \neq b^B$  and both are known functions of  $(x, y)$ . Then, considering the boundary nodes on  $\Gamma^A$  and  $\Gamma^I$  and the internal nodes on  $\Omega^A$ , the usual DRM approximation may be written for subregion  $A$

$$\mathbf{b}^A = \mathbf{F}^A \boldsymbol{\alpha}^A \quad (3.55)$$

from which  $\boldsymbol{\alpha}^A$ , of size  $(N^A + N^I + L^A)$ , can be obtained. A similar equation may be written for region  $B$  to obtain  $\boldsymbol{\alpha}^B$ , of size  $(N^B + N^I + L^B)$ . After evaluating  $\boldsymbol{\alpha}^A$  and  $\boldsymbol{\alpha}^B$ , equation (3.15) may be written for each subregion

$$\mathbf{H}^A \mathbf{u}^A - \mathbf{G}^A \mathbf{q}^A = (\mathbf{H}^A \hat{\mathbf{U}}^A - \mathbf{G}^A \hat{\mathbf{Q}}^A) \boldsymbol{\alpha}^A \quad (3.56)$$

$$\mathbf{H}^B \mathbf{u}^B - \mathbf{G}^B \mathbf{q}^B = (\mathbf{H}^B \hat{\mathbf{U}}^B - \mathbf{G}^B \hat{\mathbf{Q}}^B) \boldsymbol{\alpha}^B \quad (3.57)$$

Since the right-hand sides are known vectors in both cases, (3.54) and (3.55) simplify to

$$\mathbf{H}^A \mathbf{u}^A - \mathbf{G}^A \mathbf{q}^A = \mathbf{d}^A \quad (3.58)$$

and

$$\mathbf{H}^B \mathbf{u}^B - \mathbf{G}^B \mathbf{q}^B = \mathbf{d}^B \quad (3.59)$$

The matrices in the above equations may be partitioned into one submatrix referring to the boundary,  $\Gamma^A$  or  $\Gamma^B$  respectively, and one part referring to the interface  $\Gamma^I$  as follows

$$\begin{bmatrix} \mathbf{H}_\Gamma^A & \mathbf{H}_I^A \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma^A \\ \mathbf{u}_I^A \end{bmatrix} - \begin{bmatrix} \mathbf{G}_\Gamma^A & \mathbf{G}_I^A \end{bmatrix} \begin{bmatrix} \mathbf{q}_\Gamma^A \\ \mathbf{q}_I^A \end{bmatrix} = \begin{bmatrix} \mathbf{d}^A \end{bmatrix} \quad (3.60)$$

$$\begin{bmatrix} \mathbf{H}_\Gamma^B & \mathbf{H}_I^B \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma^B \\ \mathbf{u}_I^B \end{bmatrix} - \begin{bmatrix} \mathbf{G}_\Gamma^B & \mathbf{G}_I^B \end{bmatrix} \begin{bmatrix} \mathbf{q}_\Gamma^B \\ \mathbf{q}_I^B \end{bmatrix} = \begin{bmatrix} \mathbf{d}^B \end{bmatrix} \quad (3.61)$$

Along the interface, where both  $u$  and  $q$  are unknown, the following compatibility and equilibrium conditions hold [15]

$$\begin{aligned} u^A &= u^B = u_I \\ q^A &= -q^B = q_I \end{aligned} \quad (3.62)$$

Thus, equations (3.58) and (3.59) may be combined to form

$$\begin{bmatrix} \mathbf{H}_\Gamma^A & \mathbf{H}_I^A & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_I^B & \mathbf{H}_\Gamma^B \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma^A \\ \mathbf{u}_I^A \\ \mathbf{u}_\Gamma^B \end{bmatrix} - \begin{bmatrix} \mathbf{G}_\Gamma^A & \mathbf{G}_I^A & \mathbf{0} \\ \mathbf{0} & -\mathbf{G}_I^B & \mathbf{G}_\Gamma^B \end{bmatrix} \begin{bmatrix} \mathbf{q}_\Gamma^A \\ \mathbf{q}_I^A \\ \mathbf{q}_\Gamma^B \end{bmatrix} = \begin{bmatrix} \mathbf{d}^A \\ \mathbf{d}^B \end{bmatrix} \quad (3.63)$$

The system (3.61) can be rearranged to read

$$\begin{bmatrix} \mathbf{H}_\Gamma^A & \mathbf{H}_I^A & -\mathbf{G}_I^A & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_I^B & \mathbf{G}_I^B & \mathbf{H}_\Gamma^B \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma^A \\ \mathbf{u}_I \\ \mathbf{q}_I \\ \mathbf{u}_\Gamma^B \end{bmatrix} = \begin{bmatrix} \mathbf{G}_\Gamma^A & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_\Gamma^B \end{bmatrix} \begin{bmatrix} \mathbf{q}_\Gamma^A \\ \mathbf{q}_\Gamma^B \end{bmatrix} + \begin{bmatrix} \mathbf{d}^A \\ \mathbf{d}^B \end{bmatrix} \quad (3.64)$$

such that the matrix on the left is square but that on the right is not. Applying the boundary conditions in the usual way, the system

$$\mathbf{Ax} = \mathbf{y} \quad (3.65)$$

is produced, where  $\mathbf{A}$  is of size  $(N^A + 2N^I + N^B) \times (N^A + 2N^I + N^B)$ .

Results at internal nodes can be obtained in standard form once all values are known for the boundaries and the interface. This involves using equation (3.22), but considering each subregion separately.

### 3.6.2 Integration over Internal Regions

If only a small portion of the domain has a distinct distributed source, this may be handled by integrating around the boundary of this internal region. This technique was developed by Niku and Brebbia [10].

Consider the domain  $\Omega$  shown in figure 3.8, where the Laplace equation holds, containing a sub-domain  $\Omega^B$  where a distributed source  $b$  exists.

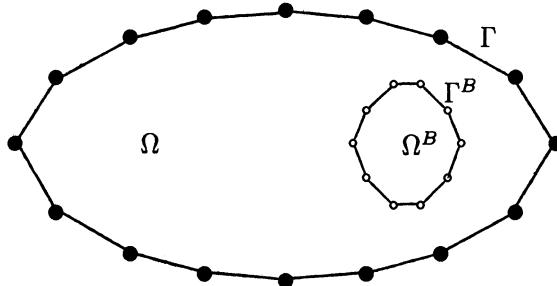


Figure 3.8: Distributed Source on Small Internal Region  $\Omega^B$

Equation (3.13) gives the DRM relationship for the case of  $b$  acting over the whole of  $\Omega$ . The corresponding expression for this case can be obtained observing the following:

- (a) Source points are only defined on  $\Gamma$ , that is to say, there will be no unknowns associated with  $\Gamma^B$ .

- (b) There will be a series of points on  $\Gamma^B$  associated with the elements used to discretize this internal boundary, however these will not be node points but collocation points for the calculation of  $\mathbf{H}^B$ ,  $\mathbf{G}^B$ ,  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  on the right-hand side of the equation. Let these points be  $N^B$  in number.
- (c) As a consequence,  $\mathbf{H}^B$  and  $\mathbf{G}^B$  on the right-hand side of equation (3.15) have no relation to  $\mathbf{H}$  and  $\mathbf{G}$  on the left-hand side.
- (d) Since the points on  $\Gamma^B$  are not source points and do not have unknowns associated with them, the  $c_i \hat{u}_{ij}$  terms of (3.13) are zero. The source points on  $\Gamma$  are exterior to  $\Gamma^B$ .

Given the above, equation (3.13) becomes for this case

$$c_i u_i + \sum_{k=1}^N H_{ik} u_k - \sum_{k=1}^N G_{ik} q_k = \sum_{j=1}^{N^B} \alpha_j \left( \sum_{\ell=1}^{N^B} H_{i\ell}^B \hat{u}_{\ell j} - \sum_{\ell=1}^{N^B} G_{i\ell}^B \hat{q}_{\ell j} \right) \quad (3.66)$$

Note that the summations in  $j$  and  $\ell$  refer only to points on  $\Gamma^B$  and do not include the nodes on  $\Gamma$ . The values of  $\alpha_j$  are calculated from

$$\mathbf{F}^B \boldsymbol{\alpha}^B = \mathbf{b}^B \quad (3.67)$$

where  $\mathbf{F}^B$  is of size  $(N^B \times N^B)$  and  $\mathbf{b}^B$  contains the values of the distributed source at the points on  $\Gamma^B$ . The set of equations obtained by the application of (3.64) to the collocation points can thus be written in the usual form

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{d} \quad (3.68)$$

to which the boundary conditions may be applied to produce  $\mathbf{Ax} = \mathbf{y}$ .

## 3.7 References

1. Brebbia, C. A.  
The Boundary Element Method for Engineers, Pentech Press, London, 1978.
2. Brebbia, C. A., Telles, J. C. F. and Wrobel, L. C.  
Boundary Element Techniques, Springer-Verlag, Berlin and New York, 1984.
3. Takhteyev, V. and Brebbia, C. A.  
Analytical Integrations in Boundary Elements, Technical Note, *Engineering Analysis*, Vol. 7, No. 2, pp 95-100, 1990.
4. Tang, W., Brebbia, C. A. and Telles, J. C. F.  
A Generalized Approach to Transfer the Domain Integrals onto Boundary Ones for Potential Problems in BEM, in *Boundary Elements IX*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.
5. Nowak, A. J. and Brebbia, C. A.  
The Multiple Reciprocity Method: A New Approach for Transforming BEM Domain Integrals to the Boundary, *Engineering Analysis*, Vol. 6, No. 3, pp 164-167, 1989.

6. Nardini, D. and Brebbia, C. A.  
A New Approach to Free Vibration Analysis using Boundary Elements, in *Boundary Element Methods in Engineering*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1982.
7. Nardini, D. and Brebbia, C. A.  
Boundary Integral Formulation of Mass Matrices for Dynamic Analysis, in *Topics in Boundary Element Research*, Vol. 2, Springer-Verlag, Berlin and New York, 1985.
8. Wrobel, L. C., Brebbia, C. A. and Nardini, D.  
The Dual Reciprocity Boundary Element Formulation for Transient Heat Conduction, in *Finite Elements in Water Resources VI*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
9. Wrobel, L. C. and Brebbia, C. A.  
The Dual Reciprocity Boundary Element Formulation for Non-Linear Diffusion Problems, *Computer Methods in Applied Mechanics and Engineering*, Vol 65, No. 2, pp 147-164, 1987.
10. Niku, S. M. and Brebbia, C. A.  
Dual Reciprocity Boundary Element Formulation for Potential Problems with Arbitrarily Distributed Forces, Technical Note, *Engineering Analysis*, Vol. 5, No. 1, pp 46-48, 1988.
11. Partridge, P. W. and Brebbia, C. A.  
Computer Implementation of the BEM Dual Reciprocity Method for the Solution of Poisson Type Equations, *Software for Engineering Workstations*, Vol. 5, No. 4, pp 199-206, 1989.
12. Partridge, P. W. and Brebbia, C. A.  
Computer Implementation of the BEM Dual Reciprocity Method for the Solution of General Field Problems, *Communications in Applied Numerical Methods*, Vol. 6, No. 2, pp 83-92, 1990.
13. Partridge, P. W. and Brebbia, C. A.  
The BEM Dual Reciprocity Method for Diffusion Problems, in *Computational Methods in Water Resources VIII*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1990.
14. Partridge, P. W. and Wrobel, L. C.  
The Dual Reciprocity Boundary Element Method for Spontaneous Ignition, *International Journal for Numerical Methods in Engineering*, Vol. 30, pp 953-963, 1990.
15. Brebbia, C. A. and Dominguez, J.  
Boundary Elements: An Introductory Course, Computational Mechanics Publications, Southampton and McGraw-Hill, New York, 1989.
16. Cameron, A. D., Casey, J. A. and Simpson, G. B.  
Benchmark Tests for Thermal Analysis, NAFEMS Publications, Glasgow, 1986.

# Chapter 4

## The Dual Reciprocity Method for Equations of the Type

$$\nabla^2 u = b(x, y, u)$$

### 4.1 Introduction

In the previous chapter the boundary element Dual Reciprocity Method was developed and applied to problems governed by a Poisson-type equation in which the right-hand side is a known function of position, *i.e.*

$$\nabla^2 u = b(x, y) \quad (4.1)$$

for which the basic DRM matrix expression was established in the form

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\boldsymbol{\alpha} = \mathbf{d} \quad (4.2)$$

In such analyses, as is usual in BEM, the solution is divided into two parts:

- (a) A boundary solution, and
- (b) An internal solution.

Since  $b$  in equation (4.1) is a known function, the vector  $\boldsymbol{\alpha}$  may be calculated from the relationship

$$\boldsymbol{\alpha} = \mathbf{F}^{-1}\mathbf{b} \quad (4.3)$$

In this chapter, the range of application of the DRM will be extended to problems governed by equations of the type

$$\nabla^2 u = b(x, y, u) \quad (4.4)$$

where the non-homogeneous term may also be a combination, sum or product, of functions. This classification of equations is due to the way in which the DRM is

applied. Equations of the type (4.4) involve the Laplacian plus one or more additional terms, each of which is a function of the problem variable,  $u$ . This includes linear as well as non-linear equations. In DRM analysis only truly non-linear problems are solved using iterative techniques, whereas if cell integration is employed any equation of the type (4.4), linear or non-linear, would require iterations.

Thus, a very large number of engineering problems may be studied with the techniques explained in this chapter. A computer program for solving equations of the type (4.4) will be presented in section 4.5. In section 4.6, three-dimensional analysis will be considered and in chapter 5 the method is applied to time-dependent problems.

As an introduction and in order to generate the necessary basic relationships, the case

$$\nabla^2 u + u = 0 \quad (4.5)$$

is considered. This is the simplest equation of the type (4.4) which may be postulated. Function  $b$  is now defined as  $-u$ . Thus, from (4.3)

$$\alpha = -\mathbf{F}^{-1}\mathbf{u} \quad (4.6)$$

and equation (4.2) becomes

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = -(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\mathbf{u} \quad (4.7)$$

This may be represented in schematic form as

$$\begin{array}{ccccc}
 \begin{array}{c} N \\ \text{---} \\ \text{---} \\ L \end{array} & & & & \\
 \begin{array}{|c|c|} \hline BS & 0 \\ \hline IS & \mathbf{I} \\ \hline \end{array} & - & \begin{array}{|c|c|} \hline BS & 0 \\ \hline IS & 0 \\ \hline \end{array} & = & \begin{array}{|c|c|} \hline BS \\ \hline IS \\ \hline \end{array} \\
 \mathbf{H} & \mathbf{u} & \mathbf{G} & \mathbf{q} &
 \end{array}
 \quad (4.8)$$
  

$$\begin{array}{ccccc}
 \left[ \begin{array}{|c|c|} \hline BS & 0 \\ \hline IS & 0 \\ \hline \end{array} \right] & - & \begin{array}{|c|c|} \hline BS & 0 \\ \hline 0 & \\ \hline \end{array} & \left[ \begin{array}{|c|c|} \hline (BS+IS) & \\ \hline (BS+IS) & \mathbf{I} \\ \hline \end{array} \right] & \left[ \begin{array}{|c|c|} \hline (BS+IS) & BS \\ \hline (BS+IS) & IS \\ \hline \end{array} \right] \\
 \mathbf{G} & \hat{\mathbf{Q}} & \mathbf{H} & \hat{\mathbf{U}} & \mathbf{F}^{-1} \mathbf{u}
 \end{array}$$

In equation (4.8) the zero submatrices are marked  $\mathbf{0}$ , identity matrices are marked  $\mathbf{I}$  and other non-zero submatrices are marked  $BS$ ,  $IS$  or  $(BS+IS)$ , as defined in equation (3.34). For this class of problems it is no longer strictly possible to separate

boundary and interior solutions as the presence of the fully populated matrix  $\mathbf{F}^{-1}$  results in a coupled problem in which both sets of values are calculated simultaneously.

For the moment, it may be noted that the results have the same accuracy as when separate boundary and interior solutions are obtained and their coupling has been successfully done to take advantage of the resulting simplification in programming effort. In chapter 5, a method for the partial uncoupling of this type of problem will be presented, which requires less computer memory.

The matrices  $\mathbf{G}$ ,  $\mathbf{H}$ ,  $\hat{\mathbf{Q}}$  and  $\hat{\mathbf{U}}$  of equation (4.8) were defined and used in the previous chapter. The matrix  $\mathbf{F}$  was defined but only used in connection with the calculation of vector  $\alpha$ . When dealing with problems governed by equations of the type (4.4) or the time-dependent cases to be covered in chapter 5,  $\alpha$  cannot be obtained explicitly and will always be expressed in the matrix equation as  $\mathbf{F}^{-1}\mathbf{b}$  (in the current example,  $\mathbf{b} = -\mathbf{u}$ ). The right-hand side of (4.7) thus becomes a matrix expression multiplying the unknown  $\mathbf{b}$  which will be different in each case. The calculation of  $\mathbf{F}^{-1}$  poses no problems if the constant is included in the  $f$  expansion (3.25) and if no two nodes coincide. Matrix  $\mathbf{F}$  depends only on geometric data and has no relation to either governing equation or boundary conditions. It may be calculated once and stored in a data file for use with all subsequent analyses involving the same discretization.

For the examples to be considered in this chapter, the right-hand side of (4.4) is an unknown function. The known vector  $\mathbf{d}$ , equation (4.2), has to be replaced by a matrix expression, equation (4.7). Defining

$$\mathbf{S} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1} \quad (4.9)$$

then equation (4.7) becomes

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = -\mathbf{S}\mathbf{u} \quad (4.10)$$

The calculation of  $\mathbf{S}$  is done by multiplying known matrices. Collecting the terms in  $\mathbf{u}$  on the left-hand side produces

$$(\mathbf{H} + \mathbf{S})\mathbf{u} = \mathbf{G}\mathbf{q} \quad (4.11)$$

On the boundary,  $N$  values of  $u$  and  $q$  are unknown while the  $L$  values of  $u$  at interior nodes are all unknown. Note that the part of vector  $\mathbf{q}$  from position  $N+1$  to position  $N+L$ , where  $q$  is not defined, is multiplied by zero partitions of  $\mathbf{G}$ .

After applying the boundary conditions the usual equation

$$\mathbf{A}\mathbf{x} = \mathbf{y} \quad (4.12)$$

is obtained where  $\mathbf{A}$  is of size  $(N+L) \times (N+L)$  and  $\mathbf{x}$  contains  $N$  boundary values of  $u$  or  $q$  plus  $L$  interior values of  $u$ .

Equations (4.7) to (4.12) form the basis of application of the Dual Reciprocity Method to equations of the type (4.4). The only difference in each new case will be a new vector  $\mathbf{b}$  to replace  $-\mathbf{u}$  in equations (4.7) and (4.8). The treatment of these

vectors will be explained for each case to be studied in a systematic way and the reader will have no difficulty in proceeding from one case to the next or extending the analysis to other problems.

The computer program presented in section 4.5 can be used to analyse an equation of the type (4.5) if the parameters IDDX and IDDY are read as zero. Some results for equation (4.5) are now presented for the geometry shown in figure 4.1.

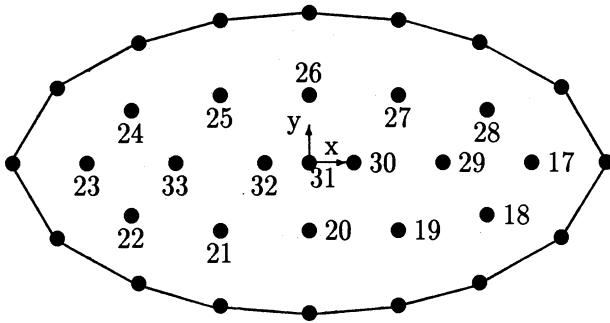


Figure 4.1: Ellipse Problem Discretization

Since homogeneous boundary conditions will result in the trivial solution  $u = q = 0$  at all nodes, a non-homogeneous condition has to be used, for example

$$u = \sin x \quad (4.13)$$

Equation (4.13) may easily be shown to be a particular solution of (4.5) such that, if imposed as a boundary condition, it will also constitute the problem solution and may be used to check the results.

The problem domain is shown in figure 4.1 and consists of an ellipse which has a semi-major axis of length 2 and a semi-minor axis of length 1. The boundary is discretized into 16 linear boundary elements and the solution calculated at 17 internal nodes. Results are given in table 4.1 for the expansions  $f = 1+r$  and  $f = 1+r+r^2+r^3$ .

It can be seen immediately that the simplest  $f$  expansion,  $f = 1 + r$  produces excellent results. For comparison, results for  $f = 1 + r + r^2 + r^3$  are included but evidently it is unnecessary to increase the order of the  $f$  expansion in this case. The linear elements are also seen to be accurate.

## 4.2 The Convective Case

Differential equations including first-order space derivatives of the problem variable are very common in the mathematical modelling of engineering problems. Whilst

	$x$	$y$	$f = 1 + r$	$f = 1 + r + r^2 + r^3$	Eqn. (4.13)
$u_{17}$	1.5	0.0	0.994	0.995	0.997
$u_{18}$	1.2	-0.35	0.928	0.932	0.932
$u_{19}$	0.6	-0.45	0.562	0.566	0.565
$u_{20}$	0.0	-0.45	0.0	0.0	0.0
$u_{29}$	0.9	0.0	0.780	0.784	0.783
$u_{30}$	0.3	0.0	0.294	0.296	0.295
$u_{31}$	0.0	0.0	0.0	0.0	0.0

Table 4.1: Results for  $\nabla^2 u = -u$ 

there is no practical difficulty in modelling equations containing these terms with finite elements or finite differences, the results have been shown in many cases to be inaccurate and special techniques, e.g. “upwinding”, have been introduced to deal with the resulting short wave oscillations in the solution. This question is discussed in, for example, references [1,2]. Such problems do not occur in boundary element analysis but the treatment of these derivatives is difficult using internal cells, e.g. reference [3].

By contrast, convective terms can be easily accommodated in the DRM treatment given in section 4.1. Consider, for example, an equation of the type

$$\nabla^2 u = \frac{\partial u}{\partial x} \quad (4.14)$$

Comparing this with (4.4) it is seen that in this case  $b = \partial u / \partial x$ , such that  $\mathbf{b} = \partial \mathbf{u} / \partial x$ , i.e. the nodal values of the derivative of  $u$  with respect to  $x$ . Thus, substituting into equation (4.3), one obtains

$$\boldsymbol{\alpha} = \mathbf{F}^{-1} \frac{\partial \mathbf{u}}{\partial x} \quad (4.15)$$

and equation (4.7) becomes for this case

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1} \frac{\partial \mathbf{u}}{\partial x} \quad (4.16)$$

or, using (4.9),

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{S} \frac{\partial \mathbf{u}}{\partial x} \quad (4.17)$$

A mechanism must now be established to relate the nodal values of  $u$  to the nodal values of its derivative  $\partial u / \partial x$ . At this point it should be remembered that the basic approximation of the DRM technique is equation (3.17), i.e.

$$\mathbf{b} = \mathbf{F}\boldsymbol{\alpha} \quad (4.18)$$

A similar equation may be written for  $u$

$$\mathbf{u} = \mathbf{F}\beta \quad (4.19)$$

where  $\beta \neq \alpha$ . Differentiating (4.19) produces

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\partial \mathbf{F}}{\partial x} \beta \quad (4.20)$$

Rewriting equation (4.19) as  $\beta = \mathbf{F}^{-1}\mathbf{u}$ , then (4.20) becomes

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \mathbf{u} \quad (4.21)$$

Substituting into (4.17) one obtains

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{S} \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \mathbf{u} \quad (4.22)$$

Calling

$$\mathbf{R} = \mathbf{S} \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \quad (4.23)$$

produces the system of equations

$$(\mathbf{H} - \mathbf{R})\mathbf{u} = \mathbf{G}\mathbf{q} \quad (4.24)$$

which can be handled in exactly the same way as (4.11). Both  $\partial \mathbf{F}/\partial x$  and  $\mathbf{F}^{-1}$  are fully-populated  $(N + L) \times (N + L)$  matrices. The terms in the  $\partial \mathbf{F}/\partial x$  matrix are obtained by differentiating the  $f$  expansion

$$\begin{aligned} \frac{\partial f}{\partial x} &= \frac{\partial f}{\partial r} \frac{\partial r}{\partial x} = \frac{\partial f}{\partial r} \frac{r_x}{r} = \\ &= \frac{r_x}{r} + 2r_x + 3rr_x + \dots + mr_x r^{m-2} \end{aligned} \quad (4.25)$$

It is important to note that  $\partial \mathbf{F}/\partial x$  is a skew-symmetric geometric matrix, the evaluation of which does not involve the source points; thus, the components  $r_x$  in (4.25) are of opposite sign to those used in the definition of the  $\mathbf{H}$  matrix (see section 3.3.2).

A similar treatment can be carried out in the case of the equation

$$\nabla^2 u = \frac{\partial u}{\partial y} \quad (4.26)$$

for which the same equation (4.24) will be obtained with

$$\mathbf{R} = \mathbf{S} \frac{\partial \mathbf{F}}{\partial y} \mathbf{F}^{-1} \quad (4.27)$$

where the terms of  $\partial \mathbf{F}/\partial y$  are defined by

$$\frac{\partial f}{\partial y} = \frac{r_y}{r} + 2r_y + 3rr_y + \dots + mr_y r^{m-2} \quad (4.28)$$

with  $r_y$  having the same sign as  $r_x$  as discussed above.

### 4.2.1 Results for the Case $\nabla^2 u = -\partial u / \partial x$

Some results will be presented for the equation  $\nabla^2 u = -\partial u / \partial x$ . The geometry of the problem analysed is the same as in the last section. A particular solution for this case is

$$u = e^{-x} \quad (4.29)$$

which, when imposed as an essential boundary condition, also constitutes the problem solution as discussed in section 4.1 for expression (4.13).

In the previous chapter, the  $f$  expansions  $f = r$ ,  $f = 1 + r$  and  $f = 1$  at one node,  $f = r$  at the remaining nodes were considered for the case  $b = b(x, y)$ , equation (3.1). In the applications discussed in this chapter, the “complexity” of the  $b$  functions is higher and hence the necessity for higher-order  $f$  expansions must be investigated.

For the equations containing derivatives in this chapter, the following  $f$  expansions are considered:

Case (i)  $f = 1 + r$

Case (ii)  $f = 1 + r + r^2$

Case (iii)  $f = 1 + r + r^2 + r^3$

Results for  $\nabla^2 u = -\partial u / \partial x$  are shown in table 4.2. The results at the 12 points given in the table describe the complete solution of the problem due to symmetry.

	$x$	$y$	$f$ Case(i)	$f$ Case(ii)	$f$ Case(iii)	Eqn. (4.29)
$u_{17}$	1.5	0.0	0.229	0.229	0.214	0.223
$u_{18}$	1.2	-0.35	0.307	0.309	0.274	0.301
$u_{19}$	0.6	-0.45	0.555	0.560	0.523	0.549
$u_{20}$	0.0	-0.45	1.003	1.011	1.006	1.000
$u_{21}$	-0.6	-0.45	1.819	1.828	1.833	1.822
$u_{22}$	-1.2	-0.35	3.323	3.324	3.318	3.320
$u_{23}$	-1.5	0.0	4.489	4.479	4.465	4.482
$u_{29}$	0.9	0.0	0.411	0.415	0.363	0.406
$u_{30}$	0.3	0.0	0.745	0.751	0.725	0.741
$u_{31}$	0.0	0.0	1.002	1.010	1.002	1.000
$u_{32}$	-0.3	0.0	1.348	1.358	1.361	1.350
$u_{33}$	-0.9	0.0	2.448	2.457	2.462	2.460

Table 4.2: Results for  $\nabla^2 u = -\partial u / \partial x$

From table 4.2 it can be seen that the best results are obtained for case (i), which used the simplest expansion  $f = 1 + r$ . These results indicate that the higher-order  $f$  expansions are unnecessary, at least in this case. The largest errors in table 4.2, which are less than 1%, are found for case (iii) which used the expansion  $f = 1 + r + r^2 + r^3$ . This suggests that the higher-order  $f$  expansions may be, using the

standard terminology, “too stiff”. This conclusion is also verified by results obtained for other examples in this chapter.

### 4.2.2 Results for the Case $\nabla^2 u = -(\partial u / \partial x + \partial u / \partial y)$

Consider the equation

$$\nabla^2 u = -\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y}\right) \quad (4.30)$$

Using the method previously given produces the DRM system of equations

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = -\mathbf{S}\left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right)\mathbf{F}^{-1}\mathbf{u} \quad (4.31)$$

where matrix  $\mathbf{S}$  is defined in (4.9). Defining matrix  $\mathbf{R}$  now as

$$\mathbf{R} = \mathbf{S}\left(\frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y}\right)\mathbf{F}^{-1} \quad (4.32)$$

the system (4.31) can be rewritten in the form

$$(\mathbf{H} + \mathbf{R})\mathbf{u} = \mathbf{G}\mathbf{q} \quad (4.33)$$

Boundary conditions can be applied to the above equation and the resulting system solved in the usual way.

Results are presented for an example using the geometry shown in figure 4.1. A particular solution to equation (4.30) is given by the expression

$$u = e^{-x} + e^{-y} \quad (4.34)$$

which is also imposed as an essential boundary condition. The solution is completely non-symmetric and hence results are given in table 4.3 for all internal nodes. The same  $f$  expansions considered in the previous section are again used here.

The same observations as before can be made regarding this table, *i.e.* that the results are accurate and no advantage is noticed in using the higher-order  $f$  expansions.

### 4.2.3 Internal Derivatives of the Problem Variables

In chapter 2, expressions (2.34) were presented as the standard equations for obtaining the derivatives of the problem variable at internal points after the complete boundary solution is known. Sometimes it is difficult to evaluate these expressions due to the presence of domain integrals. This can easily be understood if a domain integral is added to expressions (2.34) to produce

	$x$	$y$	$f$ Case(i)	$f$ Case(ii)	$f$ Case(iii)	Eqn. (4.34)
$u_{17}$	1.5	0.0	1.231	1.230	1.214	1.223
$u_{18}$	1.2	-0.35	1.714	1.714	1.669	1.720
$u_{19}$	0.6	-0.45	2.107	2.107	2.057	2.117
$u_{20}$	0.0	-0.45	2.557	2.558	2.547	2.568
$u_{21}$	-0.6	-0.45	3.378	3.378	3.385	3.390
$u_{22}$	-1.2	-0.35	4.745	4.735	4.718	4.739
$u_{23}$	-1.5	0.0	5.485	5.474	5.451	5.482
$u_{24}$	-1.2	-0.35	4.017	4.021	4.004	4.025
$u_{25}$	-0.6	-0.45	2.456	2.460	2.437	2.460
$u_{26}$	0.0	0.45	1.641	1.639	1.615	1.637
$u_{27}$	0.6	0.45	1.192	1.188	1.153	1.186
$u_{28}$	1.2	0.35	1.014	1.010	0.982	1.006
$u_{29}$	0.9	0.0	1.400	1.400	1.345	1.406
$u_{30}$	0.3	0.0	1.731	1.733	1.691	1.741
$u_{31}$	0.0	0.0	1.989	1.992	1.963	2.000
$u_{32}$	-0.3	0.0	2.335	2.340	2.351	2.350
$u_{33}$	-0.9	0.0	3.438	3.444	3.428	3.460

Table 4.3: Results for  $\nabla^2 u = -(\partial u / \partial x + \partial u / \partial y)$ 

$$\begin{aligned}\frac{\partial u}{\partial x} &= \int_{\Gamma} q \frac{\partial u^*}{\partial x} d\Gamma - \int_{\Gamma} u \frac{\partial q^*}{\partial x} d\Gamma + \int_{\Omega} b \frac{\partial u^*}{\partial x} d\Omega \\ \frac{\partial u}{\partial y} &= \int_{\Gamma} q \frac{\partial u^*}{\partial y} d\Gamma - \int_{\Gamma} u \frac{\partial q^*}{\partial y} d\Gamma + \int_{\Omega} b \frac{\partial u^*}{\partial y} d\Omega\end{aligned}\quad (4.35)$$

The use of the DRM provides a simple alternative to (4.35) which can be used without modification in any case, independently of governing equation, type of domain source, etc. In DRM programs the matrices to be used will be available in the computer memory, thus the calculation involved consists only of the multiplication of known matrices and can be done in a few lines of the program at very low computer cost.

Consider the basic DRM expression

$$\mathbf{b} = \mathbf{F}\boldsymbol{\alpha} \quad (4.36)$$

In the treatment of derivatives it has been seen that the same expression can be used for the problem variable such that

$$\mathbf{u} = \mathbf{F}\boldsymbol{\beta} \quad (4.37)$$

Differentiating the above produces

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\partial \mathbf{F}}{\partial x} \boldsymbol{\beta} \quad (4.38)$$

Inverting (4.37) and substituting into (4.38) produces

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \mathbf{u} \quad (4.39)$$

Thus, the nodal values of the derivative are expressed as the product of two known matrices and the known nodal values of the problem variable. A similar equation can be deduced for the derivatives with respect to  $y$ . Matrix  $\mathbf{F}^{-1}$  has already been computed and stored in the computer memory.

Some results for derivatives calculated at internal nodes using this process are given in table 4.4 for the problem analysed in section 4.1. Exact results have been obtained differentiating equation (4.13).

Node	Eqn. (4.39)	Exact
18	0.343	0.362
19	0.817	0.825
20	0.992	1.000
29	0.617	0.621
30	0.943	0.955
31	0.992	1.000

Table 4.4: Values of  $\partial u / \partial x$  at Nodes Shown in Figure 4.1

The results can be seen to be accurate even for the very small number of boundary elements used. This method may be applied in any situation and represents a very simple and powerful extension of the DRM technique.

### 4.3 The Helmholtz Equation

In this section a standard eigenvalue/eigenvector problem will be considered. The solution to the Helmholtz equation provides the natural or fundamental frequencies and vibration modes of a system. The equation in its usual form is given by

$$\nabla^2 u + \mu^2 u = 0 \quad (4.40)$$

where the coefficient  $\mu$  is related to the natural frequencies of the system.

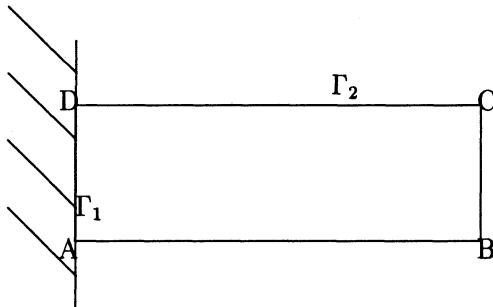


Figure 4.2: Vibrating Beam

Consider the vibrating beam shown in figure 4.2. In this case  $u$  represents displacements and  $\mu^2 = \rho\omega^2/E$  where  $\rho$  and  $E$  are material properties and  $\omega$  are the natural frequencies [4]. In the case of harbour resonance problems the equation becomes [5]

$$\nabla^2 H + \frac{\omega^2}{hg} H = 0 \quad (4.41)$$

where  $h$  is the mean water depth measured from a reference datum,  $g$  is gravity acceleration,  $\omega$  is the natural frequency as before and  $H = h + \eta$ , i.e. mean water depth plus free surface elevation. The Helmholtz equation also has important applications in acoustics [6] and other areas.

### 4.3.1 DRM Formulations

The Helmholtz equation can be modelled with the DRM noticing that the source term in (4.4) is now

$$b = -\mu^2 u \quad (4.42)$$

such that (4.6) becomes

$$\alpha = -\mu^2 F^{-1} u \quad (4.43)$$

This gives the matrix equation

$$Hu - Gq = -\mu^2 (H\hat{U} - G\hat{Q})F^{-1}u \quad (4.44)$$

or, using the  $S$  matrix defined by (4.9),

$$Hu - Gq = -\mu^2 Su \quad (4.45)$$

This equation has been studied by Nardini and Brebbia [7] who partitioned the matrices as shown below to eliminate the terms in  $q$  which are unknown on part of the boundary:

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} - \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} = -\mu^2 \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (4.46)$$

Considering the problem of the vibrating beam shown in figure 4.2, the beam is clamped on the part *AD* of the boundary, so  $u = 0$ . On the free boundary *ABCD* in the figure the prescribed condition is  $q = 0$ , such that equation (4.46) may be rewritten as

$$\begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix} \begin{bmatrix} 0 \\ u \end{bmatrix} - \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} q \\ 0 \end{bmatrix} = -\mu^2 \begin{bmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix} \begin{bmatrix} 0 \\ u \end{bmatrix} \quad (4.47)$$

or

$$\begin{aligned} H_{12}u - G_{11}q &= -\mu^2 S_{12}u \\ H_{22}u - G_{21}q &= -\mu^2 S_{22}u \end{aligned} \quad (4.48)$$

Eliminating  $q$  between equations (4.48) gives

$$(H_{22} - G_{21}G_{11}^{-1}H_{12})u = -\mu^2(S_{22} - G_{21}G_{11}^{-1}S_{12})u \quad (4.49)$$

or

$$Ku = \mu^2 Mu \quad (4.50)$$

where  $K$  and  $M$  represent stiffness and mass matrices, respectively, and are defined from equation (4.49). Equation (4.50) represents a generalized algebraic eigenvalue/eigenvector problem, the solution of which can be obtained directly by a variant of the subspace iteration scheme.

A method proposed by Nardini and Brebbia [7-9], in the context of elasticity problems, was to reduce the generalized eigenvalue problem (4.50) to a standard one by inversion of matrix  $K$ , obtaining  $Au = \lambda u$  with  $A = K^{-1}M$  and  $\lambda = 1/\mu^2$ . Matrix  $A$  was then transformed into three-diagonal form by the Householder algorithm, and the eigenvalues and eigenvectors of the transformed matrix were found by the  $Q - R$  algorithm [10].

More recently, another alternative approach was presented by Ali *et al.* [11] which avoids the inversion of matrix  $F$ . The idea is to pose the eigenvalue problem in terms of  $\Phi = F^{-1}u$  rather than  $u$ , which leaves the eigenvalues unchanged. A new Lanczos eigenvalue extraction procedure, based on the Lanczos algorithm [12], was employed in this formulation which will be discussed in more detail in a later section.

All the above solution procedures may present difficulties since matrix  $A$  is non-symmetric and as a consequence, some of the eigenvalues will be complex. Nardini and Brebbia [7-9] found that the complex eigenvalues, if present, appear in the higher modes, which are normally of less importance. In any case, problems with higher-order vibration modes are inevitable when representing a continuum problem by a finite number of degrees of freedom, as will be seen from the results presented in the next section.

### 4.3.2 DRM Results for Vibrating Beam

In order to allow the reader to solve eigenvalue problems using the Boundary Element Method with the programs given in this book, without recurrence to special algorithms, a simple alternative will be given. This solution method is very similar to that described in section 4.1.

In this procedure, instead of obtaining the first  $n$  eigenvalues and associated eigenvectors simultaneously, as is usually done when solving the standard eigenvalue equation (4.50), each is obtained separately by incrementing  $\mu$  in small steps from an initial value. At each step, equation (4.40) is solved using the procedure given in section 4.1, multiplying the right-hand side of (4.45) by a different value of  $\mu^2$ . If  $\mu$  is not near a natural frequency, the displacements obtained at the nodes are small. As  $\mu$  is increased the maximum values of displacements occur as the natural frequencies are approached. This situation may be better understood considering figure 4.3.

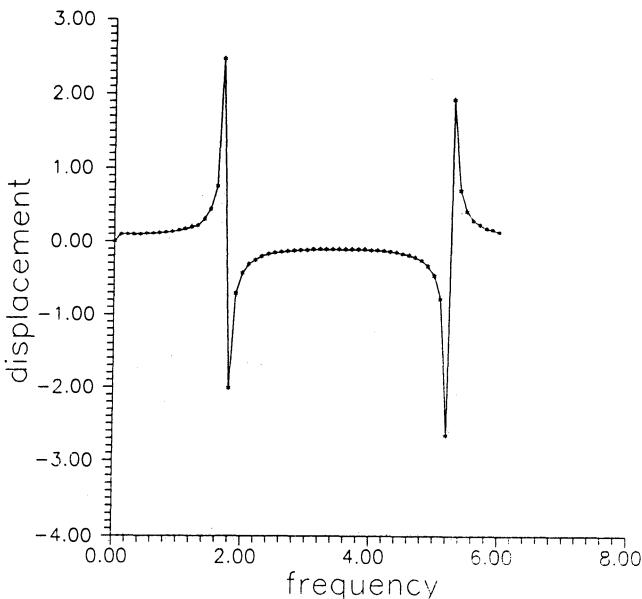


Figure 4.3: Curve  $u \times \mu$  for Tip of Vibrating Beam

Thus, considering equation (4.45) (which is the same as (4.10) with the right-hand side multiplied by the constant  $\mu^2$ ), computer results may be obtained altering

Program 3, with IDDX, IDDY and IFUNC all set to zero, such that the equation is solved on a step-by-step basis, incrementing the value of  $\mu$  at each step. The use of the parameters IDDX, IDDY and IFUNC in Program 3 will be explained with an example in section 4.5.5. Matrices  $\mathbf{H}$ ,  $\mathbf{G}$  and  $\mathbf{S}$  are all unaltered by the increments of  $\mu$  and may be calculated once and stored. If desired, variable increments of  $\mu$  may be used.

If the homogeneous boundary conditions  $u = 0$  on  $\Gamma_1$ ,  $q = 0$  on  $\Gamma_2$  are imposed, only the trivial solution  $u = q = 0$  everywhere will be obtained. To avoid this, a small displacement  $u_0$  must be imposed along the boundary  $\Gamma_1$ . The exact value of  $u_0$  is of no importance, and does not affect the results. This is because the *magnitude* of the displacements in any resonance problem is always multiplied by an arbitrary constant. It is the deformed *shape* which is important; the exact magnitudes are only given by a full dynamic analysis, and do not depend on the type and size of excitation.

For the clamped-free beam depicted in figure 4.2 the exact solution for the natural frequencies is given by [4]

$$\mu_m = \frac{\pi}{L} \left( m - \frac{1}{2} \right) \quad (4.51)$$

considering the problem as one-dimensional. The deformed shape is then given by

$$u_m = C \sin \left[ \left( m - \frac{1}{2} \right) \frac{\pi x}{L} \right] \quad (4.52)$$

where  $m$  is an integer which takes values up to the desired order. When using discrete numerical methods, the maximum number of eigenvalues which may be obtained is equal to the number of degrees of freedom of the problem. Accurate results cannot be expected for higher vibration modes; in particular, if using (4.50) where  $\mathbf{K}$  and  $\mathbf{M}$  are non-symmetric, the higher modes will have complex eigenvalues. In the above expressions,  $L$  is the length of the beam and  $C$  an arbitrary constant. From (4.51) the natural frequencies of the beam can be calculated by the expression

$$\omega^2 = \frac{E\mu^2}{\rho} \quad (4.53)$$

where  $\rho$  and  $E$  are material constants. Equation (4.52) gives the deformed shape for a given  $\mu_m$  once  $C$  is defined. In what follows,  $C = 1$ .

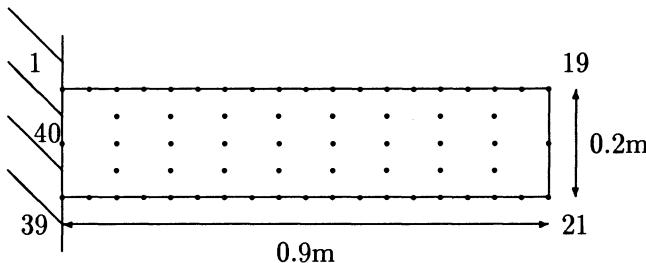


Figure 4.4: BEM Discretization of Beam

Numerical results are now presented for the discretization shown in figure 4.4. The beam is 0.9m long and was discretized using 40 linear boundary elements and 24 internal nodes. A small arbitrary displacement  $u_0$  was imposed at the boundary node at the middle of the clamped side (node 40) for the reasons previously given, while at all the remaining nodes the condition  $q = 0$  was prescribed.

The discontinuity in  $q$  at nodes 1 and 39 may be taken into account by imposing boundary conditions for  $q$  before node 39 and  $q$  after node 1 as described in [13].

Equation (4.45) was solved starting from  $\mu = 0$  and using a variable step on  $\mu$ . The program starts with  $\Delta\mu = 0.1$ ; if for two successive iterations the average increase in the value of  $u$  at all nodes is more than 40%, then  $\Delta\mu$  is decreased to 0.01. When mean values of  $u$  started to drop,  $\Delta\mu$  was increased to 0.5. This value was used until values of  $u$  started to increase again, when  $\Delta\mu$  became 0.1, etc.

Results for the first four natural frequencies of the clamped beam are compared in table 4.5 with the exact solutions calculated from equation (4.51). The same results were obtained using  $u_0$  values of 0.1, 0.5 and 5.0.

$\mu_i$	exact	DRM
1	1.74	1.74
2	5.24	5.24
3	8.73	8.75
4	12.22	12.36

Table 4.5: Results for  $\mu$  for Beam of Figure 4.4

The deformed shape of the beam is plotted in figure 4.5 for each of the first four natural frequencies. For the computer results, all values have been divided by the maximum value obtained at node 19. The exact results were obtained from equation (4.52) with  $C = 1$ .

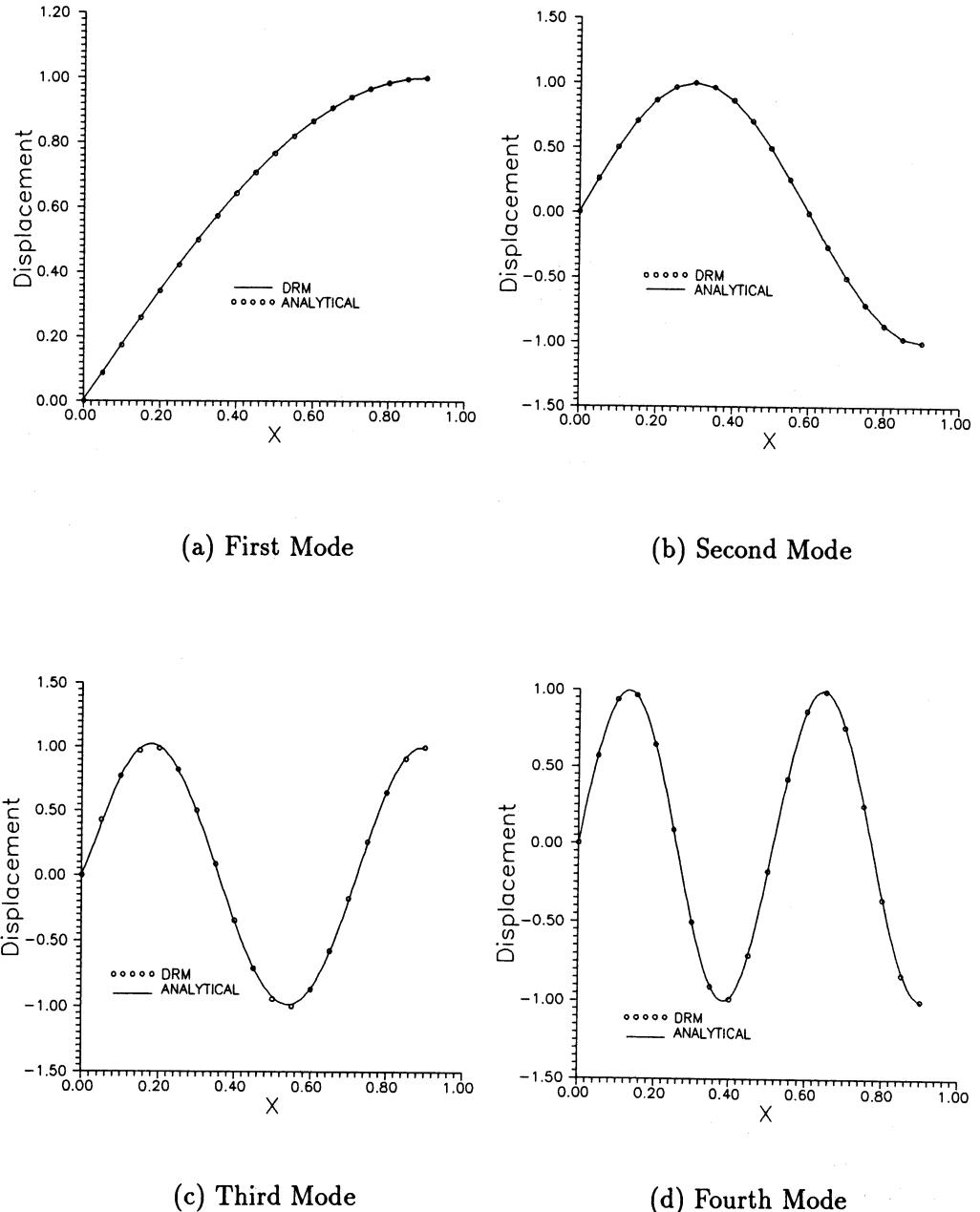


Figure 4.5: Deformed Beam Shape for First Four Natural Frequencies

Results are plotted for boundary nodes 1-19; those at remaining nodes can be obtained by symmetry. Note that the agreement between the DRM results and the exact solution is excellent. Other natural frequencies may be obtained, but this requires the use of a finer discretization. It may be noted in figure 4.5 (d) that the fourth vibration mode contains 7 quarter wavelengths. Employing 19 nodes per side, there are thus  $19/7$  nodes per quarter wavelength for this mode. The discretization has thus reached its limit. If  $\mu_5$  is to be obtained, a finer discretization must be used. For a good representation of the deformed shape it is evident that there should be at least 3 nodes per quarter wavelength.

To test the sensitivity of the method to the number of boundary elements and also to the number of internal nodes, five other discretizations were tested, which are shown in figure 4.6.

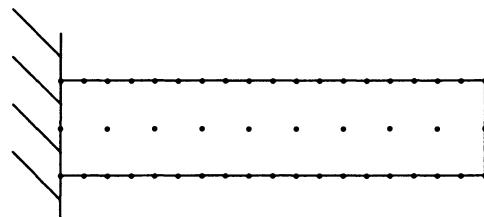


Figure 4.6 (a) 19 nodes/side; 8 internal nodes

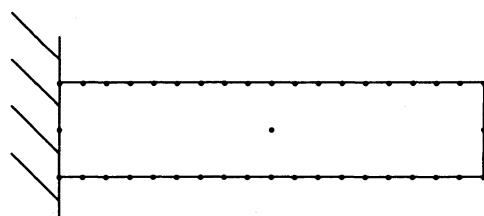


Figure 4.6 (b) 19 nodes/side; 1 internal node

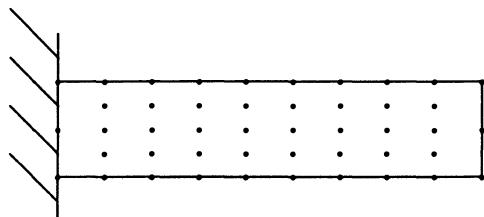


Figure 4.6 (c) 10 nodes/side; 24 internal nodes

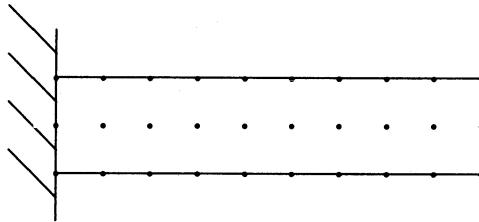


Figure 4.6 (d) 10 nodes/side; 8 internal nodes

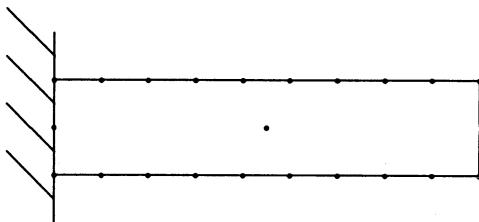


Figure 4.6 (e) 10 nodes/side; 1 internal node

Results for  $\mu$  are shown in table 4.6. Analysing the table, it may be seen that for this type of problem the use of a number of internal nodes is advisable, particularly for higher vibration modes. As expected, the discretization with 10 nodes/side is less accurate for mode 4 as now there are only 10/7 nodes for each quarter wavelength.

$\mu$	N=22			N=40		
	10 nodes/long side			19 nodes/long side		
Exact	L = 1	L = 8	L = 24	L = 1	L = 8	L = 24
1.74	1.75	1.75	1.75	1.74	1.74	1.74
5.24	5.26	5.26	5.26	5.22	5.23	5.24
8.73	9.03	8.88	8.86	8.84	8.76	8.75
12.22	13.30	12.76	12.61	12.77	12.44	12.36

Table 4.6: Results for  $\mu$  for Clamped-Free Beam Using Different Discretizations

Similar results have been obtained by Kontoni *et al.* [14] for the beam of figure 4.4 when subject to different end conditions. In what follows, results are shown for the cases of clamped-clamped and free-free beams for which the respective one-dimensional analytical solutions are of the form [15]:

- Clamped-clamped:

$$\mu_m = \frac{m\pi}{L} \quad (4.54)$$

$$u_m = C \sin \left( \frac{m\pi x}{L} \right) \quad (4.55)$$

- Free-free:

$$\mu_m = \frac{m\pi}{L} \quad (4.56)$$

$$u_m = C \cos \left( \frac{m\pi x}{L} \right) \quad (4.57)$$

$\mu$	N=22			N=40		
	10 nodes/long side			19 nodes/long side		
Exact	L = 1	L = 8	L = 24	L = 1	L = 8	L = 24
3.49	3.49	3.50	3.50	3.48	3.49	3.49
6.98	7.12	7.05	7.05	6.98	6.99	6.99
10.47	11.13	10.79	10.70	10.80	10.58	10.54
13.96	16.09	14.86	14.54	15.14	14.37	14.20

Table 4.7: Results for  $\mu$  for Clamped-Clamped Beam Using Different Discretizations

$\mu$	N=22			N=40		
	10 nodes/long side			19 nodes/long side		
Exact	L = 1	L = 8	L = 24	L = 1	L = 8	L = 24
3.49	3.49	3.50	3.50	3.48	3.48	3.49
6.98	7.08	7.05	7.05	6.99	6.98	6.98
10.47	11.03	10.77	10.71	10.71	10.56	10.54
13.96	15.40	14.84	14.61	14.67	14.35	14.23

Table 4.8: Results for  $\mu$  for Free-Free Beam Using Different Discretizations

In the free-free beam problem, a small arbitrary value  $q_0$  was imposed at node 40 to avoid rigid body motions.

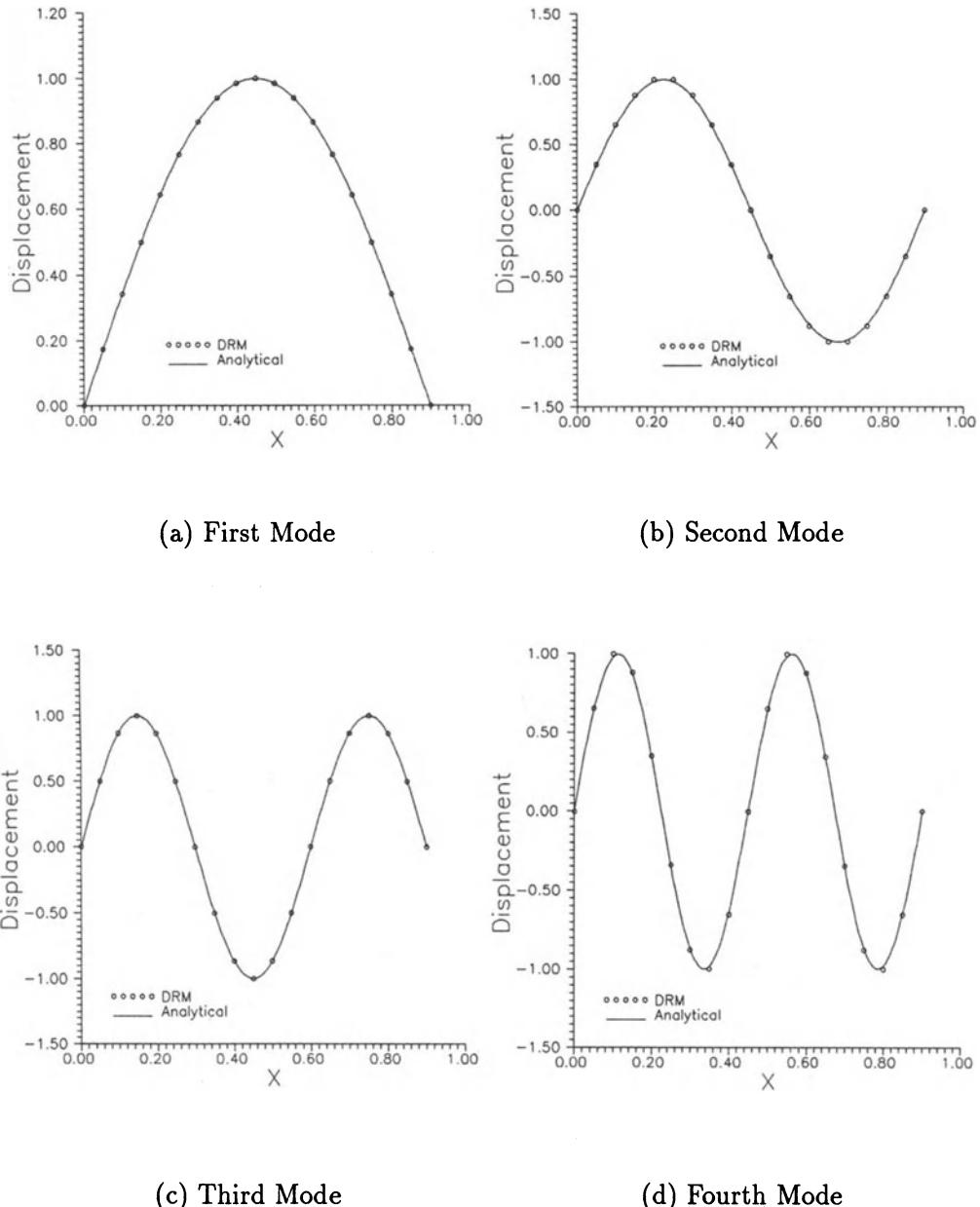


Figure 4.7: First Four Natural Modes for Clamped-Clamped Beam

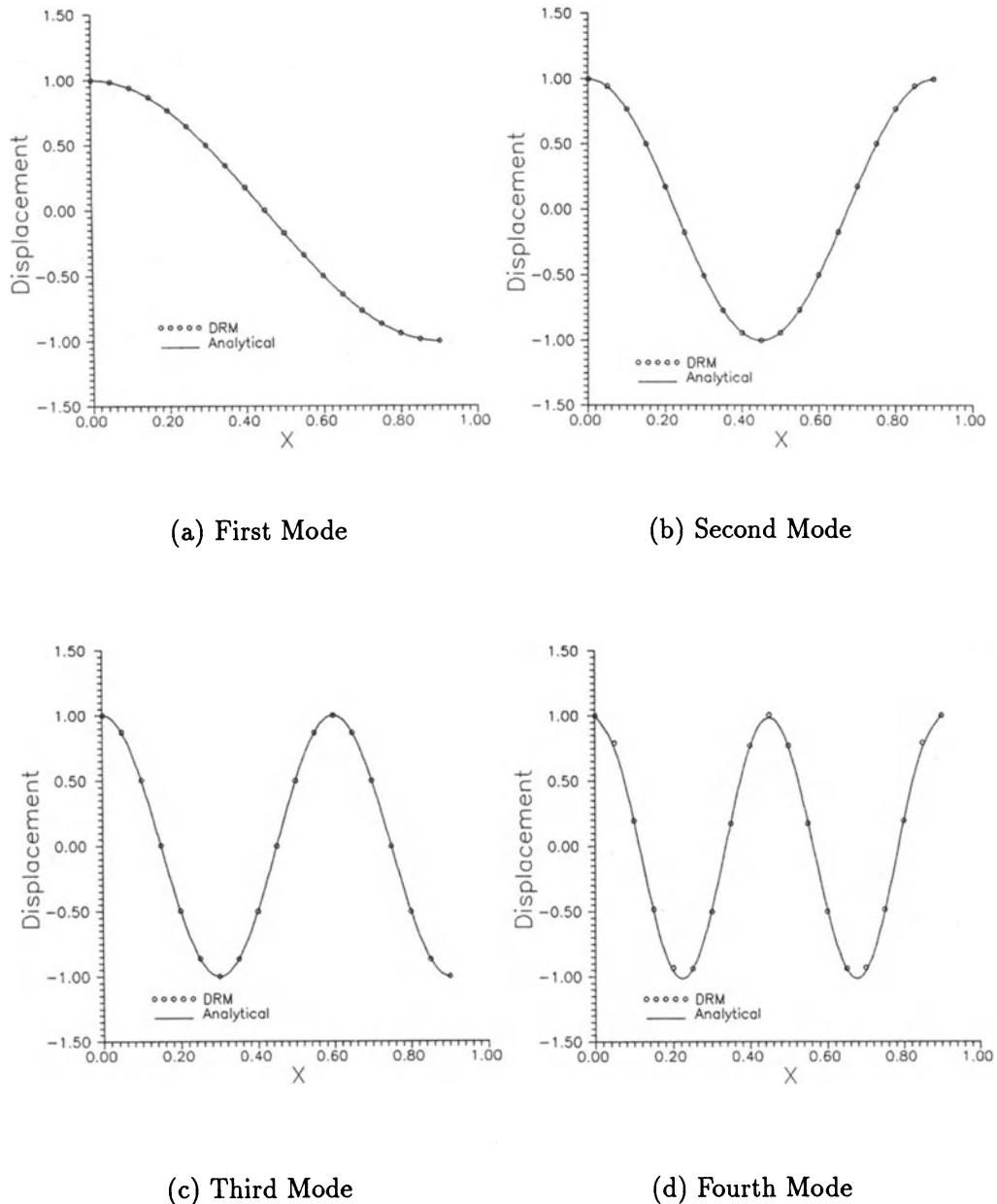


Figure 4.8: First Four Natural Modes for Free-Free Beam

It must be noted that by using the previous formulation the problem of dealing with complex eigenvalues is avoided. Any vibration mode may be obtained if an adequate discretization is used. Extremely fine discretizations must be employed if results for high-order vibration modes are desired, following the rule of approximately three nodes per side for each quarter wavelength for the highest mode to be obtained. This is also true for finite element analysis or any other discrete numerical method. In many cases in engineering practice, high-order vibration modes are of lesser importance.

### 4.3.3 Results for Non-Inversion DRM

The non-inversion DRM formulation developed by Ali *et al.* [11] has been applied to a number of problems with Neumann boundary conditions ( $q = 0$ ). In this case, equation (4.44) reduces to

$$\mathbf{H}\mathbf{u} = -\mu^2(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\Phi \quad (4.58)$$

Substituting for the values of  $\mathbf{u}$  in terms of  $\Phi$ , this becomes

$$\mathbf{H}\mathbf{F}\Phi = -\mu^2(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\Phi \quad (4.59)$$

The eigenvalues of the above equation are the same as that of equation (4.44), the only difference being in that equation (4.59) does not require any matrix inversion. The eigenvectors in terms of  $\mathbf{u}$  can be obtained by the relation  $\mathbf{u} = \mathbf{F}\Phi$ .

In the case of mixed boundary conditions, equation (4.44) has to be partitioned as in (4.47). This condensation process involves the inversion of a small sub-matrix ( $\mathbf{G}_{11}$ ) although the inversion of matrix  $\mathbf{F}$  can still be avoided.

Ali *et al.* [11] analysed several problems with the above formulation, two of which are reproduced in what follows. The first concerns the impedance tube problem, a set-up commonly used in acoustic laboratories for experimentation purposes. The tube length and width were chosen to be 40m and 2m, respectively; the discretization employed 16 linear elements along the length and 2 along the width, with no internal points. The results of the analysis are shown in table 4.9, and appear to be in good agreement with the theoretical solution.

$\mu_i$	theory	DRM
1	0	0
2	4.26	4.25
3	8.53	8.50
4	12.91	12.75
5	17.30	17.00

Table 4.9: Results for Impedance Tube

The second problem reproduced from Ali *et al.* [11] is a trapezoidal model (figure 4.9) for which finite element solutions and experimental results were reported in [16].

The results of table 4.10 intend to show the improvement in the accuracy of the DRM solution with the inclusion of internal points, as discussed in the previous section.

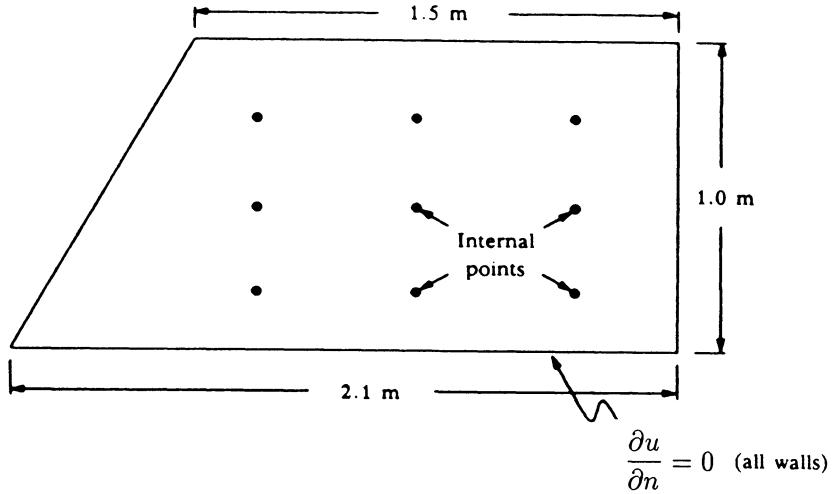


Figure 4.9: Trapezoidal Model

$\mu_i$	DRM (no internal points)	DRM (9 internal points)	Experiment	FEM
1	93.7	92.9	93	92.5
2	169.3	165	164	162.5
3	187.6	182.3	182	179.1

Table 4.10: Results for Trapezoidal Model

## 4.4 Non-Linear Cases

Many different sources of non-linearity exist in continuum mechanics problems, such as those of material, boundary conditions, geometry and governing equations. Material non-linearities occur when the material properties are a function of the problem variable. This type of non-linearity can sometimes be treated by eliminating the domain integrals through a transformation such as Kirchhoff's [17]. Non-linear boundary conditions occur when a boundary flux is a non-linear function of the problem solution. This type of boundary condition involves the inclusion of a special boundary integral in the basic equation (2.19) [18] but does not generate domain integrals, so DRM is

not necessary to deal with it.<sup>1</sup> Geometric non-linear problems occur in the case of large displacements in structural analysis or moving boundary problems such as heat transfer with phase change. Non-linearities in the governing equations are normally handled by using the fundamental solution to a simplified, linear equation. The non-linear terms give rise to domain integrals on the right-hand side of the equation which have to be treated using an iterative scheme. This type of problem is also difficult to treat with other methods of transforming domain integrals to the boundary, but using the DRM the non-linear terms can be easily incorporated into the analysis.

When handling the Helmholtz equation in the previous section, both  $\mu$  and  $u$  were unknown. This problem was solved on a step-by-step basis assuming an initial value of  $\mu$ , calculating the corresponding  $u$  and examining it to see if the assumed  $\mu$  was the value required. This iterative procedure can also be used in some types of non-linear analysis, as will be seen in section 4.4.2.

#### 4.4.1 Burger's Equation

This equation, which has the structure of a convection-diffusion equation but contains the full non-linearity of the one-dimensional flow equation, can be derived by making certain simplifying assumptions to the Navier-Stokes equations. Its usual form, for steady-state situations, is

$$\nabla^2 u + u \frac{\partial u}{\partial x} = 0 \quad (4.60)$$

in which the variable  $u$  is a velocity. To deal with this case with the DRM one notes that  $b$  in equation (4.4) is now

$$b = -u \frac{\partial u}{\partial x} \quad (4.61)$$

such that (4.3) becomes

$$\alpha = -F^{-1} b_1 \quad (4.62)$$

in which vector  $b_1$  contains the values of  $u \partial u / \partial x$  at the nodal points.

The treatment of the derivative as given in section 4.2 produces the following expansion

$$\frac{\partial u}{\partial x} = \frac{\partial F}{\partial x} F^{-1} u \quad (4.63)$$

Equation (4.63) thus expresses the derivative as a product of two matrices and a vector, the result of which is a vector.

It is now necessary to include the function  $u$  which pre-multiplies the derivative term. A simple matrix representation is proposed in the form

---

<sup>1</sup>A problem with non-linear boundary conditions will be considered in chapter 5

$$\mathbf{U} = \begin{bmatrix} u_1 & 0 & 0 & \dots \\ 0 & u_2 & 0 & \\ 0 & 0 & \ddots & \\ \vdots & & & \end{bmatrix} \quad (4.64)$$

By the above definition, matrix  $\mathbf{U}$  is diagonal and contains the values of  $u$  at nodes. Thus,

$$\mathbf{b}_1 = \mathbf{U} \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \mathbf{u} \quad (4.65)$$

This term is a product of three matrices and a vector. Substituting (4.65) into (4.62), and the result in (4.2), one obtains

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = -(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\mathbf{U} \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \mathbf{u} \quad (4.66)$$

Defining matrix  $\mathbf{R}$  such that

$$\mathbf{R} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\mathbf{U} \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \quad (4.67)$$

equation (4.66) becomes

$$(\mathbf{H} + \mathbf{R})\mathbf{u} = \mathbf{G}\mathbf{q} \quad (4.68)$$

which is similar to equation (4.11).

The solution procedure is now iterative since the coefficients of matrix  $\mathbf{R}$  are function of  $u$ . First  $\mathbf{R} = \mathbf{0}$  is assumed and a Laplace equation solved in the form

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{q} \quad (4.69)$$

to obtain a first estimate of  $u$  and  $q$ . A matrix  $\mathbf{U}_1$  is then constructed using (4.64), the suffix indicating the iteration number. The next step is to calculate matrix  $\mathbf{R}_1$  through (4.67) and solve equation (4.68) to obtain  $\mathbf{U}_2$ . This process is continued until convergence is obtained.

Note that only matrix  $\mathbf{U}$  is altered at each iteration; the others may be calculated once, stored and used in each iteration to build first matrix  $\mathbf{R}$  and then equation (4.68). Computer Program 3 may be used with some simple modifications in order to carry out iterative processes like the above.

## Application

In the results given below, the values of  $u$  at internal nodes were considered convergent if their difference between two successive iterations was less than 1%. Equation (4.60) was solved using the discretization shown in figure 4.1. A particular solution for this problem is

$$u = 2/x \quad (4.70)$$

which when imposed as a boundary condition is also the exact solution as explained in section 4.1, expression (4.13). Expression (4.70) has a singularity at  $x = 0$ ; to avoid it, the origin of the cartesian system in figure 4.1 was displaced to the point (-3,0). The expansions  $f = 1 + r$  and  $f = 1 + r + r^2$  were employed, with results shown in table 4.11.

	$x$	$y$	$f = 1 + r$	$f = 1 + r + r^2$	Eqn. (4.70)
$u_{17}$	4.5	0.0	0.445	0.446	0.444
$u_{18}$	4.2	-0.35	0.477	0.481	0.476
$u_{19}$	3.6	-0.45	0.558	0.563	0.555
$u_{20}$	3.0	-0.45	0.669	0.676	0.666
$u_{21}$	2.4	-0.45	0.834	0.841	0.833
$u_{22}$	1.8	-0.35	1.110	1.113	1.111
$u_{23}$	1.5	0.0	1.333	1.331	1.333
$u_{29}$	3.9	0.0	0.514	0.519	0.512
$u_{30}$	3.3	0.0	0.608	0.614	0.606
$u_{31}$	3.0	0.0	0.669	0.675	0.666
$u_{32}$	2.7	0.0	0.742	0.749	0.740
$u_{33}$	2.1	0.0	0.949	0.956	0.952

Table 4.11: DRM Results for Burger's Equation

Convergence was obtained for both  $f$  expansions in four iterations without using relaxation techniques. Both solutions can be seen to be excellent.

### 4.4.2 Spontaneous Ignition: The Steady-State Case

We have already seen an example in which an unknown parameter appears in the equation, and another in which a non-homogeneous term depends on the problem variable. Both of these situations occur in the problem of spontaneous ignition, which will be examined next as an example of the simplicity of treating relatively complex non-linear equations with the DRM.

## The Governing Equations

The problem of spontaneous ignition is a heat conduction problem involving a reactive solid. The partial differential equation describing the problem is a transient diffusion equation with a nonlinear reaction-heating term [19-21].

The differential equation for heat diffusion is usually written as

$$K\nabla^2T + F = \rho c \frac{\partial T}{\partial t} \quad (4.71)$$

for isotropic materials, in which  $T$  is temperature,  $F$  is a source term and  $\rho, c, K$  are density, specific heat and thermal conductivity, respectively. In the case of a reactive solid the source term  $F$  becomes [20]

$$F = \rho Q z e^{-E/RT} \quad (4.72)$$

where  $Q$  is the heat of decomposition of the solid,  $z$  the collision number,  $E$  is the Arrhenius activation energy and  $R$  the universal gas constant. Equation (4.71) can be simplified by a change of variables. Defining [20]

$$u = E \frac{(T - T_a)}{RT_a^2} \quad (4.73)$$

where  $T_a$  is the ambient temperature, the equation in terms of the new variable becomes,

$$\nabla^2 u + \gamma e^u = \frac{1}{k} \frac{\partial u}{\partial t} \quad (4.74)$$

where  $k = K/\rho c$  is the thermal diffusivity and

$$\gamma = \frac{\rho Q E z}{K R T_a^2} e^{-E/RT_a} \quad (4.75)$$

In order to obtain (4.74) the Frank-Kamenetskii approximation [20] was employed, i.e.

$$e^{-E/RT} = e^{-E/RT_a} e^u \quad (4.76)$$

It is usual to define the non-dimensional Frank-Kamenetskii parameter as  $\delta = \ell^2 \gamma$ , where  $\ell$  is a characteristic problem dimension. The physical significance of  $\delta$  is discussed next. For this, it is assumed that a given reactive material has an ignition temperature  $T_m$ , the ambient temperature is considered to be  $T_a$  and the initial temperature to be  $T_0$  at time  $t = 0$ . From equation (4.73) the corresponding values of  $u_m$ ,  $u_a$  and  $u_0$  can be obtained.

For  $\delta < \delta_c$  where  $\delta_c$  is a critical value, a reactive solid at temperature  $T_0$  can be placed in an environment at temperature  $T_a$  ( $T_a < T_m$ ), and the temperature field within the solid will change according to equation (4.74) until all points are at a temperature  $T_a \leq T < T_m$  (not uniform), thus achieving a steady-state situation where

ignition does not occur. For  $\delta > \delta_c$ , the temperature within the solid will continue to increase until some point reaches the ignition temperature  $T_m$ .

Thus, to model the process of spontaneous ignition, a knowledge of  $\delta_c$  is essential. This value can be obtained for a given geometric shape considering  $\partial u / \partial t = 0$  in equation (4.74) [21]. The result is

$$\nabla^2 u = -\gamma e^u \quad (4.77)$$

Equation (4.77) will now be solved using boundary elements for a series of common two-dimensional geometric shapes to obtain  $\gamma_c$  for each case. The time-dependent equation (4.74) will be applied in chapter 5 to simulate the full process of spontaneous ignition. Both types of non-linearity referred to at the beginning of the section are present in equation (4.77) as both  $\gamma$  and  $u$  are unknown.

### DRM Formulation for Spontaneous Ignition

The DRM solution may be obtained using the procedures described in chapter 3 and computer Program 2 may be used with some modifications. The matrix equation is

$$Hu - Gq = (H\hat{U} - G\hat{Q})\alpha \quad (4.78)$$

where, in this case,

$$\alpha = -F^{-1}\gamma b_2 \quad (4.79)$$

in which vector  $b_2$  contains the nodal values of  $e^u$ . Equation (4.78) will be solved by iterating on both  $\gamma$  and  $u$ .

It is known [20] that the behaviour of solutions to equation (4.77) is such that above the critical value of  $\gamma_c$  no solution should be expected, and below  $\gamma_c$  a multiplicity of solutions can occur. This situation is shown in figure 4.10.

For  $\gamma < \gamma_c$  the solution has two branches which are called the upper and lower solutions. Consider first the lower solution. The problem is started with  $u = 0$  everywhere and  $\gamma = \Delta\gamma$ . Thus, a new set of values of  $u$  may be calculated iterating on the term  $\gamma e^u$ , as explained in section 4.4.1. The first solution will be called  $u_1$  where the suffix is the iteration number on  $\gamma$ . Then  $\gamma$  is incremented to  $2\Delta\gamma$  and a new set of values  $u_2$  is calculated. Following the lower curve,  $u_2 > u_1$  at all points. The iterations continue until, for a given  $\gamma$ , the process does not converge any longer. Non-convergence is characterized by  $u_m^n > u_{m-1}^{n-1}$  for two successive iterations on  $u$  at fixed  $\gamma$ . Here,  $m$  is the iteration cycle on  $\gamma$  and  $n$  the iteration cycle on  $u$ . The iterative process on  $u$  is as described for the non-linear term in Burger's equation in section 4.4.1, whilst that on  $\gamma$  is as described for  $\mu$  in the Helmholtz equation in section 4.3.2.

Non-convergence implies that  $\gamma > \gamma_c$  (see figure 4.10). Thus,  $\Delta\gamma$  is reduced to  $\Delta\gamma/10$  and the process recommences from the last value of  $\gamma$  for which convergence was obtained. The solution process is continued in a similar way for ever smaller values of  $\Delta\gamma$  until the required degree of accuracy is obtained. The last value at which  $\gamma$  converges is the required  $\gamma_c$  for the geometry considered.

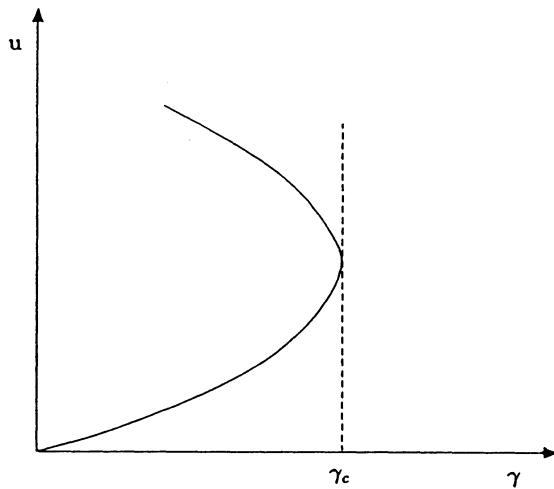


Figure 4.10: Plot of  $\gamma \times u$

If the upper curve is followed, then the solution process starts with  $\gamma = \Delta\gamma$  and  $u = C$  at all nodes, where  $C$  is an arbitrarily large value. As  $\gamma$  increases, at any given node  $u_2 < u_1$ . The resulting value of  $\gamma_c$  will be the same as that obtained following the lower curve.

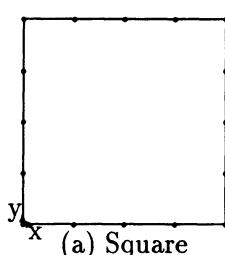
In the DRM iterative process, vector  $\alpha$  in equation (4.78) is a known function calculated, at each iteration, as

$$\alpha = -F^{-1}\gamma_m b_2 \quad (4.80)$$

where  $b_2$  is a vector evaluated using the values of  $u$  obtained at iteration  $n - 1$  for  $\gamma_m$ .

## Applications

Results are now presented for the geometric shapes shown in figure 4.11 with their respective discretizations.



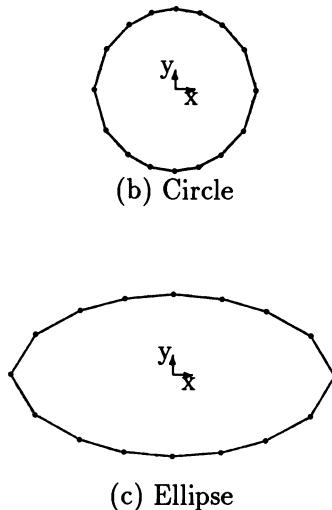


Figure 4.11: 2D Geometry Discretizations for Spontaneous Ignition

The square (figure 4.11(a)) is of unit side, the circle (figure 4.11(b)) is of unit radius and the ellipse (figure 4.11(c)) has a semi-major axis = 2 and a semi-minor axis = 1. Thus, in all cases, the characteristic dimension of the problem  $\ell = 1$ , such that  $\gamma = \delta$ . A set of results was deemed to have converged if two successive iterations produced  $(u^{n-1} - u^n)/(u^{n-1} + u^n) < 0.001$  at all internal nodes.

Numerical results are presented in table 4.12 for different boundary discretizations with fixed numbers of regularly spaced internal nodes (9 for the square, 17 for the circle and the ellipse). It can be seen that, in this case, the use of curved quadratic elements produces little difference in the results if the same number of boundary and internal nodes are used. The reason for this behaviour has to do with the boundary conditions of the problem, which impose a constant value of  $u$  everywhere. For the circle, this produces a constant boundary flux  $q$ . For the ellipse and the square, the flux is variable but its influence on the solution at internal points is of less importance.

A finite element (FEM) solution [21] is also included for comparison. The FEM mesh for each problem consisted respectively of 8, 20 and 34 quadratic isoparametric elements with 37, 75 and 125 node points for half of the region.

shape	32 linear BE	16 quadratic BE	FEM [21]	Exact
square	1.763	1.770	1.703	1.7 [20]
circle	2.032	2.031	2.001	2.0 [19]
ellipse	1.251	1.252	1.234	-

Table 4.12: DRM Results for  $\gamma_c$  for Different Discretizations

The results given in table 4.13 are for the same geometric shapes with a fixed boundary discretization and different numbers of regularly spaced internal nodes. It can be seen that, in increasing the number of internal nodes the solution converges towards the exact value; however, solutions within the usual engineering accuracy may be obtained with a reduced number of internal nodes, with little computer time and data preparation effort.

shape	5 nodes	9 nodes	17 nodes	80 nodes
square	-	1.770	-	1.707
circle	2.080	-	2.031	2.004
ellipse	1.276	-	1.252	1.235

Table 4.13: Results for 16 Quadratic BE for Different Numbers of Internal Nodes

The relatively large number of internal nodes necessary to obtain very accurate solutions is due to the special characteristics of the spontaneous ignition problem, as can be seen from results presented in other sections, where usually only a few such nodes are sufficient to provide the required accuracy.

In all cases  $f = 1 + r$  was employed. Results for the simulation of the full spontaneous ignition process using the time-dependent equation (4.74) are given in chapter 5.

#### 4.4.3 Non-Linear Material Problems

In practical problems of heat transfer, non-linear materials for which the conductivity is a function of temperature are quite common.

It is usual to employ the Kirchhoff transformation, introduced in the BEM context by Bialecki and Nowak [22], Khader and Hanna [23], for this type of problem. This transformation eliminates the domain integrals which would otherwise arise due to the non-linear terms, and linearizes the problem if only Neumann and Dirichlet type boundary conditions are present. However, in the case of a convective or third kind boundary condition, the resulting boundary integral equation is non-linear, requiring an iterative solution procedure [18].

Applied to a non-linear material problem, the DRM will always lead to an iterative solution, however a boundary condition of the third kind can be taken into account in a simple way. A standard computer program may be used. Cases for which the conductivity is a function of either the temperature or spatial location may be considered, thus the formulation is more general than the Kirchhoff transform solution.

The steady-state heat transfer equation, neglecting heat sources, can be written as

$$\frac{\partial}{\partial x} \left( K \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( K \frac{\partial u}{\partial y} \right) = 0 \quad (4.81)$$

The conductivity  $K$  is considered to be a known function of temperature,  $K = K(u)$ .

Three types of boundary conditions will be considered:

$$\begin{aligned} u &= \bar{u} \quad \text{on } \Gamma_1 \\ q &= k \frac{\partial u}{\partial n} = \bar{q} \quad \text{on } \Gamma_2 \\ q &= h(u_f - u) \quad \text{on } \Gamma_3 \end{aligned} \quad (4.82)$$

where  $q$  represents the heat flux,  $h$  is a heat transfer coefficient and  $u_f$  is the temperature of a surrounding medium.

### The Kirchhoff Transformation

The basic idea of the Kirchhoff transformation is to construct a new dependent variable  $U = U(u)$  such that (4.81) becomes linear in the new variable. The derivatives of  $U$  with respect to  $x$  and  $y$  are then

$$\begin{aligned} \frac{\partial U}{\partial x} &= \frac{dU}{du} \frac{\partial u}{\partial x} \\ \frac{\partial U}{\partial y} &= \frac{dU}{du} \frac{\partial u}{\partial y} \end{aligned} \quad (4.83)$$

Comparing the right side of the above expressions with (4.81), one concludes that the proper choice of  $U$  is such that

$$\frac{dU}{du} = K(u) \quad (4.84)$$

or, in integral form,

$$U = T[u] = \int_{u_a}^u K(u) du \quad (4.85)$$

where  $u_a$  is an arbitrary reference value. It follows from (4.81) that

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = 0 \quad (4.86)$$

Boundary conditions of the first and second kind become:

$$\begin{aligned} U &= \bar{U} \\ \frac{\partial U}{\partial n} &= \bar{q} \end{aligned} \quad (4.87)$$

The transformation of  $\bar{u}$  into  $\bar{U}$  on the boundary before solution and the transformation of results for  $U$  back into  $u$  after solution depend on the nature of the function  $K = K(u)$  [18,22]. The former is known as a Kirchhoff transformation and the latter as inverse Kirchhoff transformation.

Equation (4.86) may be handled as described in chapter 2. No iterations are necessary and Program 1 for the Laplace equation may be used in its solution.

## Incorporation of Boundary Conditions of the Third Kind

The incorporation of boundary conditions of the third kind into the analysis requires a modification of the general procedure described in chapter 2. The boundary integral involving the flux in equation (2.40) is now split into a part referring to the boundary  $\Gamma_{1+2}$  and a part referring to  $\Gamma_3$ , as follows:

$$c_i U_i + \int_{\Gamma} U q^* d\Gamma = \int_{\Gamma_{1+2}} q u^* d\Gamma + \int_{\Gamma_3} h u_f u^* d\Gamma - \int_{\Gamma_3} h T^{-1}[U] u^* d\Gamma \quad (4.88)$$

This equation may be discretized in the usual way, however the final term will have to be iterated upon. Program 1 cannot be used to solve (4.88) due to the addition of the new boundary condition.

## The Dual Reciprocity Formulation

The application of the DRM to equation (4.81) requires rewriting the equation in a form similar to (4.4). The non-homogeneous term  $b$  in this case will depend on the nature of the function  $K = K(u)$ . Initially the case

$$K = K_0(1 - \beta u) \quad (4.89)$$

will be considered, in which  $K_0$  and  $\beta$  are material constants.

Substituting (4.89) into (4.81) and simplifying, one finds that there are several different possible expressions for the resulting right-hand side term, *i.e.*:

$$\begin{aligned} \nabla^2 u &= -\frac{\beta}{(1 + \beta u)} \left( \frac{\partial u}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} \right) = b \\ \nabla^2 u &= -\frac{1}{\beta u} \nabla^2 u - \frac{1}{u} \left( \frac{\partial u}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} \right) = b \end{aligned} \quad (4.90)$$

The DRM can be used to take either right-hand side to the boundary, and it will be shown that the results are very similar in both cases. The former equation (4.90) needs the approximation only of first derivatives while the latter needs the approximation of second derivatives.

## The $\mathbf{b}$ Vector using First Derivatives

The DRM approximation to first derivatives is given in section 4.2 as

$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\partial \mathbf{F}}{\partial x} \mathbf{F}^{-1} \mathbf{u} \quad (4.91)$$

with a similar expression holding for the derivative with respect to  $y$ . The  $(N + L)$  values of  $\frac{\partial u}{\partial x}$  may thus be evaluated using the values of  $u$  from the previous iteration. Defining a diagonal matrix  $\mathbf{T}_x$  such that

$$T_x(i, i) = \frac{\beta}{(1 + \beta u_i)} \left\{ \frac{\partial u}{\partial x} \right\}_i \quad (4.92)$$

with a similar definition for  $\mathbf{T}_y$ , then

$$\mathbf{b} = - \left( \mathbf{T}_x \frac{\partial \mathbf{F}}{\partial x} + \mathbf{T}_y \frac{\partial \mathbf{F}}{\partial y} \right) \mathbf{F}^{-1} \mathbf{u} \quad (4.93)$$

It must be stressed that matrices  $\mathbf{T}_x$  and  $\mathbf{T}_y$  have to be updated at each iteration.

Defining

$$\mathbf{S} = -(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}}) \left( \mathbf{T}_x \frac{\partial \mathbf{F}}{\partial x} + \mathbf{T}_y \frac{\partial \mathbf{F}}{\partial y} \right) \mathbf{F}^{-1} \quad (4.94)$$

then the first equation (4.90) becomes

$$(\mathbf{H} - \mathbf{S})\mathbf{u} = \mathbf{G} \left( \frac{\partial \mathbf{u}}{\partial n} \right) \quad (4.95)$$

Equation (4.95) is reordered according to the known boundary values and solved iteratively.

### The $\mathbf{b}$ Vector using Second Derivatives

The DRM approximation to the vector  $\mathbf{b}$  will now be derived for the second equation (4.90), which involves second derivatives:

$$\nabla^2 u = -\frac{1}{\beta u} \nabla^2 u - \frac{1}{u} \left( \frac{\partial u}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} \right) \quad (4.96)$$

It should be noted that (4.96) is only valid for  $\beta \neq 0$ .

A DRM approximation to second derivatives is obtained differentiating (4.20):

$$\frac{\partial^2 \mathbf{u}}{\partial x^2} = \frac{\partial^2 \mathbf{F}}{\partial x^2} \boldsymbol{\gamma} \quad (4.97)$$

with

$$\boldsymbol{\gamma} = \mathbf{F}^{-1} \mathbf{u} \quad (4.98)$$

such that

$$\frac{\partial^2 \mathbf{u}}{\partial x^2} = \frac{\partial^2 \mathbf{F}}{\partial x^2} \mathbf{F}^{-1} \mathbf{u} \quad (4.99)$$

A similar expression may be written for the second derivative with respect to  $y$ .

It should be noted that in this case  $f = 1 + r$  proves to be an inadequate approximating function, as the second-order space derivatives of  $r$  are singular when  $r = 0$ . To overcome this, the expansion

$$f = 1 + r^2 + r^3 \quad (4.100)$$

was adopted.

One can now write

$$\mathbf{b} = - \left[ \mathbf{U} \left( \frac{\partial^2 \mathbf{F}}{\partial x^2} + \frac{\partial^2 \mathbf{F}}{\partial y^2} \right) \mathbf{F}^{-1} + \left( \mathbf{U}_x \frac{\partial \mathbf{F}}{\partial x} + \mathbf{U}_y \frac{\partial \mathbf{F}}{\partial y} \right) \mathbf{F}^{-1} \right] \mathbf{u} \quad (4.101)$$

where  $\mathbf{U}$  is a diagonal matrix containing nodal values of  $1/(\beta u)$ ,  $\mathbf{U}_x$  is a diagonal matrix containing nodal values of  $\frac{1}{u} \frac{\partial u}{\partial x}$  and similarly for  $\mathbf{U}_y$ . All of these are obtained using the values of  $u$  at nodes from the previous iteration.

With an appropriate definition of the  $\mathbf{S}$  matrix, the final equation is the same as (4.95).

### Inclusion of Convective Boundary Conditions

Equation (4.95) is reordered before solving, given that at each boundary node either  $u$  or  $\frac{\partial u}{\partial n} = \frac{q}{K}$  is known. In the case of a boundary condition of the third kind

$$q = h(u_f - u) \quad (4.102)$$

such that

$$\frac{\partial u}{\partial n} = \frac{h}{K} (u_f - u) \quad (4.103)$$

equation (4.95) becomes

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{e} + \mathbf{GEu} = \mathbf{Su} \quad (4.104)$$

In the above equation,  $\mathbf{e}$  is a vector containing nodal values of  $hu_f/K$  and  $\mathbf{E}$  is a diagonal matrix containing nodal values of  $h/K$ . The value of  $K = (1 + \beta u)$  is evaluated using values of  $u$  from the previous iteration. Equation (4.104) is only used on  $\Gamma_3$  while equation (4.95) is used on other parts of the boundary.

For the first iteration, terms in  $\beta$  are ignored, and the linear solution is obtained. After that, values of  $q$  are obtained on  $\Gamma_3$  using expression (4.102).

As the application of the DRM is similar in different cases, the above may be implemented starting from Program 3, given in this chapter.

### Problem Including Boundary Conditions of the First and Second Kinds

The geometry of the problem considered is shown in figure 4.12, consisting of a square plate of unit side. Nine equal constant boundary elements were used to discretize each side. The boundary conditions employed were  $u = 300$  at  $x = 0$ ,  $u = 400$  at  $x = 1$  and  $q = 0$  at  $y = 0$  and  $y = 1$ .

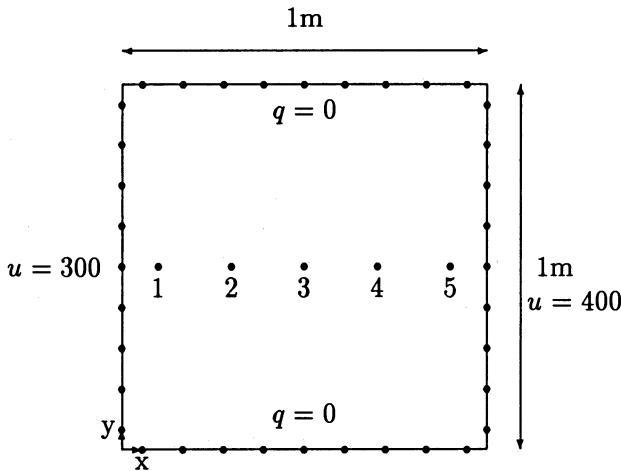


Figure 4.12: Problem 1: Square Plate

Equation (4.81) was solved with

$$K = K_0 \left[ 1 + \beta \left( \frac{u - u_0}{u_0} \right) \right] \quad (4.105)$$

with  $K_0 = 1$ ,  $\beta = 3$  and  $u_0 = 300$ .

The expressions for the Kirchhoff transformation and inverse transformation were taken from [18] for this situation,

$$U = \frac{K_0 u_0}{2\beta} \left[ 1 + \beta \left( \frac{u - u_0}{u_0} \right) \right]^2 - \frac{K_0 u_0}{2\beta} \quad (4.106)$$

such that  $\bar{U} = 0$  for  $\bar{u} = 300$  and  $\bar{U} = 150$  for  $\bar{u} = 400$ , and

$$u = u_0 + \frac{u_0}{\beta} \left( 1 + \frac{2\beta U}{K_0 u_0} \right)^{\frac{1}{2}} - \frac{u_0}{\beta} \quad (4.107)$$

In the case of the DRM solution

$$b = -\frac{\beta}{u_0 + \beta(u - u_0)} \left( \frac{\partial u}{\partial x} \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} \frac{\partial u}{\partial y} \right) \quad (4.108)$$

Results are given in table 4.14 at the 5 nodes numbered in figure 4.12. The DRM solution converged in 4 iterations. For this type of non-linearity, the difference between the results obtained and the Laplace solution ( $\beta = 0$ ) is small.

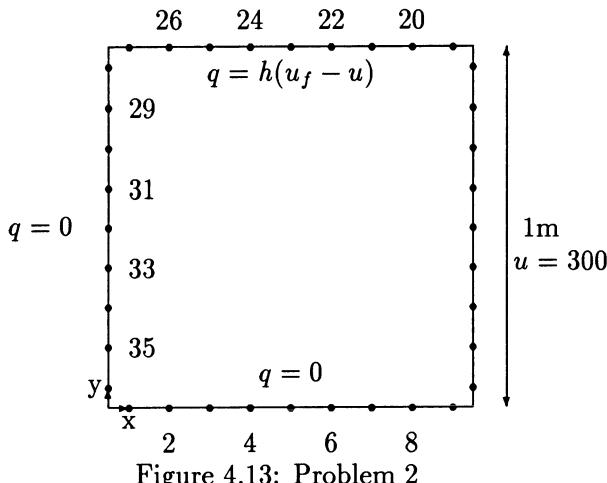
The DRM solution is seen to be in reasonable agreement with that obtained using the Kirchhoff transformation.

Node	$x$	$y$	DRM $\beta = 3$	Kirchhoff Transform	Laplace $(\beta = 0)$
1	.1	.5	314.15	314.00	310
2	.3	.5	338.34	337.82	330
3	.5	.5	358.49	358.11	350
4	.7	.5	376.27	376.08	370
5	.9	.5	392.43	392.36	390

Table 4.14: Results for  $u$  for Problem 1

### Results for a Problem with a Boundary Condition of the Third Kind

The problem shown in figure 4.13 was published in [22] where the Kirchhoff transformation solution was originally presented. The geometry and boundary discretization are the same as for the previous example, however in this case the boundary conditions are  $q = 0$  at  $x = 0$  and  $y = 0$ ,  $u = 300$  at  $x = 1$ , and the convective boundary condition at  $y = 1$ , i.e.  $q = h(u_f - u)$  with  $h = 10$  and  $u_f = 500$ .



A linear variation for the conductivity-temperature relation was also adopted for this problem,  $K = K_0(1 + \beta u)$ , with  $k_0 = 1$  and  $\beta = 0.3$ .

The DRM solutions used 25 evenly spaced internal nodes, and were carried out for each of the two equations (4.90). The solution using first derivatives was labelled DRM1 and that using second derivatives, DRM2. Results for  $u$  for the 12 nodes shown in figure 4.13 are given in table 4.15.

The DRM results converged in 5 iterations for a tolerance of 0.00001. It can be seen that the agreement between both DRM results and the Kirchhoff transformation

Node	DRM1 $\beta = 0.3$	DRM2 $\beta = 0.3$	Kirchhoff Transform	Laplace ( $\beta = 0$ )
2	307.13	306.66	306.79	385.12
4	306.04	305.70	305.79	372.98
6	304.26	304.03	304.08	351.83
8	301.94	301.86	301.84	323.48
20	307.05	306.95	306.93	442.50
22	311.84	311.61	311.61	469.57
24	314.64	314.34	314.33	478.68
26	316.11	315.78	315.71	482.30
29	313.53	313.15	313.14	455.84
31	310.55	310.17	310.22	424.88
33	308.60	308.19	308.28	402.82
35	307.59	307.07	307.25	390.57

Table 4.15: Results for  $u$  for Problem 2

solution is excellent, particularly in the case of DRM2 which uses second derivatives.

## 4.5 Computer Program 3

This program solves problems of the type  $\nabla^2 u = b(x, y, u)$  for a series of different  $b$  functions. For time-dependent problems, computer Program 4 should be used.

Different  $f$  expansions may be used, as long as the constant and one other term are included. Both equation and  $f$  expansion are defined as data input as will be seen in section 4.5.1. Linear elements are again used for simplicity. The same limitation of 200 nodes applies as for Programs 1 and 2. The reader may modify the program to deal with other equations,  $f$  expansions or elements without difficulty. Input and output for a test problem are presented at the end of the section. The program is written in modular form, using some of the routines described for Programs 1 and 2, its structure being similar to both. Program 3 is again dimensioned for a maximum of 200 nodes, and its flow chart is depicted in figure 4.14.

The program modules:

- ASSEM2
- SOLVER
- INPUT2
- OUTPUT

have been described and listed in chapters 2 and 3 and will be used in the form already given. In this section the following new routines are presented:

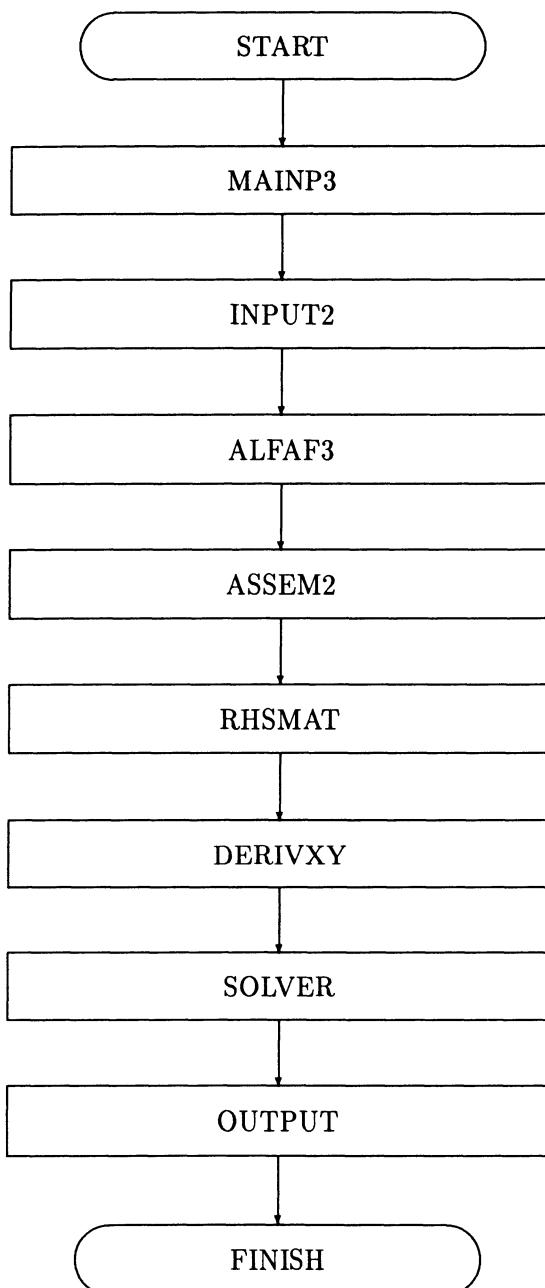


Figure 4.14: Modular Structure for Program 3

- **MAINP3**
- **ALFAF3**
- **RHSMAT**
- **DERIVXY**

To construct Program 3, start from Program 2 and carry out the following operations:

- i. Substitute **MAINP3** for **MAINP2**
- ii. Substitute **ALFAF3** for **ALFAF2**
- iii. Substitute **RHSMAT** for **RHSVEC**
- iv. Remove **INTERM**
- v. Include **DERIVXY**

The variable names used in the program are the same as those used in Programs 1 and 2. Where new variables are introduced, these will have symbols as close as possible to those used in the text. At the end of the section, results and data for a test problem are given.

#### 4.5.1 MAINP3

The first step in the program is the definition of the equation to be modelled. The available options are shown in table 4.16.

IDDX	IDDY	IFUNC	Equation
0	0	0	$\nabla^2 u = -u$
1	0	0	$\nabla^2 u = -\partial u / \partial x$
0	1	0	$\nabla^2 u = -\partial u / \partial y$
1	1	0	$\nabla^2 u = -(\partial u / \partial x + \partial u / \partial y)$
0	0	1	not used
1	0	1	$\nabla^2 u = -\psi(x, y) \partial u / \partial x$
0	1	1	$\nabla^2 u = -\psi(x, y) \partial u / \partial y$
1	1	1	$\nabla^2 u = -\psi(x, y) (\partial u / \partial x + \partial u / \partial y)$

Table 4.16: Options Available in Program 3

Each option is accessed defining the three parameters indicated in table 4.16 in the first line of the data file. The parameters are defined in the order IDDX, IDDY, IFUNC. If an option involving  $\psi(x, y) \neq 1$  is to be used, the relevant function must be specified in the routine **DERIVXY** listed in section 4.5.4.

The next step is the definition of the  $f$  expansion to be employed. The expansion used is

$$f = C_1 + C_2r + C_3r^2 + C_4r^3 \quad (4.109)$$

The constant  $C_1$  is necessarily different from 0; in addition, one of the others must also be different from zero. In the example given,  $C_1 = C_2 = 1, C_3 = C_4 = 0$ . The use of  $C_i$  different from 1 or zero will produce a scaling effect, which may be important for very large or very small geometries. The reader may thus test a large number of  $f$  expansions without altering the program. The  $C_i$  are specified in the second line of the data file.

The remaining data is as for Program 2, and INPUT2 is used to read it. Equation option and  $f$  expansion are printed as the first output of the program.

The next step is to call routine ALFAF3 to calculate  $\mathbf{F}^{-1}$ . For the equations considered in this chapter,  $b$  is an unknown function, and thus  $\alpha$  is not known explicitly. In this case, vector  $\alpha$  is substituted by the product  $\mathbf{F}^{-1}\mathbf{b}$ .

Subroutine ASSEM2 is called as before to evaluate and store matrices  $\mathbf{H}$  and  $\mathbf{G}$ , apply the boundary conditions, and set up the system  $\mathbf{Ax} = \mathbf{y}$ .

The routine RHSMAT, which replaces RHSVEC for unknown  $b$  functions, produces now a matrix  $S$  instead of vector  $D$ . Next, boundary conditions are applied to  $S$ . In the case of a known  $u$ , the corresponding coefficients of  $S$  are multiplied by its value and the result added to  $XY$ . If  $u$  is unknown, the coefficients of  $S$  are subtracted from the corresponding coefficients of  $A$ , thus  $\mathbf{Ax} = \mathbf{y}$  will include the contribution of the source term.  $XY$  starts as  $y$  and becomes  $x$  after calling SOLVER. Results are distributed between  $U$  and  $Q$  according to the boundary condition type.

In this program, there is no need to call INTERM as the boundary and internal solutions are coupled as explained in section 4.1. Both solutions will be obtained simultaneously in SOLVER. This represents a simplification of the program structure.

The final step is a call to routine OUTPUT which prints the results.

```

COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),P(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/XY(200),A(200,200)
COMMON/FACTOR/C1,C2,C3,C4,IDDX,IDDY,IFUNC
COMMON/TWO/S(200,200),FINV(200,200)
DOUBLE PRECISION XY,A,AM,CD,ALM
DOUBLE PRECISION X,Y,FINV
REAL LE,LJ
INTEGER CON
C
C  MAINP3
C
C  DEFINE EQUATION TO BE MODELLED
C
C  OPTIONS WITH IFUNC = 0
C
C  IDDX = 0 AND IDDY = 0: NABLA2U=-U
C  IDDX = 1 AND IDDY = 0: NABLA2U=-DUDX
C  IDDX = 0 AND IDDY = 1: NABLA2U=-DUDY
C  IDDX = 1 AND IDDY = 1: NABLA2U=-(DUDX+DUDY)
C

```

```

C OPTIONS WITH IFUNC = 1
C
C IDDX = 1 AND IDDY = 0: NABLA2U=-PSI(X,Y)DUDX
C IDDY = 0 AND IDDX = 1: NABLA2U=-PSI(X,Y)DUDY
C IDDX = 1 AND IDDY = 1: NABLA2U=-PSI(X,Y)(DUDX+DUDY)
C
      READ(5,1065) IDDX, IDDY, IFUNC
1065   FORMAT(3I3)
C
C WRITE OPTION CHOSEN
C
      IF(IFUNC.EQ.0)THEN
        IF(IDDY.EQ.0)THEN
          IF(IDDX.EQ.0)THEN
            WRITE(6,1061)
1061      FORMAT('' EQUATION MODELLED: NABLA2U=-U '')'
            ELSE
              WRITE(6,1060)
1060      FORMAT('' EQUATION MODELLED: NABLA2U=-DUDX '')'
              END IF
            ELSE
              IF(IDDX.EQ.0)THEN
                WRITE(6,1058)
1058      FORMAT('' EQUATION MODELLED: NABLA2U=-DUDY '')'
                ELSE
                  WRITE(6,1057)
1057      FORMAT('' EQUATION MODELLED: NABLA2U=-(DUDX+DUDY) '')'
                  END IF
                END IF
              ELSE
                IF(IDDY.EQ.0)THEN
                  IF(IDDX.EQ.0)THEN
                    WRITE(6,1061)
                    ELSE
                      WRITE(6,1160)
1160      FORMAT('' EQUATION MODELLED: NABLA2U=-F(X,Y)DUDX '')'
                      END IF
                    ELSE
                      IF(IDDX.EQ.0)THEN
                        WRITE(6,1158)
1158      FORMAT('' EQUATION MODELLED: NABLA2U=-F(X,Y)DUDY '')'
                        ELSE
                          WRITE(6,1157)
1157      FORMAT('' EQUATION MODELLED:
C                         NABLA2U=-F(X,Y)(DUDX+DUDY) '')'
                          END IF
                        END IF
                      END IF
                    END IF
                  END IF
                END IF
              END IF
            END IF
          END IF
        END IF
      END IF
    END IF
  END IF
C
C DEFINE F EXPANSION TO BE USED:
C
C F EXPANSION WILL BE F= C1+C2*R+C3*R2+C4*R3
C

```

```

      READ(5,2005)C1,C2,C3,C4
2005  FORMAT(4F3.1)
C
C  WRITE F EXPANSION IN USE
C
      WRITE(6,2006)
2006  FORMAT(//' F EXPANSION: '//)
      WRITE(6,2007)C1,C2,C3,C4
2007  FORMAT(' F = ',F2.0,' + ',F2.0,'*R + ',F2.0,
1 '*R2 + ',F2.0,'*R3 '//)
C
C DATA INPUT
C
C IF AN OPTION WITH IFUNC = 1 IS IN USE
C F(X,Y) MUST BE DEFINED IN ONE LINE OF ROUTINE DERIVXY
C
      CALL INPUT2
C
C CALCULATION OF F-1
C
      CALL ALFAF3
C
C CALCULATION OF G AND H MATRICES; APPLY BOUNDARY CONDITIONS
C RESULTING MATRIX IN A; RESULTING VECTOR IN XY
C
      CALL ASSEM2
C
C CALCULATION OF MATRIX S
C
      CALL RHSMAT
C
C DISTRIBUTE ELEMENTS OF S ACCORDING TO BOUNDARY CONDITIONS
C
      DO 5 I = 1,NN+L
      DO 5 J = 1,NN+L
      KK=KODE(J)
      IF(KK.EQ.0.OR.KK.EQ.2)THEN
      A(I,J) = A(I,J)-S(I,J)
      ELSE
      XY(I) = XY(I)+S(I,J)*U(J)
      END IF
5       CONTINUE
C
C SOLVE FOR BOUNDARY AND INTERNAL VALUES
C
      NH=NN
      NN=NN+L
      CALL SOLVER
      NN=NH
C
C PUT VALUES IN APPROPRIATE ARRAY; U OR Q
C
      DO 76 J1=1,NN+L

```

```

      KK=KODE(J1)
      IF(KK.EQ.0.OR.KK.EQ.2)THEN
        U(J1) = XY(J1)
      ELSE
        Q(J1) = XY(J1)
      END IF
76      CONTINUE
C
C WRITE RESULTS
C
      CALL OUTPUT
      STOP
      END

```

#### 4.5.2 Subroutine ALFAF3

In chapter 3 the routine ALFAF2 was described to calculate  $\alpha = F^{-1}b$ . For the type of equations now under consideration,  $b$  is not a known function and so vector  $\alpha$  cannot be calculated explicitly. The present routine calculates matrix  $F^{-1}$ .

Matrix  $F$  is initially calculated and then inverted using Gauss elimination. Vector  $XY$  used for the inversion process is each column of an identity matrix in turn. This same vector returned by SOLVER will be the corresponding column of the matrix  $F^{-1}$  which is then stored in FINV.

```

SUBROUTINE ALFAF3
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),P(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/XY(200),A(200,200)
COMMON/FACTOR/C1,C2,C3,C4,IDDX,IDDY,IFUNC
COMMON/TWO/S(200,200),FINV(200,200)
DIMENSION F(200,200)
DOUBLE PRECISION XY,A,AM,CD,ALM
DOUBLE PRECISION X,Y,FINV,F,R,XJ,YJ,XI,YI
REAL LE,LJ
INTEGER CON
NH = NN
NN = NN + L
C
C SET UP THE F MATRIX
C
DO 5 I = 1,NN
  XI = X(I)
  YI = Y(I)
  DO 2 J = 1,NN
    XJ = X(J)
    YJ = Y(J)
    R = SQRT((XI-XJ)**2 + (YI-YJ)**2)
    F(I,J) = C1 + C2*R + C3*R*R + C4*R*R*R
2      CONTINUE
5      CONTINUE
C

```

```

C CALCULATE THE F-1 MATRIX IN FINV
C
      DO 9 IT = 1,NN
      DO 10 I=1,NN
         XY(I) = 0.
         DO 10 J=1,NN
            A(I,J) = F(I,J)
10      XY(IT) = 1.
      CALL SOLVER
      DO 11 I = 1,NN
11      FINV(I,IT) = XY(I)
9       CONTINUE
C
C CLEAR A AND XY FOR LATER USE
C
      DO 16 I = 1,NN
         XY(I) = 0.
         DO 16 J = 1,NN
            A(I,J) = 0.
16      CONTINUE
      NN = NH
      RETURN
      END

```

#### 4.5.3 Subroutine RHSMAT

The structure of **RHSMAT** is very similar to that of **RHSVEC** of Program 2. The changes are due to the fact that the right-hand side of the equation is now a matrix, called **S** in the program and defined as

$$\mathbf{S} = (\mathbf{G}\hat{\mathbf{Q}} - \mathbf{H}\hat{\mathbf{U}})\mathbf{F}^{-1} \quad (4.110)$$

multiplied by an unknown vector **U**. Matrix **S** may be multiplied by other matrices if derivatives are present on the governing equation. The routine is divided into five sections:

1. Calculate  $\hat{\mathbf{Q}}$ .

This matrix is unassembled and is stored in **HAT**.

2. Evaluate  $\mathbf{G}\hat{\mathbf{Q}}$ .

The result is stored in the array **S**.

3. Calculate  $\hat{\mathbf{U}}$ .

This matrix is assembled and stored in **HAT**. Note that both **HAT**=  $\hat{\mathbf{U}}$  and **HH**=  $\mathbf{H}$  are  $(N + L) \times (N + L)$  matrices, as shown in equation (3.34).

4. Evaluate  $\mathbf{H}\hat{\mathbf{U}}$ .

The result is subtracted from S. Note that the terms  $c_i \hat{u}_{ij}$  of equation (3.13) are included in this operation.

5. Transfer S into HH as temporary storage.

$S = (G\hat{Q} - H\hat{U})F^{-1}$  is finally calculated by multiplying S by FINV.

Subroutine DERIVXY is called if there are derivatives present in the source term. In DERIVXY, S will be multiplied by the necessary extra matrices for each case.

```

SUBROUTINE RHSMAT
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),P(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/XY(200),A(200,200)
COMMON/FACTOR/C1,C2,C3,C4,IDDX,IDDY,IFUNC
COMMON/TWO/S(200,200),FINV(200,200)
COMMON/EIGHT/GQHU(200,200)
COMMON/DRM/HH(200,200),GG(200,400)
DIMENSION HAT(200,400)
DOUBLE PRECISION XY,A,AM,CD,ALM
DOUBLE PRECISION X,Y,FINV
REAL LE,LJ
INTEGER CON

DO 1 J1=1,NN+L
    DO 10 J2 = 1,2*NE
        HAT(J2,J1) = 0.
10    CONTINUE
1     CONTINUE
C
C HAT STARTS AS QHAT
C
DO 60 J1 = 1,NE
    N1 = CON(J1,1)
    N2 = CON(J1,2)
    X1 = X(N1)
    X2 = X(N2)
    Y1 = Y(N1)
    Y2 = Y(N2)
    DD = LE(J1)
    JP1 = 2*J1 - 1
    JP2 = 2*J1
    DO 50 J2 = 1,NN+L
        XP = X(J2)
        YP = Y(J2)
        R1 = SQRT((X1-XP)**2 + (Y1-YP)**2)
        R2 = SQRT((X2-XP)**2 + (Y2-YP)**2)
        QHAT1 = (C1*0.5+C2*R1/3.+C3*(R1**2)/4.+C4*(R1**3)/5.)
1        *((X1-XP)*(Y1-Y2)+(Y1-YP)*(X2-X1))/DD
        QHAT2 = (C1*0.5+C2*R2/3.+C3*(R2**2)/4.+C4*(R2**3)/5.)
1        *((X2-XP)*(Y1-Y2)+(Y2-YP)*(X2-X1))/DD
C
C QHAT CALCULATED AT BEGINNING AND END OF EACH ELEMENT
C

```

```

      HAT(JP1,J2) = QHAT1
      HAT(JP2,J2) = QHAT2
50      CONTINUE
60      CONTINUE
C
C PRE-MULTIPLY QHAT BY G AND PUT RESULT IN S
C
80      DO 88 J1 = 1,NN+L
        DO 88 J2 = 1,NN+L
          S(J1,J2)=0.
          DO 80 J3 = 1,2*NE
            S(J1,J2) = S(J1,J2) + GG(J1,J3)*HAT(J3,J2)
80      CONTINUE
88      CONTINUE
C
C HAT NOW BECOMES UHAT
C
150     DO 160 J1=1,NN+L
      XI=X(J1)
      YI=Y(J1)
      DO 150 J2=1,NN+L
        HAT(J1,J2)=0.
        XP = X(J2)
        YP = Y(J2)
        R=SQRT((XI-XP)**2+(YI-YP)**2)
        UHAT = C2*(R**3)/9.+C1*(R**2)/4.+C3*(R**4)/16.+C4*(R**5)/25.
        HAT(J1,J2)=UHAT
150     CONTINUE
160     CONTINUE
C
C PRE-MULTIPLY UHAT BY H AND SUBTRACT FROM S
C
180     DO 181 J1=1,NN+L
        DO 181 J2=1,NN+L
          DO 180 J3=1,NN+L
            S(J1,J2)=S(J1,J2)-HH(J1,J3)*HAT(J3,J2)
180     CONTINUE
181     CONTINUE
C
C TRANSFER MATRIX GQ-HU INTO HH AS TEMPORARY STORAGE
C
170     DO 170 I=1,NN+L
        DO 170 J=1,NN+L
          HH(I,J)=S(I,J)
170     CONTINUE
C
C DO (GQHAT-HUHAT)F-1 AND PUT IN S: FINV IS F-1
C
180     DO 56 I = 1,NN+L
        DO 56 J=1,NN+L
          S(I,J)=0.
          DO 55 J1=1,NN+L
            S(I,J) = S(I,J)+HH(I,J1)*FINV(J1,J)
55      CONTINUE
56      CONTINUE

```

```

55      CONTINUE
56      CONTINUE
C
C SUBROUTINE DERIVXY WILL MODIFY S IF DERIVATIVES ARE PRESENT
C
IF((IDDX+IDDY).NE.0)CALL DERIVXY
RETURN
END

```

#### 4.5.4 Subroutine DERIVXY

This subroutine is called if derivatives are present in the source term. The first stage is to define the derivative matrices in  $\mathbf{F}$

$$\mathbf{F} = \text{IDDX} \times \frac{\partial \mathbf{F}}{\partial x} + \text{IDDY} \times \frac{\partial \mathbf{F}}{\partial y} \quad (4.111)$$

This matrix is skew-symmetric. The result is multiplied by  $\text{FINV}$  and stored in  $\text{CC}$

$$\text{CC} = \left( \text{IDDX} \times \frac{\partial \mathbf{F}}{\partial x} + \text{IDDY} \times \frac{\partial \mathbf{F}}{\partial y} \right) \times \text{FINV} \quad (4.112)$$

The next step is to multiply by  $\psi(x, y)$  if  $\text{IFUNC} \neq 0$

$$\text{CC} = \psi(x, y) \times \text{CC} \quad (4.113)$$

Note that  $\psi(x, y) = -2/x$  is defined in the program, and its use will be explained in the next section. If any other function  $\psi(x, y)$  is to be used, this must be defined in the line indicated by the comment statement. If  $\text{IFUNC} = 0$  operation (4.113) is not carried out, and  $\text{CC}$  continues to be defined by (4.112). Next,  $\mathbf{S}$  is multiplied by  $\text{CC}$  and the result stored in  $\mathbf{F}$ , in such a way that we now have

$$\mathbf{F} = (\mathbf{G}\hat{\mathbf{Q}} - \mathbf{H}\hat{\mathbf{U}})\mathbf{F}^{-1}\Psi \left( \text{IDDX} \frac{\partial \mathbf{F}}{\partial x} + \text{IDDY} \frac{\partial \mathbf{F}}{\partial y} \right) \mathbf{F}^{-1} \quad (4.114)$$

Matrix  $\mathbf{F}$  is returned to  $\mathbf{S}$ , so that  $\mathbf{S}$  is defined by expression (4.110) on returning to **RHSMAT**.

```

SUBROUTINE DERIVXY
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),P(200),LE(200),
1 CON(200,2),KODE(200),Q(200),POEI(4),FDEP(4),ALPHA(200)
COMMON/FIVE/XY(200),A(200,200)
COMMON/FACTOR/C1,C2,C3,C4,IDDX,IDDY,IFUNC
COMMON/TWO/S(200,200),FINV(200,200)
DIMENSION F(200,200),CC(200,200)
DOUBLE PRECISION XY,A,AM,CD,ALM,F,CC
DOUBLE PRECISION X,Y,XJ,YJ,R,XI,YI,PX,PY,FINV
REAL LE,LJ
INTEGER CON
NH = NN
NN = NN+L

```

```

C
C DF/DX, DF/DY OR (DF/DX+DF/DY)
C ACCORDING TO THE INPUT VALUES OF IDDX AND IDDY
C ARE NOW IN F
C
DO 1 I=1,NN
XI = X(I)
YI = Y(I)
DO 2 J = 1,NN
XJ = X(J)
YJ = Y(J)
R = SQRT((XJ-XI)**2 + (YJ-YI)**2)
F(I,J) = 0.
IF(R.EQ.0.) GO TO 2
PX = C2*(XJ-XI)/R + C3*2*(XJ-XI) + C4*3*(XJ-XI)*R
PY = C2*(YJ-YI)/R + C3*2*(YJ-YI) + C4*3*(YJ-YI)*R
IF(IDDX.NE.0) F(I,J) = F(I,J) - PX
IF(IDDY.NE.0) F(I,J) = F(I,J) - PY
2      CONTINUE
1      CONTINUE
C
C MULTIPLY BY F-1
C
DO 4 I = 1,NN
DO 4 J = 1,NN
CC(I,J) = 0.
DO 5 J1 = 1,NN
CC(I,J) = CC(I,J)+F(I,J1)*FINV(J1,J)
5      CONTINUE
4      CONTINUE
C
C IF IFUNC IS NOT = 0
C MULTIPLY BY PSI(X,Y)
C NOTE: LINE 11 CC(I,J) = -CC(I,J)*2./(X(I))
C SHOULD BE CHANGED TO ACCOMMODATE THE CHOSEN PSI(X,Y)
C IN PLACE OF 2./(X(I))
C
IF(IFUNC.NE.0) THEN
DO 11 J = 1,NN
DO 11 I = 1,NN
11      CC(I,J) = -CC(I,J)*2./(X(I))
END IF
C
C MULTIPLY RESULT BY (GQHAT-HUHAT)F-1
C
DO 6 I = 1,NN
DO 6 J = 1,NN
F(I,J) = 0.
DO 7 J1 = 1,NN
F(I,J) = F(I,J) + S(I,J1)*CC(J1,J)
7      CONTINUE
6      CONTINUE
C

```

```

C STORE IN S
C
      DO 8 I =1,NN
        DO 8 J = 1,NN
          S(I,J) = F(I,J)
8       CONTINUE
      NN = NH
      RETURN
      END

```

### 4.5.5 Results of Test Problems

Two test problems will now be considered: the convective problem described in section 4.2.1 and another in which a function  $\psi(x, y)$  is used to multiply the derivative of  $u$  on the right-hand side of the governing equation.

#### Convective Problem

The data file and computer output for the problem analysed in section 4.2.1 are printed below. The equation is  $\nabla^2 u = -\partial u / \partial x$ . Approximation  $f = 1 + r$  is employed. The results can be compared to column  $f = 1 + r$  of table 4.2. The data is as described for Program 2, with two extra lines at the beginning of the file as described in section 4.5.1.

Data input and computer output are listed in what follows.

#### Data Input

```

1 0 0
1. 1. 0. 0.
16 16 17
 2.0      0.0
 1.705706 -0.522150
 1.178800 -0.807841
 0.597614 -0.954310
 0.0      -1.0
-0.597614 -0.954310
-1.178800 -0.807841
-1.705706 -0.522150
-2.0      0.0
-1.705706 0.522150
-1.178800 0.807841
-0.597614 0.954310
 0.0      1.0
 0.597614 0.954310
 1.178800 0.807841
 1.705706 0.522150
 1.5      0.0
 1.2      -0.35
 0.6      -0.45
 0.0      -0.45

```

-0.6	-0.45
-1.2	-0.35
-1.5	0.0
-1.2	0.35
-0.6	0.45
0.0	0.45
0.6	0.45
1.2	0.35
0.9	0.0
0.3	0.0
0.0	0.0
-0.3	0.0
-0.9	0.0
1	0.135335
1	0.181644
1	0.307647
1	0.550122
1	1.000000
1	1.817776
1	3.250471
1	5.505270
1	7.389056
1	5.505270
1	3.250471
1	1.817776
1	1.000000
1	0.550122
1	0.307647
1	0.181644

## Computer Output

EQUATION MODELLED: NABLA2U=-DUDX

F EXPANSION:

F = 1. + 1.\*R + 0.\*R2 + 0.\*R3

NUMBER OF BOUNDARY NODES = 16

NUMBER OF BOUNDARY ELEMENTS = 16

NUMBER OF INTERNAL NODES = 17

DATA FOR BOUNDARY NODES

NODE	X	Y	TYPE	U	Q
1	2.000000	0.000000	1	0.135	0.000
2	1.705706	-0.522150	1	0.182	0.000
3	1.178800	-0.807841	1	0.308	0.000
4	0.597614	-0.954310	1	0.550	0.000

5	0.000000	-1.000000	1	1.000	0.000
6	-0.597614	-0.954310	1	1.818	0.000
7	-1.178800	-0.807841	1	3.250	0.000
8	-1.705706	-0.522150	1	5.505	0.000
9	-2.000000	0.000000	1	7.389	0.000
10	-1.705706	0.522150	1	5.505	0.000
11	-1.178800	0.807841	1	3.250	0.000
12	-0.597614	0.954310	1	1.818	0.000
13	0.000000	1.000000	1	1.000	0.000
14	0.597614	0.954310	1	0.550	0.000
15	1.178800	0.807841	1	0.308	0.000
16	1.705706	0.522150	1	0.182	0.000

## COORDINATES OF INTERNAL NODES

NODE	X	Y
------	---	---

17	1.500000	0.000000
18	1.200000	-0.350000
19	0.600000	-0.450000
20	0.000000	-0.450000
21	-0.600000	-0.450000
22	-1.200000	-0.350000
23	-1.500000	0.000000
24	-1.200000	0.350000
25	-0.600000	0.450000
26	0.000000	0.450000
27	0.600000	0.450000
28	1.200000	0.350000
29	0.900000	0.000000
30	0.300000	0.000000
31	0.000000	0.000000
32	-0.300000	0.000000
33	-0.900000	0.000000

## ELEMENT DATA

NO	NODE 1	NODE 2	LENGTH
1	1	2	0.599374
2	2	3	0.599374
3	3	4	0.599358
4	4	5	0.599358
5	5	6	0.599358
6	6	7	0.599358
7	7	8	0.599374
8	8	9	0.599374
9	9	10	0.599374
10	10	11	0.599374
11	11	12	0.599358
12	12	13	0.599358
13	13	14	0.599358
14	14	15	0.599358

15	15	16	0.599374
16	16	1	0.599374

**BOUNDARY RESULTS**

NODE	FUNCTION	DERIVATIVE
1	0.135335	-0.173976
2	0.181644	-0.160100
3	0.307647	-0.122819
4	0.550122	-0.090221
5	1.000000	0.011295
6	1.817776	0.319686
7	3.250471	1.045739
8	5.505270	3.554578
9	7.389056	6.407219
10	5.505270	3.554621
11	3.250471	1.045720
12	1.817776	0.319680
13	1.000000	0.011308
14	0.550122	-0.090228
15	0.307647	-0.122809
16	0.181644	-0.160112

**RESULTS AT INTERIOR NODES**

NODE	FUNCTION
17	0.229062
18	0.306990
19	0.554839
20	1.003448
21	1.819429
22	3.322700
23	4.488789
24	3.322700
25	1.819431
26	1.003448
27	0.554840
28	0.306991
29	0.411512
30	0.744661
31	1.001964
32	1.348246
33	2.448370

**Product of Terms**

The equation that will be considered in this case is

$$\nabla^2 u = -\psi(x, y) \frac{\partial u}{\partial x}$$

As an example  $\psi(x, y)$  is considered to be  $2/x$ , such that the same results of the analysis of the Burger equation of section 4.4.1 are obtained. The difference in this case is that, as the function is given explicitly, no iterations are needed. The discretization, boundary conditions and exact solution are as in section 4.4.1.

In order to use Program 3 for this case, IDDX is set to 1, IDDY to 0, and IFUNC to 1.

The data input and computer output for this case are now listed:

### Data Input

```

1 0 1
1. 1. 0. 0.
16 16 17
      5.0      0.0
4.705706 -0.522150
4.178800 -0.807841
3.597614 -0.954310
3.0      -1.0
2.402386 -0.954310
1.821200 -0.807841
1.294294 -0.522150
1.0      0.0
1.294294 0.522150
1.821200 0.807841
2.402386 0.954310
3.0      1.0
3.597614 0.954310
4.178800 0.807841
4.705706 0.522150
4.5      0.0
4.2      -0.35
3.6      -0.45
3.0      -0.45
2.4      -0.45
1.8      -0.35
1.5      0.0
1.8      0.35
2.4      0.45
3.0      0.45
3.6      0.45
4.2      0.35
3.9      0.0
3.3      0.0
3.0      0.0
2.7      0.0
2.1      0.0
1  0.400000
1  0.425016
1  0.478606
1  0.555924
1  0.666667
1  0.832506

```

```

1  1.098177
1  1.545244
1  2.000000
1  1.545244
1  1.098177
1  0.832506
1  0.666667
1  0.555924
1  0.478606
1  0.425016

```

## Computer Output

EQUATION MODELLED:  $\nabla^2 U = -F(X, Y) \frac{\partial U}{\partial X}$

F EXPANSION:

$F = 1. + 1.*R + 0.*R^2 + 0.*R^3$

NUMBER OF BOUNDARY NODES = 16

NUMBER OF BOUNDARY ELEMENTS = 16

NUMBER OF INTERNAL NODES = 17

DATA FOR BOUNDARY NODES

NODE	X	Y	TYPE	U	Q
1	5.000000	0.000000	1	0.400	0.000
2	4.705706	-0.522150	1	0.425	0.000
3	4.178800	-0.807841	1	0.479	0.000
4	3.597614	-0.954310	1	0.556	0.000
5	3.000000	-1.000000	1	0.667	0.000
6	2.402386	-0.954310	1	0.833	0.000
7	1.821200	-0.807841	1	1.098	0.000
8	1.294294	-0.522150	1	1.545	0.000
9	1.000000	0.000000	1	2.000	0.000
10	1.294294	0.522150	1	1.545	0.000
11	1.821200	0.807841	1	1.098	0.000
12	2.402386	0.954310	1	0.833	0.000
13	3.000000	1.000000	1	0.667	0.000
14	3.597614	0.954310	1	0.556	0.000
15	4.178800	0.807841	1	0.479	0.000
16	4.705706	0.522150	1	0.425	0.000

COORDINATES OF INTERNAL NODES

NODE	X	Y
17	4.500000	0.000000
18	4.200000	-0.350000

19	3.600000	-0.450000
20	3.000000	-0.450000
21	2.400000	-0.450000
22	1.800000	-0.350000
23	1.500000	0.000000
24	1.800000	0.350000
25	2.400000	0.450000
26	3.000000	0.450000
27	3.600000	0.450000
28	4.200000	0.350000
29	3.900000	0.000000
30	3.300000	0.000000
31	3.000000	0.000000
32	2.700000	0.000000
33	2.100000	0.000000

**ELEMENT DATA**

NO	NODE 1	NODE 2	LENGTH
1	1	2	0.599374
2	2	3	0.599374
3	3	4	0.599358
4	4	5	0.599358
5	5	6	0.599358
6	6	7	0.599358
7	7	8	0.599374
8	8	9	0.599374
9	9	10	0.599374
10	10	11	0.599374
11	11	12	0.599358
12	12	13	0.599358
13	13	14	0.599358
14	14	15	0.599358
15	15	16	0.599374
16	16	1	0.599374

**BOUNDARY RESULTS**

NODE	FUNCTION	DERIVATIVE
1	0.400000	-0.081453
2	0.425016	-0.066316
3	0.478606	-0.040690
4	0.555924	-0.024784
5	0.666667	0.000963
6	0.832506	0.063821
7	1.098177	0.205074
8	1.545244	0.757824
9	2.000000	1.619545
10	1.545244	0.757831
11	1.098177	0.205075
12	0.832506	0.063815

13	0.666667	0.000965
14	0.555924	-0.024785
15	0.478606	-0.040689
16	0.425016	-0.066313

## RESULTS AT INTERIOR NODES

## NODE FUNCTION

17	0.445695
18	0.477974
19	0.558293
20	0.669646
21	0.834598
22	1.110341
23	1.334071
24	1.110340
25	0.834598
26	0.669645
27	0.558292
28	0.477974
29	0.514928
30	0.608762
31	0.669301
32	0.742762
33	0.949939

## 4.6 Three-Dimensional Analysis

The Dual Reciprocity Method can be easily extended to three-dimensional applications. The fundamental solution of the three-dimensional Laplace equation is

$$u^* = \frac{1}{4\pi r} \quad (4.115)$$

where

$$r = (r_x^2 + r_y^2 + r_z^2)^{1/2} \quad (4.116)$$

the components  $r_x$ ,  $r_y$  and  $r_z$  being the projections of the vector  $\vec{r}$  on the  $x$ ,  $y$  and  $z$  axes, for example  $r_x = x_k - x_i$ . The comments made in section 3.3.2 about the signs of the components of  $r$  are also valid in this case.

Two-dimensional boundary elements are used to discretize the surfaces of the region. These elements and their interpolation functions are described in basic boundary element textbooks, such as [13] and [18]. Reference should also be made to Appendix 2 which presents numerical integration formulae that can be applied to both two- and three-dimensional problems.

The matrices  $\mathbf{H}$  and  $\mathbf{G}$  are calculated using these elements. The DRM relationship is as before, *i.e.*

$$\boldsymbol{\alpha} = \mathbf{F}^{-1}\mathbf{b} \quad (4.117)$$

where  $b$  is the source term of the equation under consideration. The same expansions may be used for  $f$ , i.e.

$$f = 1 + r + r^2 + \dots + r^m \quad (4.118)$$

However,  $\hat{u}$  and  $\hat{q}$  are modified to

$$\hat{u} = \frac{r^2}{6} + \frac{r^3}{12} + \dots + \frac{r^{m+2}}{(m+2)^2 + m} \quad (4.119)$$

and

$$\hat{q} = \left( r_x \frac{\partial x}{\partial n} + r_y \frac{\partial y}{\partial n} + r_z \frac{\partial z}{\partial n} \right) \left( \frac{1}{3} + \frac{r}{4} + \dots + \frac{r^m}{m+3} \right) \quad (4.120)$$

This is a consequence of the fact that now

$$f = \frac{\partial^2 \hat{u}}{\partial x^2} + \frac{\partial^2 \hat{u}}{\partial y^2} + \frac{\partial^2 \hat{u}}{\partial z^2} \quad (4.121)$$

and

$$\hat{q} = \frac{\partial \hat{u}}{\partial x} \frac{\partial x}{\partial n} + \frac{\partial \hat{u}}{\partial y} \frac{\partial y}{\partial n} + \frac{\partial \hat{u}}{\partial z} \frac{\partial z}{\partial n} \quad (4.122)$$

Equations (4.119) and (4.120) are now used to calculate matrices  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$ .

Taking the above into consideration, the matrix equations for the previously discussed cases may easily be found.

#### 4.6.1 Equations of the Type $\nabla^2 u = b(x, y, z)$

Equations (3.15) and (3.22) are also valid for three-dimensional problems. A computer program may be written to solve this type of equation modifying computer Program 2, section 3.4. The main changes are:

- **INPUT2**

A third coordinate,  $z$ , is needed for all nodes. A connectivity table, i.e. a list of nodes associated with each element, is needed for the boundary elements.

The numerical integration points and weight factors must be adequate for the two-dimensional element in use (see Appendices 1 and 2).

- **ASSEM2**

Replace with new routine for the two-dimensional boundary element.

- **RHSVEC**

Change calculation of  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  using equations (4.119) and (4.120).

- INTERM

To be changed in accordance with modifications performed in ASSEM2.

The program may be easily put together by the reader and will not be given here for space limitations.

## Application

As an example, the case

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = -2 \quad (4.123)$$

is considered. A particular solution to this equation is

$$u = -\frac{1}{3}(x^2 + y^2 + z^2) \quad (4.124)$$

which may easily be verified by the reader. When imposed as an essential boundary condition (4.124) is also the problem solution. Equation (4.123) was solved for the geometry shown in figure 4.15.

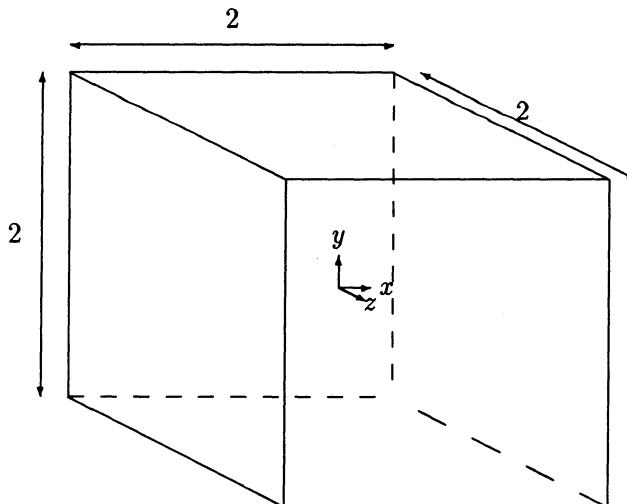


Figure 4.15: Geometry for Three-dimensional Problem

The problem consists of a cube of side 2 with the origin of the cartesian system of coordinates defined at the centroid. Each face was discretized with eight constant triangular elements, as shown in figure 4.16.

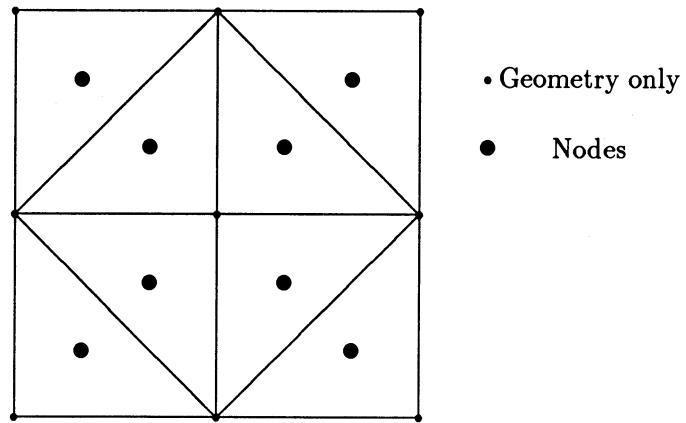


Figure 4.16: Discretization of One Face of Cube

Each face has 8 nodes making a total of 48 nodes. A total of 26 points was used to define the geometry of the elements. Each node is at the centroid of its element. The calculation of the coefficients of matrices  $\mathbf{H}$  and  $\mathbf{G}$  for this type of element is discussed in [18]. 27 internal nodes were defined at positions  $(\pm \frac{1}{2}, \pm \frac{1}{2}, \pm \frac{1}{2})$ ,  $(0, \pm \frac{1}{2}, \pm \frac{1}{2})$ ,  $(\pm \frac{1}{2}, 0, \pm \frac{1}{2})$ ,  $(\pm \frac{1}{2}, \pm \frac{1}{2}, 0)$ ,  $(0, 0, \pm \frac{1}{2})$ ,  $(\pm \frac{1}{2}, 0, 0)$ ,  $(0, \pm \frac{1}{2}, 0)$  and  $(0, 0, 0)$ .

In order to avoid confusion, reference will be made to a given node by its coordinates. Boundary conditions in  $u$  were defined using equation (4.124). Expansion  $f = 1 + r$  was employed. Results are presented in table 4.17.

$x$	$y$	$z$	$u$	Eqn. (4.97)
-0.5	-0.5	-0.5	0.2470	0.2500
0	-0.5	-0.5	0.1669	0.1667
0	0	-0.5	0.0834	0.0833
0	0	0	0.0000	0.0000

Table 4.17: Results for  $\nabla^2 u = -2$ 

The results given in the table describe the complete solution due to symmetry. The DRM results are seen to be excellent for this case, even using constant elements. It may be noted that this type of element has the advantages that: (i) the  $c_i$  terms are all equal to 0.5 at the boundary and need not be calculated, and (ii) there is no ambiguity in the definition of the normals, and thus no discontinuity in  $q$  or  $\hat{q}$ , which permits all matrices to be assembled.

## 4.6.2 Equations of the Type $\nabla^2 u = b(x, y, z, u)$

### The Convective Case in Three Dimensions

In this case the equation to be modelled is

$$\nabla^2 u = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \quad (4.125)$$

which can be handled using the procedures described in sections 4.2 and 4.2.2. As before

$$b = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \quad (4.126)$$

so

$$\alpha = \mathbf{F}^{-1} \left( \frac{\partial \mathbf{u}}{\partial x} + \frac{\partial \mathbf{u}}{\partial y} + \frac{\partial \mathbf{u}}{\partial z} \right) \quad (4.127)$$

Setting

$$\mathbf{u} = \mathbf{F}\beta \quad (4.128)$$

then

$$\beta = \mathbf{F}^{-1}\mathbf{u} \quad (4.129)$$

Equation (4.128) may be differentiated with respect to each of the coordinate directions to produce

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial x} &= \frac{\partial \mathbf{F}}{\partial x} \beta \\ \frac{\partial \mathbf{u}}{\partial y} &= \frac{\partial \mathbf{F}}{\partial y} \beta \\ \frac{\partial \mathbf{u}}{\partial z} &= \frac{\partial \mathbf{F}}{\partial z} \beta \end{aligned} \quad (4.130)$$

The vector  $\beta$  in equations (4.130) may be eliminated using (4.129). Then, substituting into (4.127), one obtains

$$\alpha = \mathbf{F}^{-1} \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{F}}{\partial z} \right) \mathbf{F}^{-1}\mathbf{u} \quad (4.131)$$

The matrix equation is as before, *i.e.*

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\alpha \quad (4.132)$$

or, substituting for  $\alpha$ ,

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1} \left( \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{F}}{\partial y} + \frac{\partial \mathbf{F}}{\partial z} \right) \mathbf{F}^{-1} \mathbf{u} \quad (4.133)$$

Vectors  $\partial\mathbf{F}/\partial x$  and  $\partial\mathbf{F}/\partial y$  are the same as used in section 4.2. A similar expression holds for  $\partial\mathbf{F}/\partial z$ .

## Application

The equation

$$\nabla^2 u = - \left( \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z} \right) \quad (4.134)$$

was solved for the geometry shown in figure 4.15, with the discretization shown in figure 4.16. The same internal nodes were used as in the previous example. The computer program can be obtained by performing similar modifications to Program 3 as described in section 4.5.1.

A particular solution for this case is

$$u = e^{-x} + e^{-y} + e^{-z} \quad (4.135)$$

which was imposed as the boundary condition and used as the problem solution. The expansion  $f = 1 + r$  was employed. Results are shown in table 4.18. The entire solution is described by the 10 values given in the table due to symmetry. Excellent agreement with the exact solution can be verified.

$x$	$y$	$z$	$u$	Eqn. (4.107)
-0.5	-0.5	-0.5	4.948	4.946
-0.5	-0.5	0.5	3.903	3.903
-0.5	0.5	0.5	2.866	2.861
0.5	0.5	0.5	1.833	1.819
0.0	-0.5	-0.5	4.294	4.292
0.0	-0.5	0.5	3.238	3.255
0.0	0.5	0.5	2.217	2.213
0.0	0.0	-0.5	3.629	3.648
0.0	0.0	0.5	2.599	2.606
0	0	0	2.982	3.000

Table 4.18: Results for  $\nabla^2 u = -(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} + \frac{\partial u}{\partial z})$

## The Case $\nabla^2 u = -u\partial u/\partial x$

Any of the applications considered in this chapter may be extended to three dimensions using the methods developed here, including product terms and non-linear terms.

The equation to be considered next is an example of them both. The DRM matrix equation for this case is exactly the same as for the two-dimensional case, i.e.

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = -(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\mathbf{U}\frac{\partial\mathbf{F}}{\partial x}\mathbf{F}^{-1}\mathbf{u} \quad (4.136)$$

except that the matrices  $\mathbf{H}$ ,  $\mathbf{G}$ ,  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  are now three-dimensional equivalents. The solution is  $u = 2/x$  as in the two-dimensional case. This problem was analysed for the geometry shown in figure 4.13 with the same nodes and elements as in the previous two examples, except that the origin was transferred to the point  $(-2, 0, 0)$  to avoid the singularity in the solution at  $x = 0$ . The solution process is exactly the same as that given in section 4.4.1 for the two-dimensional case. Expansion  $f = 1 + r$  was employed as before. Results are given in table 4.19. The three values presented describe the entire solution due to symmetry.

$x$	$y$	$z$	$u$	Exact
1.5	-0.5	-0.5	1.335	1.333
2.0	-0.5	-0.5	1.005	1.000
2.5	-0.5	-0.5	0.802	0.800

Table 4.19: Results for  $\nabla^2 u = -u \frac{\partial u}{\partial x}$

The solution converged in four iterations without using relaxation techniques, and can be seen to be very close to the exact values.

The use of higher-order  $f$  expansions for three-dimensional analysis is not recommended as tests showed them to produce inaccurate results.

## 4.7 References

1. Gresho, P. M. and Lee, R. L.  
Don't Suppress the Wiggles, They're Telling you Something! *Computers and Fluids*, Vol. 9, pp 223-252, 1981.
2. Payre, G., de Broissia, M. and Bazinet, J.  
An Upwind Finite Element Method via Numerical Integration, *International Journal for Numerical Methods in Engineering*, Vol. 18, pp 381-396, 1982.
3. Partridge, P. W.  
Analysis of Steady-State Circulation of the Guaiba River (Brazil), in *III Latin-American Computational Mechanics Congress*, Buenos Aires, 1982.
4. Craig, R. R.  
Structural Dynamics, Wiley, Chichester, 1981.
5. Connor, J. J. and Brebbia, C. A.  
Finite Element Techniques for Fluid Flow, Newnes-Butterworths, London, 1976.

6. Ciskowski, R. D. and Brebbia, C. A. (Ed.)  
Boundary Element Methods in Acoustics, Computational Mechanics Publications, Southampton and Elsevier, London, 1991.
7. Nardini, D. and Brebbia, C. A.  
A New Approach to Free Vibration Analysis using Boundary Elements, in *Boundary Element Methods in Engineering*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1982.
8. Brebbia, C. A. and Nardini, D.  
Dynamic Analysis in Solid Mechanics by an Alternative Boundary Element Procedure, *Soil Dynamics and Earthquake Engineering*, Vol. 2, pp 228-233, 1983.
9. Nardini, D. and Brebbia, C. A.  
Boundary Integral Formulation of Mass Matrices for Dynamic Analysis, in *Topics in Boundary Element Research*, Vol. 2, Springer-Verlag, Berlin and New York, 1985.
10. Wilkinson, J. H.,  
The Algebraic Eigenvalue Problem, Oxford University Press, Oxford, 1965.
11. Ali, A., Rajakumar, C. and Yunus, S. M.  
On the Formulation of the Acoustic Boundary Element Eigenvalue Problem, *International Journal for Numerical Methods in Engineering*, Vol. 31, pp 1271-1282, 1991.
12. Rajakumar, C. and Rogers, C. R.  
The Lanczos Algorithm Applied to Unsymmetric Generalized Eigenvalue Problems, *International Journal for Numerical Methods in Engineering*, in press.
13. Brebbia, C. A. and Dominguez, J.  
Boundary Elements: An Introductory Course, Computational Mechanics Publications, Southampton and McGraw-Hill, New York, 1989.
14. Kontoni, D. P. N., Partridge, P. W. and Brebbia, C. A.  
The Dual Reciprocity Boundary Element Method for the Eigenvalue Analysis of Helmholtz problems, *Advances in Engineering Software*, Vol. 13, pp 2-16, 1991.
15. Graff, K. F.  
Wave Motion in Elastic Solids, Ohio State University Press, 1975.
16. Shuku, T. and Ishihara, K.  
The Analysis of the Acoustic Field in Irregularly Shaped Rooms by the Finite Element Method, *Journal of Sound and Vibration*, Vol. 29, pp 67-76, 1973.
17. Ames, W. F.  
Nonlinear Partial Differential Equations in Engineering, Academic Press, New York, 1965.
18. Brebbia, C. A., Telles, J. C. F. and Wrobel, L. C.  
Boundary Element Techniques, Springer-Verlag, Berlin and New York, 1984.
19. Zinn, J. and Mader, C. L.  
Thermal Initiation of Explosives, *Journal of Applied Physics*, Vol. 31, pp 323-328, 1960.

20. Boddington, T., Gray, P. and Harvey, D. I.  
Thermal Theory of Spontaneous Ignition: Criticality in Bodies of Arbitrary Shape,  
*Phil. Trans. Royal Soc. London*, Vol. 270, pp 467-506, 1971.
21. Anderson, C. A. and Zienkiewicz, O. C.  
Spontaneous Ignition: Finite Element Solutions for Steady and Transient Conditions,  
*Trans. ASME Journal of Heat Transfer*, Vol. 96, pp 398-404, 1974.
22. Bialecki, R. and Nowak, A. J.  
Boundary Value Problems for Non-linear Material and Non-linear Boundary Conditions,  
*Applied Mathematical Modelling*, Vol. 5, pp 417-421, 1981.
23. Khader, M. S. and Hanna, M. C.  
An Iterative Boundary Integral Numerical Solution for General Steady Heat Conduction Problems,  
*Journal of Heat Transfer*, Vol. 103, pp 26-31, 1981.

# Chapter 5

## The Dual Reciprocity Method for Equations of the Type

$$\nabla^2 u = b(x, y, u, t)$$

### 5.1 Introduction

This chapter presents applications of the boundary element Dual Reciprocity Method to transient problems. Initially, the treatment of linear time-dependent equations is presented. This includes diffusion, wave propagation and convection-diffusion problems, for which the governing equations respectively are,

$$\nabla^2 u = \frac{1}{k} \frac{\partial u}{\partial t} \quad (5.1)$$

$$\nabla^2 u = \frac{1}{c^2} \frac{\partial^2 u}{\partial t^2} \quad (5.2)$$

$$\nabla^2 u = \frac{1}{D} \left( v_x \frac{\partial u}{\partial x} + v_y \frac{\partial u}{\partial y} - Ku + \frac{\partial u}{\partial t} \right) \quad (5.3)$$

For the DRM formulation, the fundamental solution of Laplace's equation is used and the above equations considered in the general form

$$\nabla^2 u = b(x, y, u, t) \quad (5.4)$$

The second part of the chapter deals with non-linear transient problems. Applications for this case will be restricted, for simplicity, to diffusion problems, although similar algorithms can be applied to other equations. Three types of non-linearities are considered: material ones, *i.e.* problems in which the physical parameter  $k$  depends on the function  $u$  itself; boundary conditions in which the flux is a non-linear function of  $u$ ; and non-linear internal sources, as in the case of spontaneous ignition.

## 5.2 The Diffusion Equation

We shall start by considering a diffusion problem governed by an equation of the form (5.1), written now with the notation

$$\nabla^2 u = \frac{1}{k} \dot{u} \quad (5.5)$$

in which the dot stands for time derivative and  $k$  is a material constant. The definition of the problem is completed with the specification of appropriate boundary conditions and initial conditions of the type  $u(x, y, t_0) = u_0(x, y)$ .

Comparing the above with equation (3.1), it is noted that function  $b$  in that equation is now defined as a constant term  $1/k$  multiplied by the time-derivative term  $\dot{u}$ . The application of the DRM follows the same pattern as in chapters 3 and 4, and produces a matrix equation of the form

$$\mathbf{H}u - \mathbf{G}q = \frac{1}{k}(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\boldsymbol{\alpha} \quad (5.6)$$

In the present case, approximation (3.3) implies a separation of variables in which  $f_j$  are known functions of geometry and  $\alpha_j$  unknown functions of time, *i.e.*

$$\dot{u}(x, y, t) \simeq \sum_{j=1}^{N+L} f_j(x, y) \alpha_j(t) \quad (5.7)$$

Thus, matrices  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{Q}}$  above are the same as in the previous chapters.

The next step in the formulation is similar to equation (4.6) of the previous chapter, *i.e.* the inversion of equation (3.17), written in this case as

$$\dot{\mathbf{u}} = \mathbf{F}\boldsymbol{\alpha} \quad (5.8)$$

to substitute for vector  $\boldsymbol{\alpha}$  in equation (5.6). This produces

$$\boldsymbol{\alpha} = \mathbf{F}^{-1}\dot{\mathbf{u}} \quad (5.9)$$

Substituting the above in (5.6), the following expression is obtained

$$\mathbf{H}u - \mathbf{G}q = \frac{1}{k}(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}\dot{\mathbf{u}} \quad (5.10)$$

The term multiplying  $\dot{\mathbf{u}}$  can be seen as a “heat capacity” matrix,

$$\mathbf{C} = -\frac{1}{k}\mathbf{S} = -\frac{1}{k}(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1} \quad (5.11)$$

and equation (5.10) rewritten in the form

$$\mathbf{C}\dot{\mathbf{u}} + \mathbf{H}u = \mathbf{G}q \quad (5.12)$$

System (5.12) is similar in form to the one obtained using the finite element method. Hence, any standard direct time-integration scheme can be used to find

a solution to the above system. It should be noted, however, that the vector  $\mathbf{q}$  of fluxes is present in (5.12), thus rendering it a system of equations of “mixed” type, as opposed to “displacement” finite element formulations.

For simplicity, a two-level time integration scheme will be employed here [1]. A linear approximation can be proposed for the variation of  $u$  and  $q$  within each time step, in the form

$$u = (1 - \theta_u)u^m + \theta_u u^{m+1} \quad (5.13)$$

$$q = (1 - \theta_q)q^m + \theta_q q^{m+1} \quad (5.14)$$

$$\dot{u} = \frac{1}{\Delta t}(u^{m+1} - u^m) \quad (5.15)$$

where  $\theta_u$  and  $\theta_q$  are parameters which position the values of  $u$  and  $q$ , respectively, between time levels  $m$  and  $m+1$ . Substituting these approximations into (5.12) yields:

$$\left( \frac{1}{\Delta t} \mathbf{C} + \theta_u \mathbf{H} \right) \mathbf{u}^{m+1} - \theta_q \mathbf{G} \mathbf{q}^{m+1} = \left[ \frac{1}{\Delta t} \mathbf{C} - (1 - \theta_u) \mathbf{H} \right] \mathbf{u}^m + (1 - \theta_q) \mathbf{G} \mathbf{q}^m \quad (5.16)$$

The right side of (5.16) is known at time  $(m + 1)\Delta t$ , since it involves values which have been specified as initial conditions or calculated previously. Upon introducing the boundary conditions at time  $(m + 1)\Delta t$ , one can rearrange the left side of (5.16) and solve the resulting system of equations for each time level.

Note that the elements of matrices  $\mathbf{H}$ ,  $\mathbf{G}$  and  $\mathbf{S}$  depend only on geometrical data. Thus, they can all be computed once and stored. If the value of  $\Delta t$  is kept constant, the system matrix can be reduced only once as well, and the time advance procedure will consist of a simple recursive scheme with only algebraic operations involved.

### 5.3 Computer Program 4

This program solves problems of the type  $\nabla^2 u = b(x, y, u, t)$  for a function  $b$  equal to  $1/k \partial u / \partial t$  (equation 5.5). A series of tests carried out by the authors indicated that, in general, good accuracy can be obtained for values of  $\theta_u$  and  $\theta_q$  equal to 0.5 and 1.0, respectively [2]. Thus, the program considers these particular values, in which case equation (5.16) becomes

$$\left( \frac{2}{\Delta t} \mathbf{C} + \mathbf{H} \right) \mathbf{u}^{m+1} - 2 \mathbf{G} \mathbf{q}^{m+1} = \left( \frac{2}{\Delta t} \mathbf{C} - \mathbf{H} \right) \mathbf{u}^m \quad (5.17)$$

The same options of  $f$  expansions as in Program 3 are available, as long as the constant and one other term are included. The  $f$  expansion and some specific data for transient analysis are set using data input as will be seen in section 5.3.1. Linear elements are again used for simplicity. Input and output for a test problem are also printed. The program is in modular form, using some of the routines described in previous chapters.

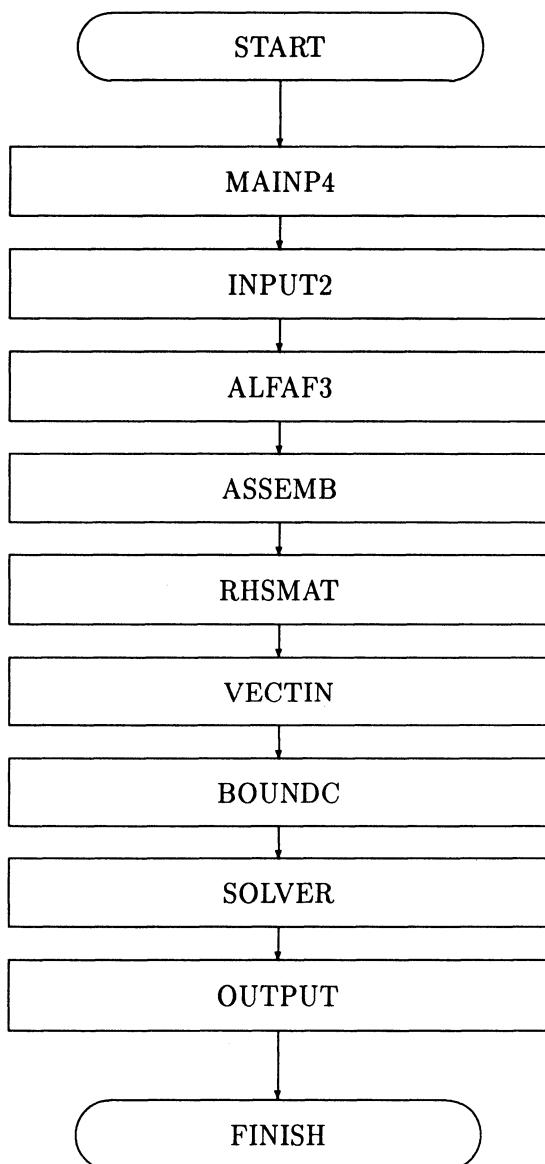


Figure 5.1: Modular Structure for Program 4

The program modules:

- INPUT2
- ALFAF3
- RHSMAT
- SOLVER
- OUTPUT

were described and listed previously and will be used in the form already given. In this section the following new routines will be described and listed:

- MAINP4
- ASSEMB
- VECTIN
- BOUND<sub>C</sub>

To construct Program 4, one can start from Program 3 and carry out the following operations:

- i. Substitute **MAINP4** for **MAINP3**
- ii. Substitute **ASSEMB** for **ASSEM2**
- iii. Remove **DERIVXY**
- iv. Include **VECTIN** and **BOUND<sub>C</sub>**

The variable names used in the program are the same as those used in Programs 1, 2 and 3. Where new variables are introduced, these will have symbols as close as possible to those used in the text. At the end of the section, the data set and results for a test problem are given.

### 5.3.1 MAINP4

The first step is to define the  $f$  expansion to be employed which in this case is the same as described in chapter 4. Next, some specific data required for the diffusion analysis are read. These include:

- i. NUMTS - number of time steps
- ii. DT - time step value
- iii. TI - potential value at initial time (assumed constant for simplicity)
- iv. CK - value of the material parameter  $k$

The remaining data are as for Program 2, and as INPUT2 will be used to read the data, this point will not be further commented upon. The  $f$  expansion and data for diffusion analysis are printed as the first output of the program.

The next step is to call routine ALFAF3 to calculate  $\mathbf{F}^{-1}$ . Then, ASSEMB is called to evaluate and store  $\mathbf{H}$  and  $\mathbf{G}$ , and routine RHSMAT to calculate  $\hat{\mathbf{U}}$ ,  $\hat{\mathbf{Q}}$  and  $\mathbf{S} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}$ .

The program then enters a DO loop over the time steps, in which the vector of initial conditions (right-hand side of equation 5.17) is first calculated in routine VECTIN and boundary conditions subsequently applied to the left-hand side of (5.17) in routine BOUNDC. Vector XY starts as  $\mathbf{y}$  and after SOLVER is called it returns as  $\mathbf{x}$ . Results are distributed between  $\mathbf{u}$  and  $\mathbf{q}$  according to the boundary condition type. These results are printed in OUTPUT; the program then proceeds to a new time step until NUMTS is achieved.

```

COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(100),
1 CON(100,2),KODE(200),Q(100),POEI(4),FDEP(4),ALPHA(100),NI
COMMON/FIVE/XY(100),A(100,100)
COMMON/FACTOR/C1,C2,C3,C4,IDDX,IDDY,IFUNC
COMMON/TWO/S(200,200),FINV(200,200)
COMMON/SIX/NUMTS,DT,TI,CK
COMMON/SEVEN/UP(100),QP(100)
COMMON/DRM/HH(100,100),GG(100,200)
REAL LE,LJ
INTEGER CON
C
C MAINP4
C
C DEFINE F EXPANSION TO BE USED:
C
C F EXPANSION IS OF THE FORM F= C1+C2*R+C3*R2+C4*R3
C
      READ(5,2005)C1,C2,C3,C4
2005  FORMAT(4F3.1)
C
C WRITE F EXPANSION IN USE
C
      WRITE(6,2006)
2006  FORMAT(//' F EXPANSION: '//)
      WRITE(6,2007)C1,C2,C3,C4
2007  FORMAT(' F = ',F2.0,' + ',F2.0,'*R + ',F2.0,
     1 '*R2 + ',F2.0,'*R3 '//)
C
C READ DATA FOR DIFFUSION ANALYSIS
C
      READ(5,3005)NUMTS,DT,TI,CK
3005  FORMAT(I5,3F10.5)
C
C WRITE DATA FOR DIFFUSION ANALYSIS
C
      WRITE(6,3006)
3006  FORMAT(' DATA FOR DIFFUSION ANALYSIS: //')

```

```

      WRITE(6,3007)NUMTS,DT,TI,CK
3007  FORMAT(' NUMBER OF TIME STEPS = ',I3,/
     1       ' TIME STEP VALUE = ',F10.6,/,
     1       ' INITIAL VALUE OF U = ',F10.6,/,
     1       ' PARAMETER K = ',F10.6,//)

C
C DATA INPUT
C
      CALL INPUT2
C
C SET UP THE VECTOR OF INITIAL VALUES
C INITIAL U IS CONSTANT, INITIAL Q IS ZERO
C
      DO 3008 J1=1,NN+L
      UP(J1)=TI
      QP(J1)=0.
3008  CONTINUE
C
C CALCULATION OF F-1
C
      CALL ALFAF3
C
C CALCULATION OF MATRICES G AND H
C
      CALL ASSEMB
C
C CALCULATION OF MATRIX S
C
      CALL RHSMAT
C
C START OF TIME STEP PROCEDURE
C
      DO 10 ITS=1,NUMTS
C
C CALCULATION OF VECTOR OF INITIAL CONDITIONS IN XY
C
      CALL VECTIN
C
C APPLY BOUNDARY CONDITIONS
C RESULTING MATRIX IN A; RESULTING VECTOR IN XY
C
      CALL BOUNDC
C
C SOLVE FOR BOUNDARY AND INTERNAL VALUES
C
      NH=NN
      NN=NN+L
      CALL SOLVER
      NN=NH
C
C PUT VALUES IN APPROPRIATE ARRAY; U OR Q
C
      DO 76 J1=1,NN+L

```

```

      KK=KODE(J1)
      IF(KK.EQ.0.OR.KK.EQ.2)THEN
        U(J1) = XY(J1)
      ELSE
        Q(J1) = XY(J1)
      END IF
76      CONTINUE
C
C SET UP INITIAL CONDITIONS FOR NEXT TIME STEP
C
      DO 77 J1=1,NN+L
      UP(J1) = U(J1)
      QP(J1) = Q(J1)
77      CONTINUE
C
C WRITE RESULTS
C
      TS=ITS*DT
      WRITE(6,1005)TS
1005  FORMAT(//' RESULTS AT TIME ',F10.5)
      CALL OUTPUT
C
C END OF TIME STEP LOOP
C
10      CONTINUE
      STOP
      END

```

### 5.3.2 Subroutine ASSEMB

This is a modified version of subroutine ASSEM2 used in the previous programs which only calculates and stores matrices  $\mathbf{G}$  and  $\mathbf{H}$ . Boundary conditions are now applied in subroutine BOUND<sub>C</sub> to be described in subsection 5.3.4.

```

SUBROUTINE ASSEMB
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(100),
1 CON(100,2),KODE(200),Q(100),POEI(4),FDEP(4),ALPHA(100),NI
COMMON/DRM/HH(100,100),GG(100,200)
REAL LE,LJ
INTEGER CON
C
C EVALUATE G AND H MATRICES FOR LINEAR BOUNDARY ELEMENTS
C
      PI = 3.141592654
C
C COEFFICIENTS OF G AND H CALCULATED USING FORMULAS (2.46-47, 2.49-50)
C
      DO 1 J1 = 1,NN+L
      XI = X(J1)
      YI = Y(J1)
      DO 21 J = 1,NN
      HH(J1,J)=0.

```

```

21      CONTINUE
CC = 0.
DO 2 J2 = 1,NE
LJ = LE(J2)
N1 = CON(J2,1)
N2 = CON(J2,2)
X1 = X(N1)
X2 = X(N2)
Y1 = Y(N1)
Y2 = Y(N2)
H1 = 0.
H2 = 0.
G1 = 0.
G2 = 0.
DO 3 J3 = 1,NI
E = POEI(J3)
W = FDEP(J3)
XX = X1 + (1.+E)*(X2-X1)/2.
YY = Y1 + (1.+E)*(Y2-Y1)/2.
R = SQRT((XI-XX)**2 + (YI-YY)**2)
PP = ((XI-XX)*(Y1-Y2)+(YI-YY)*(X2-X1))/(R*R*4.*PI)*W
H1 = H1 + (1.-E)*PP/2.0
H2 = H2 + (1.+E)*PP/2.0
PP = ALOG(1./R)/(4.*PI)*LJ*W
G1 = G1 + (1.-E)*PP/2.
G2 = G2 + (1.+E)*PP/2.
3      CONTINUE
C
C DIAGONAL TERMS OF H CALCULATED BY ADDING OFF-DIAGONAL COEFFICIENTS
C
CC = CC - H1 - H2
C
C ANALYTICAL INTEGRATION FOR G WHEN I AND J ARE ON THE SAME ELEMENT
C
GE = LJ*(3./2.-ALOG(LJ))/(4.*PI)
C
C STORE FULL G AND H MATRICES FOR USE WITH DRM
C H ASSEMBLED, G UNASSEMBLED
C
IF(N1.EQ.J1) G1=GE
IF(N2.EQ.J1) G2=GE
HH(J1,N1)=HH(J1,N1)+H1
HH(J1,N2)=HH(J1,N2)+H2
GG(J1,2*J2-1) = G1
GG(J1,2*J2)    = G2
2      CONTINUE
C
C DIAGONAL COEFFICIENTS OF MATRIX H
C
HH(J1,J1) = CC
1      CONTINUE
RETURN
END

```

### 5.3.3 Subroutine VECTIN

This subroutine calculates the vector of initial conditions, given by the right-hand side of equation (5.17):

$$\left( \frac{2}{\Delta t} \mathbf{C} - \mathbf{H} \right) \mathbf{u}^m \quad (5.18)$$

The vector resulting from the above expression is stored in XY.

```

SUBROUTINE VECTIN
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(100),
1 CON(100,2),KODE(200),Q(100),POEI(4),FDEP(4),ALPHA(100),NI
COMMON/TWO/S(200,200),FINV(200,200)
COMMON/FIVE/XY(100),A(100,100)
COMMON/SIX/NUMTS,DT,TI,CK
COMMON/SEVEN/UP(100),QP(100)
COMMON/DRM/HH(100,100),GG(100,200)
REAL LE
INTEGER CON
C
C CALCULATE VECTOR OF INITIAL CONDITIONS
C
CT = -2./(CK*DT)
DO 1 I = 1,NN+L
XY(I) = 0.
      DO 2 J = 1,NN+L
      XY(I) = XY(I) + (CT*S(I,J)-HH(I,J))*UP(J)
2      CONTINUE
1      CONTINUE
      RETURN
END

```

### 5.3.4 Subroutine BOUNDC

This subroutine applies the boundary conditions to the left-hand side of equation (5.17), *i.e.*

$$\left( \frac{2}{\Delta t} \mathbf{C} + \mathbf{H} \right) \mathbf{u}^{m+1} - 2G\mathbf{q}^{m+1} \quad (5.19)$$

The system matrix is stored in A, and the independent term added to XY.

```

SUBROUTINE BOUNDC
COMMON/ONE/NN,NE,L,X(200),Y(200),U(200),D(200),LE(100),
1 CON(100,2),KODE(200),Q(100),POEI(4),FDEP(4),ALPHA(100),NI
COMMON/TWO/S(200,200),FINV(200,200)
COMMON/FIVE/XY(100),A(100,100)
COMMON/SIX/NUMTS,DT,TI,CK
COMMON/SEVEN/UP(100),QP(100)
COMMON/DRM/HH(100,100),GG(100,100)
REAL LE
INTEGER CON

```

```

C
C APPLY BOUNDARY CONDITIONS AND CALCULATE VECTOR OF
C INDEPENDENT COEFFICIENTS
C
      CT = -2.*/(CK*DT)
      DO 1 I = 1,NN+L
C
C CONTRIBUTION OF MATRICES H AND S (ASSEMBLED)
C
      DO 2 J = 1,NN+L
      A(I,J) = 0.
      KK=KODE(J)
      IF(KK.EQ.0.OR.KK.EQ.2)THEN
      A(I,J) = CT*S(I,J)+HH(I,J)
      ELSE
      XY(I) = XY(I) - (CT*S(I,J)+HH(I,J))*U(J)
      END IF
2     CONTINUE
C
C CONTRIBUTION OF MATRIX G (UNASSEMBLED)
C
      DO 3 J = 1,NE
      N1 = CON(J,1)
      N2 = CON(J,2)
      KK = KODE(N1)
      IF(KK.EQ.0.OR.KK.EQ.2) THEN
      XY(I) = XY(I) + 2.*GG(I,2*J-1)*Q(N1)
      ELSE
      A(I,N1) = A(I,N1) - 2.*GG(I,2*J-1)
      END IF
      KK = KODE(N2)
      IF(KK.EQ.0.OR.KK.EQ.2) THEN
      XY(I) = XY(I) + 2.*GG(I,2*J)*Q(N2)
      ELSE
      A(I,N2) = A(I,N2) - 2.*GG(I,2*J)
      END IF
3     CONTINUE
1     CONTINUE
      RETURN
      END

```

### 5.3.5 Results of a Test Problem

The problem of heat diffusion in a square plate initially at  $30^\circ$  and cooled by the application of a thermal shock ( $u = 0^\circ$  all over the boundary) has been studied using the finite element method by Bruch and Zyzvoloski [3], and using the BEM with time-dependent fundamental solutions by Brebbia, Telles and Wrobel [4]. The FEM mesh consisted of 25 quadratic elements and 96 nodes as shown in figure 5.2. The BEM analyses of [4] used 8 linear boundary elements and 9 nodes to discretize one-quarter of the region, taking advantage of symmetry. Two different approaches were investigated in [4]: in the first (named BEM1), internal cells were employed in order to account

for the initial conditions at the beginning of each time step; in the second (BEM2), the solution process always restarted at the initial time and domain discretization was avoided.

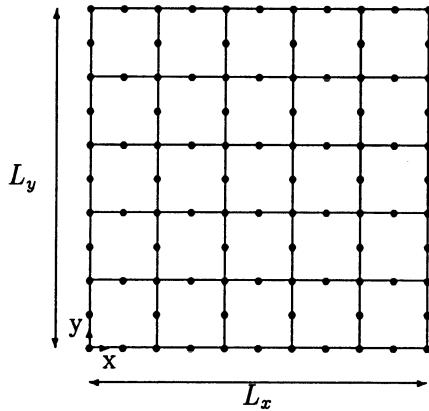


Figure 5.2: FEM Mesh of Bruch and Zyzoloski [3]

The exact solution for this problem is given in reference [3] as

$$u = \sum_{n=1}^{\infty} \sum_{j=1}^{\infty} A_{nj} \sin \frac{n\pi x}{L_x} \sin \frac{n\pi y}{L_y} \exp \left[ - \left( \frac{K_x n^2 \pi^2}{L_x^2} + \frac{K_y j^2 \pi^2}{L_y^2} \right) t \right]$$

where

$$A_{nj} = \frac{4u_0}{nj\pi^2} [(-1)^n - 1][(-1)^j - 1]$$

In all analyses, the numerical values adopted were  $L_x = L_y = 3$ ,  $K_x = K_y = 1.25$  and  $u_0 = 30$ .

This problem presents a peculiarity for application of the DRM. Since all the solution information is progressed in time through the term in  $u$  (see equation 5.17), and  $u$  is herein prescribed everywhere on the boundary, internal degrees of freedom *must* be introduced in this particular case.

Table 5.1 shows a comparison of results for a boundary discretization with 40 elements and different numbers of internal nodes, obtained with a time step  $\Delta t = 0.05$ . It can be seen that the results seem to converge to values which are slightly higher than the exact ones.

9 int. nodes	25 int. nodes	33 int. nodes	49 int. nodes	Exact[3]
1.321	1.787	1.877	1.891	1.812

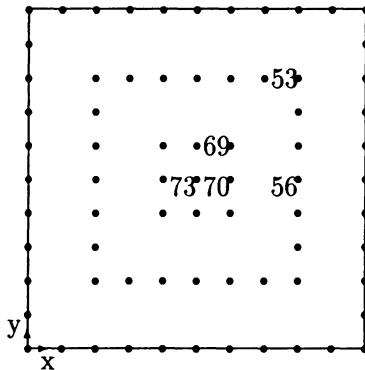
Table 5.1: DRM Results for  $u$  at Centre Point, for  $t = 1.2$ 

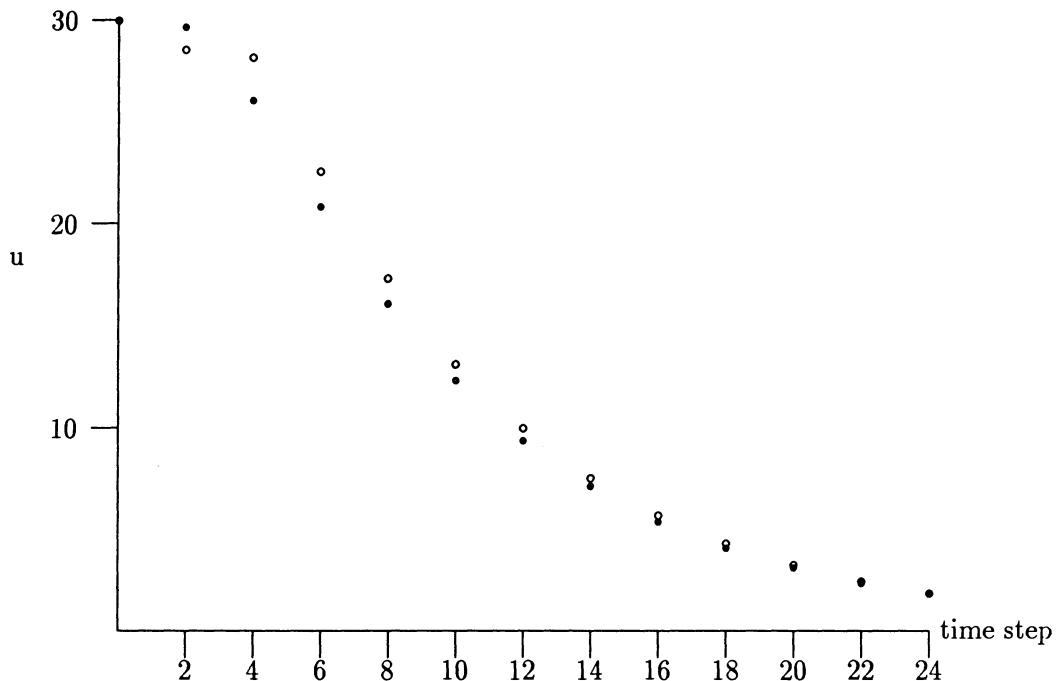
Figure 5.3: DRM Discretization

Table 5.2 presents a comparison of the DRM results with 33 internal nodes (figure 5.3) and the other BEM and FEM solutions, obtained with the same time step value ( $\Delta t = 0.05$ ).

	$x$	$y$	BEM1[4]	BEM2[4]	FEM[3]	DRM	Exact[3]
$u_{56}$	2.4	1.5	1.114	1.122	1.139	1.099	1.065
$u_{53}$	2.4	2.4	0.657	0.663	0.670	0.645	0.626
$u_{70}$	1.8	1.5	1.798	1.809	1.843	1.784	1.723
$u_{69}$	1.8	1.8	1.713	1.721	1.753	1.695	1.639
$u_{73}$	1.5	1.5	1.887	1.902	1.938	1.877	1.812

Table 5.2: Comparison of Results at  $t = 1.2$ 

Figure 5.4 shows a comparison of the time variation of  $u$  at the centre point. As expected in a thermal shock problem, the largest errors appear at the initial times since the shock is applied linearly over the first time step in the computational model and not suddenly as in the mathematical problem. It can be observed that the initial oscillations are quickly damped.

Figure 5.4: Time Variation of  $u$  at Centre Point: • Exact, ○ DRM

### 5.3.6 Data Input

```

1. 1. 0. 0.
 24 0.05      30.        1.25
 40 40 33
0.          0.
0.3         0.
0.6         0.
0.9         0.
1.2         0.
1.5         0.
1.8         0.
2.1         0.
2.4         0.
2.7         0.
3.          0.
3.          0.3
3.          0.6
3.          0.9
3.          1.2
3.          1.5
3.          1.8
3.          2.1
3.          2.4
3.          2.7
3.          3.

```

2.7	3.
2.4	3.
2.1	3.
1.8	3.
1.5	3.
1.2	3.
0.9	3.
0.6	3.
0.3	3.
0.	3.
0.	2.7
0.	2.4
0.	2.1
0.	1.8
0.	1.5
0.	1.2
0.	0.9
0.	0.6
0.	0.3
0.6	0.6
0.9	0.6
1.2	0.6
1.5	0.6
1.8	0.6
2.1	0.6
2.4	0.6
2.4	0.9
2.4	1.2
2.4	1.5
2.4	1.8
2.4	2.1
2.4	2.4
2.1	2.4
1.8	2.4
1.5	2.4
1.2	2.4
0.9	2.4
0.6	2.4
0.6	2.1
0.6	1.8
0.6	1.5
0.6	1.2
0.6	0.9
1.2	1.2
1.5	1.2
1.8	1.2
1.8	1.5
1.8	1.8
1.5	1.8
1.2	1.8
1.2	1.5
1.5	1.5

### 5.3.7 Computer Output

Due to its length, the computer output for this problem was truncated to display only the results at the final time step, *i.e.*  $t = 1.2$ .

## F EXPANSION:

$$F = 1. + 1.*R + 0.*R2 + 0.*R3$$

NUMBER OF BOUNDARY NODES = 40

NUMBER OF BOUNDARY ELEMENTS = 40

NUMBER OF INTERNAL NODES = 33

DATA FOR BOUNDARY NODES

NODE	X	Y	TYPE	U	Q
1	0.000000	0.000000	1	0.000	0.000
2	0.300000	0.000000	1	0.000	0.000
3	0.600000	0.000000	1	0.000	0.000
4	0.900000	0.000000	1	0.000	0.000
5	1.200000	0.000000	1	0.000	0.000
6	1.500000	0.000000	1	0.000	0.000
7	1.800000	0.000000	1	0.000	0.000
8	2.100000	0.000000	1	0.000	0.000
9	2.400000	0.000000	1	0.000	0.000
10	2.700000	0.000000	1	0.000	0.000
11	3.000000	0.000000	1	0.000	0.000
12	3.000000	0.300000	1	0.000	0.000
13	3.000000	0.600000	1	0.000	0.000
14	3.000000	0.900000	1	0.000	0.000
15	3.000000	1.200000	1	0.000	0.000
16	3.000000	1.500000	1	0.000	0.000
17	3.000000	1.800000	1	0.000	0.000
18	3.000000	2.100000	1	0.000	0.000
19	3.000000	2.400000	1	0.000	0.000
20	3.000000	2.700000	1	0.000	0.000
21	3.000000	3.000000	1	0.000	0.000
22	2.700000	3.000000	1	0.000	0.000
23	2.400000	3.000000	1	0.000	0.000
24	2.100000	3.000000	1	0.000	0.000
25	1.800000	3.000000	1	0.000	0.000
26	1.500000	3.000000	1	0.000	0.000
27	1.200000	3.000000	1	0.000	0.000
28	0.900000	3.000000	1	0.000	0.000
29	0.600000	3.000000	1	0.000	0.000
30	0.300000	3.000000	1	0.000	0.000
31	0.000000	3.000000	1	0.000	0.000
32	0.000000	2.700000	1	0.000	0.000
33	0.000000	2.400000	1	0.000	0.000
34	0.000000	2.100000	1	0.000	0.000
35	0.000000	1.800000	1	0.000	0.000
36	0.000000	1.500000	1	0.000	0.000
37	0.000000	1.200000	1	0.000	0.000
38	0.000000	0.900000	1	0.000	0.000
39	0.000000	0.600000	1	0.000	0.000
40	0.000000	0.300000	1	0.000	0.000

COORDINATES OF INTERNAL NODES

NODE	X	Y
------	---	---

```

41 0.600000 0.600000
42 0.900000 0.600000
43 1.200000 0.600000
44 1.500000 0.600000
45 1.800000 0.600000
46 2.100000 0.600000
47 2.400000 0.600000
48 2.400000 0.900000
49 2.400000 1.200000
50 2.400000 1.500000
51 2.400000 1.800000
52 2.400000 2.100000
53 2.400000 2.400000
54 2.100000 2.400000
55 1.800000 2.400000
56 1.500000 2.400000
57 1.200000 2.400000
58 0.900000 2.400000
59 0.600000 2.400000
60 0.600000 2.100000
61 0.600000 1.800000
62 0.600000 1.500000
63 0.600000 1.200000
64 0.600000 0.900000
65 1.200000 1.200000
66 1.500000 1.200000
67 1.800000 1.200000
68 1.800000 1.500000
69 1.800000 1.800000
70 1.500000 1.800000
71 1.200000 1.800000
72 1.200000 1.500000
73 1.500000 1.500000

```

## ELEMENT DATA

NO	NODE 1	NODE 2	LENGTH
1	1	2	0.300000
2	2	3	0.300000
3	3	4	0.300000
4	4	5	0.300000
5	5	6	0.300000
6	6	7	0.300000
7	7	8	0.300000
8	8	9	0.300000
9	9	10	0.300000
10	10	11	0.300000
11	11	12	0.300000
12	12	13	0.300000
13	13	14	0.300000

14	14	15	0.300000
15	15	16	0.300000
16	16	17	0.300000
17	17	18	0.300000
18	18	19	0.300000
19	19	20	0.300000
20	20	21	0.300000
21	21	22	0.300000
22	22	23	0.300000
23	23	24	0.300000
24	24	25	0.300000
25	25	26	0.300000
26	26	27	0.300000
27	27	28	0.300000
28	28	29	0.300000
29	29	30	0.300000
30	30	31	0.300000
31	31	32	0.300000
32	32	33	0.300000
33	33	34	0.300000
34	34	35	0.300000
35	35	36	0.300000
36	36	37	0.300000
37	37	38	0.300000
38	38	39	0.300000
39	39	40	0.300000
40	40	1	0.300000

RESULTS AT TIME 1.20000

#### BOUNDARY RESULTS

NODE	FUNCTION	DERIVATIVE
1	0.000000	-0.006096
2	0.000000	0.649397
3	0.000000	1.229460
4	0.000000	1.691521
5	0.000000	1.988043
6	0.000000	2.090044
7	0.000000	1.987848
8	0.000000	1.691523
9	0.000000	1.229341
10	0.000000	0.649450
11	0.000000	-0.006025
12	0.000000	0.649338
13	0.000000	1.229320
14	0.000000	1.691538
15	0.000000	1.988380
16	0.000000	2.089870
17	0.000000	1.987062

18	0.000000	1.691141
19	0.000000	1.230011
20	0.000000	0.650189
21	0.000000	-0.007413
22	0.000000	0.650750
23	0.000000	1.228153
24	0.000000	1.689879
25	0.000000	1.987874
26	0.000000	2.092428
27	0.000000	1.987509
28	0.000000	1.688387
29	0.000000	1.232204
30	0.000000	0.650456
31	0.000000	-0.006304
32	0.000000	0.649043
33	0.000000	1.226271
34	0.000000	1.696875
35	0.000000	1.984183
36	0.000000	2.089899
37	0.000000	1.988999
38	0.000000	1.691176
39	0.000000	1.229095
40	0.000000	0.649465

## RESULTS AT INTERIOR NODES

NODE	FUNCTION
41	0.645188
42	0.888956
43	1.045160
44	1.099494
45	1.045225
46	0.888973
47	0.645160
48	0.889004
49	1.045182
50	1.099500
51	1.045086
52	0.889171
53	0.645150
54	0.889216
55	1.045004
56	1.099621
57	1.045242
58	0.888977
59	0.644996
60	0.889045
61	1.045266
62	1.099607
63	1.044934
64	0.889092
65	1.696054

66	1.784125
67	1.696080
68	1.784304
69	1.695971
70	1.784227
71	1.695940
72	1.784351
73	1.877048

### 5.3.8 Further Applications

The one-dimensional problem of an infinite slab subjected to a thermal shock was studied next. The problem has been modelled as two-dimensional with mixed boundary conditions, *i.e.*  $u = 1$  prescribed along the faces  $x = \pm L$  and  $q = 0$  along the faces  $y = \pm l$  of a rectangular region with zero initial temperature. The numerical values adopted for the geometrical dimensions and material parameters were  $L = 5$ ,  $l = 4$ ,  $k = 1$  and  $\Delta t = 1$ .

Table 5.3 presents results for  $u$  at the centre point ( $x = y = 0$ ) for several time levels, compared to the analytical solution of [5]. The discretization employed herein consisted of 56 boundary elements and 1 internal point located at  $x = y = 0$ . Also included in the table are the results of [6], also obtained with the DRM and a similar discretization (14 linear boundary elements over one-quarter of the region, taking symmetry into consideration), one central internal point and  $\theta_u = \theta_q = 1$ .

Time	Present Analysis	Wrobel and Brebbia[6]	Analytical[5]
2	-0.051	0.016	0.025
4	0.114	0.166	0.154
6	0.278	0.302	0.298
8	0.416	0.418	0.422
10	0.527	0.515	0.526
15	0.710	0.694	0.710
20	0.838	0.807	0.823

Table 5.3: Comparison of Results for Problem 2

Some important points should be noted in the present analysis. The first regards the number of internal points, which can be small in this case due to the sufficient number of degrees of freedom along the boundary. In fact, increasing the number of internal points does not necessarily improve the solution in diffusion problems, as pointed out in [6]. Different  $f$  expansions were used here and in [6], namely  $f = 1 + r$  at all nodes versus  $f = 1$  at one node and  $f = r$  at the remaining ones, as explained in chapter 3 (section 3.2). The results are not much affected by this, and do not seem to be sensitive either to the different values of  $\theta_q$  employed in the two analyses.

A second point to be noted is the special way in which points of geometric discontinuity (corners) are treated in the analyses. In [6], discontinuous elements were used

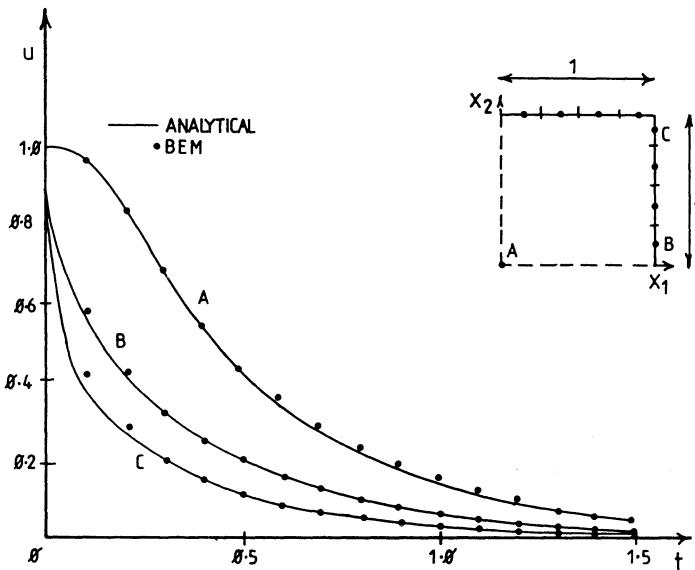


Figure 5.5: Temperature Variation at Some Points

near corners while the present analysis introduced small cutouts that only influence the solution locally. A better treatment of corners can be implemented by allowing the fluxes to be discontinuous (*i.e.*  $q_{\text{before}} \neq q_{\text{after}}$  for each node), as done in [7] and described in chapter 2.

Another application of the DRM to a diffusion problem, reproduced from [1], is the case of a square region with unit initial temperature convecting into a surrounding medium at zero temperature. The heat transfer coefficient is constant all over the boundary and equal to 2, while the thermal diffusivity is assumed to be unity.

The boundary element discretization employed only 8 constant elements over one-quarter of the region, due to the double symmetry of the problem. Results are shown in figure 5.5 for the temperature variation at some boundary and internal points, for  $\theta_u = \theta_q = 1$  and  $\Delta t = 0.05$ , together with the analytical solution given in [5].

### 5.3.9 Other Time-Stepping Schemes

While the basic DRM formulation for the diffusion equation presented in section 5.2 employed a simple two-level time integration scheme, more refined procedures have been utilized by Singh and Kalra [8] and Lahrmann and Haberland [9].

Singh and Kalra [8] adopted a least squares formulation which starts with the construction of a functional  $\Pi$  given by the integral of the square of the error over a time step, in the form

$$\Pi = \int_0^1 [\mathbf{C}\dot{\mathbf{u}} + \mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q}]^T [\mathbf{C}\dot{\mathbf{u}} + \mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q}] d\zeta \quad (5.20)$$

where  $\zeta = (t - t^m)/\Delta t$ ,  $t^m \leq t \leq t^{m+1}$ ,  $t^m = m\Delta t$  and  $t^{m+1} = (m + 1)\Delta t$ .

Introducing the boundary conditions and minimizing the functional  $\Pi$  with respect to the vector of nodal unknowns  $\mathbf{X}$ , the following set of equations is obtained:

- For  $X_i = u_i^{m+1}$ , i.e. a Neumann boundary condition prescribed at node  $i$ , the  $i$ th equation is of the form:

$$\sum_j \left\{ \left[ \frac{1}{\Delta t^2} \mathbf{C}^T \mathbf{C} + \frac{1}{2\Delta t} (\mathbf{C}^T \mathbf{H} + \mathbf{H}^T \mathbf{C}) + \frac{1}{3} \mathbf{H}^T \mathbf{H} \right]_{ij} u_j^{m+1} \right. \\ \left. - \left[ \frac{1}{2\Delta t} \mathbf{C}^T \mathbf{G} + \frac{1}{3} \mathbf{H}^T \mathbf{G} \right]_{ij} q_j^{m+1} - \left[ \frac{1}{2\Delta t} \mathbf{C}^T \mathbf{G} + \frac{1}{6} \mathbf{H}^T \mathbf{G} \right]_{ij} q_j^m \right. \\ \left. - \left[ \frac{1}{\Delta t^2} \mathbf{C}^T \mathbf{C} - \frac{1}{2\Delta t} (\mathbf{C}^T \mathbf{H} - \mathbf{H}^T \mathbf{C}) - \frac{1}{6} \mathbf{H}^T \mathbf{H} \right]_{ij} u_j^m \right\} = 0$$

- For  $X_i = q_i^{m+1}$ , i.e. a Dirichlet boundary condition prescribed at node  $i$ , the  $i$ th equation is of the form:

$$\sum_j \left\{ - \left[ \frac{1}{2\Delta t} \mathbf{G}^T \mathbf{C} + \frac{1}{3} \mathbf{G}^T \mathbf{H} \right]_{ij} u_j^{m+1} + \left[ \frac{1}{3} \mathbf{G}^T \mathbf{G} \right]_{ij} q_j^{m+1} \right. \\ \left. - \left[ \frac{1}{6} \mathbf{G}^T \mathbf{H} - \frac{1}{2\Delta t} \mathbf{G}^T \mathbf{C} \right]_{ij} u_j^m + \left[ \frac{1}{6} \mathbf{G}^T \mathbf{G} \right]_{ij} q_j^m \right\} = 0$$

The least squares scheme was applied by Singh and Kalra [8] to a number of transient heat conduction problems, and its performance compared to those of two-level schemes such as Crank-Nicolson, Galerkin and fully implicit. A quadratic rate of convergence was observed for the least squares scheme as compared, for instance, to the first-order convergence rate of the fully implicit scheme. This is offset, however, by higher computational costs.

Lahrmann and Haberland [9] derived a weighted time step solution procedure which optimizes the weighting factors  $\theta_u$  and  $\theta_q$  in equations (5.13) and (5.14). The basic idea of the scheme is to use a variable coefficient  $\theta (= \theta_u = \theta_q)$  dependent on the element size, time step and thermal diffusivity, in the form

$$\theta_i = \frac{Fo_i - 1 + e^{-Fo_i}}{Fo_i(1 - e^{-Fo_i})} \quad (5.21)$$

where  $Fo_i$  is the Fourier number at element  $i$ , i.e.  $Fo_i = k\Delta t/\Delta x_i^2$ .

The weighted time step scheme of Lahrmann and Haberland [9] has been implemented into FEM and DRBEM programs, and applied to several problems. They concluded that the method is unconditionally stable and accurate, providing very similar solutions for the FEM and the DRBEM.

## 5.4 Special $f$ Expansions

Up to this point, all DRM formulations discussed in this book have employed  $f$  expansions of the type shown in chapter 3, *i.e.*

$$f = 1 + r + r^2 + \dots + r^m \quad (5.22)$$

The above choice is in line with the behaviour of the fundamental solution to Laplace's equation, which is itself a function of  $r$  only, for two- and three-dimensional problems.

Some cases for which the expansion (5.22) is not the most appropriate have been reported in the literature. Two such cases which are related to diffusion problems will be discussed next: diffusion in axisymmetric regions [10] and in infinite regions [11].

### 5.4.1 Axisymmetric Diffusion

The fundamental solution to Laplace's equation in an axisymmetric region is of the form [4]

$$u^* = \frac{4K(m)}{(a+b)^{1/2}} \quad (5.23)$$

in which  $K$  is the complete elliptic integral of the first kind, and

$$a = R_i^2 + R_k^2 + (Z_i - Z_k)^2$$

$$b = 2R_i R_k$$

$$m = \frac{2b}{a+b}$$

where  $R$  and  $Z$  are cartesian coordinates in the generating plane.

From the above definitions, it can be seen that in this case  $u^*$  depends not only on  $r$  but also on the distance from the source and field points ( $i$  and  $k$ , respectively) to the axis of revolution. With this behaviour in mind, Wrobel *et al.* [10] adopted the following  $f$  expansion for axisymmetric diffusion problems

$$f = r \left( 1 - \frac{R_j}{4R_k} \right) \quad (5.24)$$

where  $R_j$  and  $R_k$  represent the distance from node  $j$  or node  $k$  to the axis of revolution, and  $r$  is the distance between  $j$  and  $k$ .

Since the above set of functions is clearly axisymmetric, the corresponding set of  $\hat{u}$  functions was also defined as axisymmetric, and such that

$$\nabla^2 \hat{u} = \frac{\partial^2 \hat{u}}{\partial R^2} + \frac{1}{R} \frac{\partial \hat{u}}{\partial R} + \frac{\partial^2 \hat{u}}{\partial Z^2} = f \quad (5.25)$$

The expressions of  $\hat{u}$  and  $\hat{q}$  are then of the form:

$$\hat{u} = \frac{r^3}{12} \quad (5.26)$$

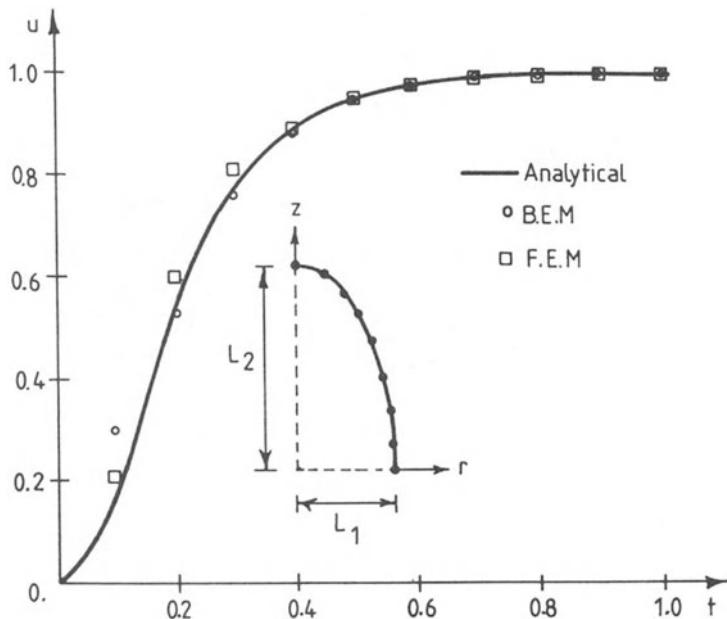


Figure 5.6: Temperature at Centre of Prolate Spheroid

$$\hat{q} = \frac{r^2}{4} \frac{\partial r}{\partial n} \quad (5.27)$$

Wrobel *et al.* [10] applied this formulation to study the temperature distribution inside a prolate spheroid initially at zero temperature, subject to a unit thermal shock at  $t = 0$ . The discretization with 8 constant boundary elements for one-half of the body, taking symmetry into consideration, is shown in figure 5.6; the numerical values adopted for the geometrical parameters were  $L_1 = 1$ ,  $L_2 = 2$ .

Results for the temperature at the centre point ( $r = z = 0$ ) are compared in figure 5.6 with an analytical solution [12] and a finite element solution obtained with three-dimensional quadratic isoparametric elements [13]. Both the DRBEM and the FEM analyses were performed with a time step value  $\Delta t = 0.025s$ .

Similarly to the problem described in section 5.3.5, the present one also has Dirichlet boundary conditions so that internal points are required to provide degrees of freedom for the function  $u$ . The results plotted in figure 5.6 were obtained with only 5 internal points, and show very good agreement with the analytical solution in spite of the reduced number of degrees of freedom of the analysis.

## 5.4.2 Infinite Regions

Loeffler and Mansur [11] presented an interesting application of the DRM to diffusion problems in infinite regions. The main distinction in this case is that, in order to fulfill the regularity conditions at infinity [4], making it possible to treat the problem

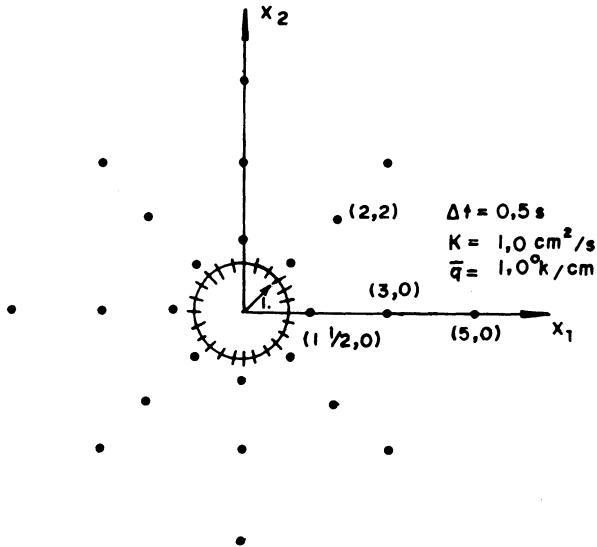


Figure 5.7: Definition of the Problem

by discretizing only the internal boundaries, the  $f$  expansion has to possess a decay which makes the integrals of  $\hat{u}$  and  $\hat{q}$  tend to zero along the boundary at infinity, *i.e.*

$$\lim_{r \rightarrow \infty} \int_{\Gamma_\infty} (q^* \hat{u} - u^* \hat{q}) d\Gamma_\infty = 0 \quad (5.28)$$

The expressions used by Loeffler and Mansur [11] are of the form

$$f = \frac{2C - r}{(r + C)^4}$$

$$\hat{u} = -\frac{2r + C}{2(r + C)^2}$$

$$\hat{q} = \frac{r}{(r + C)^3} \frac{\partial r}{\partial n}$$

in which  $C$  is a problem-dependent constant determined by the inequality

$$C \geq 50(k \Delta L t_t)^{1/3}$$

where  $k$  is the diffusivity coefficient,  $\Delta L$  is the length of the smallest element of the discretization and  $t_t$  is the total time span of the analysis.

Loeffler and Mansur [11] employed this formulation to study the temperature variation within an unbounded medium with a circular hole where a uniform heat flux is suddenly applied. Figure 5.7 shows the geometrical and physical characteristics of the problem, and the discretization with 24 boundary elements and 24 internal points. Results are plotted in figure 5.8 for the temperature variation at a point on the surface of the hole, compared to an analytical solution given in [5].

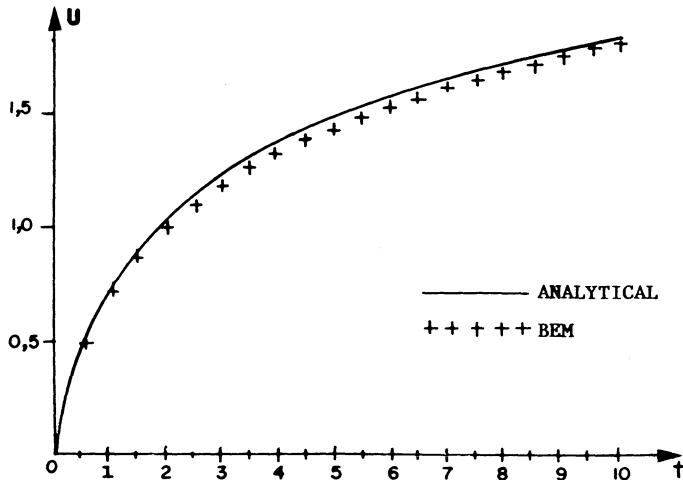


Figure 5.8: Temperature Variation at a Point on the Surface of the Hole

## 5.5 The Wave Equation

Consider the problem of a wave propagating in an elastic medium, described by equation (5.2), now written in the form

$$\nabla^2 u = \frac{1}{c^2} \ddot{u} \quad (5.29)$$

in which  $c$  is the wave celerity. In addition to boundary conditions, the dynamic character of this problem requires the specification of initial displacements and initial velocities, *i.e.*

$$u(x, y, t_0) = u_0(x, y) \quad (5.30)$$

$$\dot{u}(x, y, t_0) = \dot{u}_0(x, y) \quad (5.31)$$

The application of the DRM to wave propagation problems follows basically the same procedure as for diffusion problems, and produces a matrix equation of the form

$$\mathbf{H}u - \mathbf{G}q = \frac{1}{c^2} (\mathbf{H}\hat{U} - \mathbf{G}\hat{Q})\boldsymbol{\alpha} \quad (5.32)$$

The vector  $\boldsymbol{\alpha}$  can be related to the vector of accelerations by

$$\boldsymbol{\alpha} = \mathbf{F}^{-1} \ddot{\mathbf{u}} \quad (5.33)$$

Substituting the above into (5.32) leads to the equation

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{H}u = \mathbf{G}q \quad (5.34)$$

in which matrix  $\mathbf{M}$ , defined as

$$\mathbf{M} = -\frac{1}{c^2} \mathbf{S} = -\frac{1}{c^2} (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}}) \mathbf{F}^{-1} \quad (5.35)$$

is now interpreted as a mass matrix.

It may seem strange, at first, that the mass matrix for wave propagation problems is equal to the “heat capacity” matrix for diffusion problems (see expression 5.11), apart from a constant. However, this is consistent with the corresponding theories, and the same situation also arises in finite element analysis where both matrices result from integration of cross-products of interpolation functions.

The standard way to obtain the time response using equations like (5.34) is to use linear time integration schemes. Nardini and Brebbia [14] and Loeffler and Mansur [15] discuss a number of algorithms traditionally employed in finite element analysis like central difference approximations, Newmark and Houbolt schemes. They concluded that it is advantageous in DRM to employ the Houbolt scheme due to its introduction of artificial damping, which truncates the influence of higher modes in the response.

The Houbolt integration scheme is an implicit, unconditionally stable algorithm in which the acceleration is approximated in the form

$$\ddot{\mathbf{u}}^{m+1} = \frac{1}{\Delta t^2} (2\mathbf{u}^{m+1} - 5\mathbf{u}^m + 4\mathbf{u}^{m-1} - \mathbf{u}^{m-2}) \quad (5.36)$$

which is a backward-type finite difference formula with error of order  $O(\Delta t^2)$ . Writing equation (5.34) at time level  $(m+1)\Delta t$  and substituting the above results in the expression:

$$(2\mathbf{M} + \Delta t^2 \mathbf{H})\mathbf{u}^{m+1} - \Delta t^2 \mathbf{G}\mathbf{q}^{m+1} = 5\mathbf{M}\mathbf{u}^m - 4\mathbf{M}\mathbf{u}^{m-1} + \mathbf{M}\mathbf{u}^{m-2} \quad (5.37)$$

The above equation permits the calculation of the distribution of  $\mathbf{u}$  at time level  $(m+1)\Delta t$  by using the boundary conditions at that time and information from three previous time steps. This is normally obtained through a special starting procedure of lower order, in which the initial conditions (5.30) and (5.31) (*i.e.*  $\mathbf{u}^0$  and  $\dot{\mathbf{u}}^0$ ) are employed to calculate  $\mathbf{u}^1$  and  $\mathbf{u}^2$  [16].

Loeffler and Mansur [15] applied the above DRM formulation to study the propagation of longitudinal waves in an elastic rod fixed at one of its extremities and subjected to a Heaviside-type load on the other. The geometrical characteristics, loading and boundary conditions of the problem are depicted in figure 5.9.

Results are presented in figures 5.10 and 5.11 for displacements at point A and tractions at point B, respectively (see figure 5.9), compared to an analytical solution of the problem [17]. The discretization employed consisted of 36 constant boundary elements and 4 internal points, with a time step  $\Delta t = 0.5s$ . Loeffler and Mansur [15] divided the body into two subregions, and concluded that this had a beneficial effect in terms of accuracy. However, the most important factor appeared to be the control of the higher modes, an intrinsic property of Houbolt’s scheme [16].

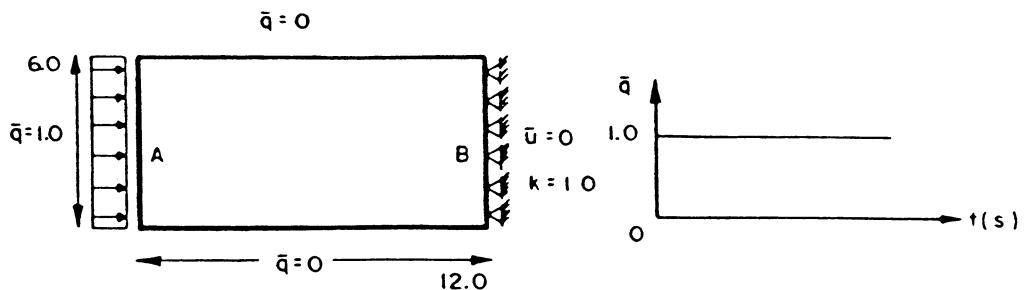


Figure 5.9: One-dimensional Rod under Heaviside-type Forcing Load

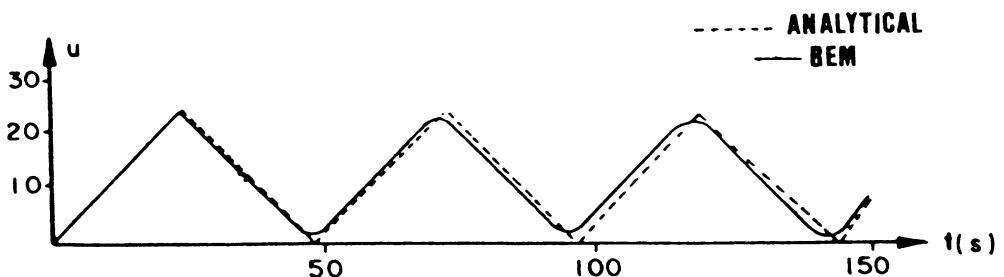


Figure 5.10: Time History of Displacements at Point A

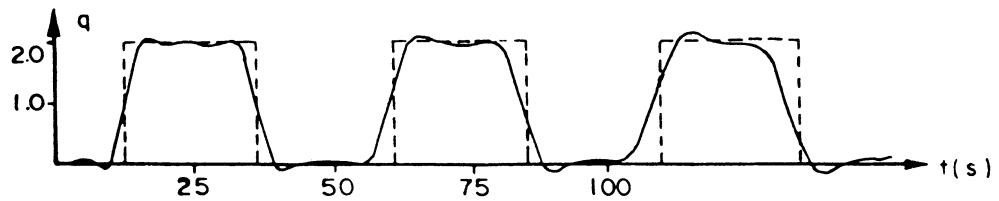


Figure 5.11: Time History of Traction at Point B

### 5.5.1 Infinite and Semi-Infinite Regions

The DRM formulation for the wave equation has been recently extended by Dai [18] to solve problems of waves propagating in an infinite or semi-infinite region. To this end, an artificial boundary  $\Gamma_\infty$  is introduced to truncate the infinite domain. In order to avoid the reflection of the waves at the artificial boundary the Sommerfeld radiation condition is applied, *i.e.*

$$\frac{\partial u}{\partial n} + \frac{1}{c} \dot{u} = 0 \quad (5.38)$$

which can be interpreted as a radiation damping condition which absorbs the wave propagating to  $\Gamma_\infty$ .

On the other hand, if a free surface exists, the boundary condition corresponding to a first-order approximation is:

$$\frac{\partial u}{\partial n} + \frac{1}{g} \ddot{u} = 0 \quad (5.39)$$

Taking the above two conditions into consideration, equation (5.34) may be partitioned in the form:

$$\begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \begin{Bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \ddot{u}_3 \end{Bmatrix} + \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = \begin{bmatrix} G_{11} & G_{12} & G_{13} \\ G_{21} & G_{22} & G_{23} \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \begin{Bmatrix} -\frac{1}{g} \ddot{u}_1 \\ -\frac{1}{g} \dot{u}_2 \\ q_3 \end{Bmatrix} \quad (5.40)$$

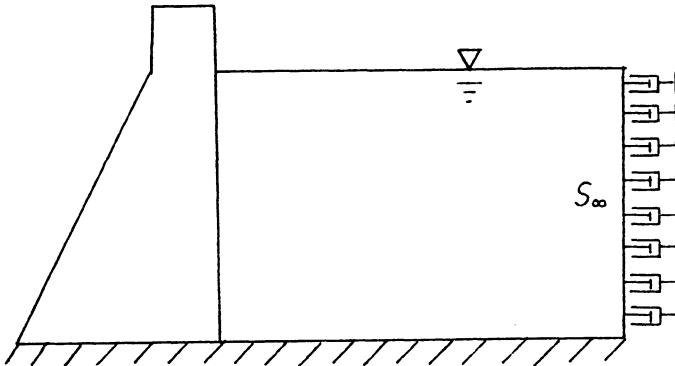


Figure 5.12: Geometry of Dam-Reservoir System

where  $\mathbf{u}_1$ ,  $\mathbf{u}_2$  and  $\mathbf{u}_3$  stand for values of  $u$  on the free surface, the artificial boundary and other boundaries, respectively. For the purpose of the application to be discussed next, it is assumed that the boundary condition on the solid boundaries is  $q_3 = \bar{q}_3$ . Thus, equation (5.40) can be rewritten in the form:

$$\overline{\mathbf{M}}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{H}\mathbf{u} = \mathbf{b} \quad (5.41)$$

with

$$\overline{\mathbf{M}} = \begin{bmatrix} \mathbf{M}_{11} + \frac{1}{g}\mathbf{G}_{11} & \mathbf{M}_{12} & \mathbf{M}_{13} \\ \mathbf{M}_{21} + \frac{1}{g}\mathbf{G}_{21} & \mathbf{M}_{22} & \mathbf{M}_{23} \\ \mathbf{M}_{31} + \frac{1}{g}\mathbf{G}_{31} & \mathbf{M}_{32} & \mathbf{M}_{33} \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} \mathbf{0} & \frac{1}{f}\mathbf{G}_{12} & \mathbf{0} \\ \mathbf{0} & \frac{1}{f}\mathbf{G}_{22} & \mathbf{0} \\ \mathbf{0} & \frac{1}{c}\mathbf{G}_{32} & \mathbf{0} \end{bmatrix}$$

$$\mathbf{b} = \{\mathbf{G}_{13}\bar{q}_3 \quad \mathbf{G}_{23}\bar{q}_3 \quad \mathbf{G}_{33}\bar{q}_3\}^T$$

where matrix  $\mathbf{C}$  may be called the radiation damping matrix.

Dai [18] applied the previous formulation to several problems, one of which is reproduced below. This concerns the study of the dynamic pressure on a rigid gravity dam due to horizontal ground acceleration. The geometry of the dam is given in figure 5.12. The material properties of water are wave speed  $c = 1430\text{m/s}$  and mass density  $\rho = 999.6\text{kg/m}^3$ . The depth of the reservoir is  $h = 160\text{m}$ .

The excitation frequency is taken as  $f = 1\text{Hz}$  and the infinite reservoir is truncated at a distance of 240m from the dam. Numerical and analytical results are compared in figure 5.13 where it can be seen that the BEM solutions give a very good estimate of the dynamic pressure. Moreover, it is obvious that the transient solution approaches the steady one as time progresses, showing the effect of the radiation damping.

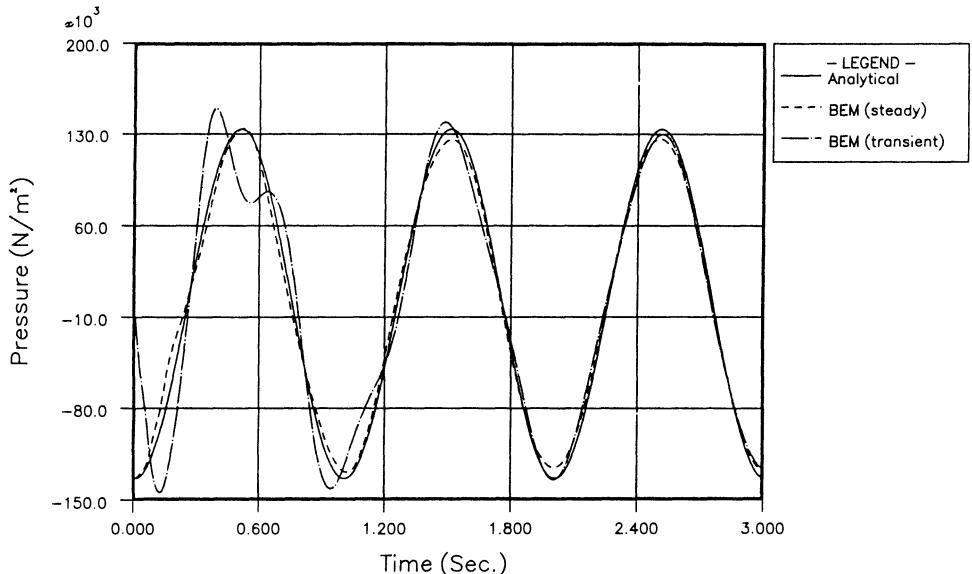


Figure 5.13: Time History of Pressure at Bottom of Dam

## 5.6 The Transient Convection-Diffusion Equation

This section deals with transient convection-diffusion problems governed by equation (5.3), rewritten here in the form

$$\nabla^2 u = \frac{v_x}{D} \frac{\partial u}{\partial x} + \frac{v_y}{D} \frac{\partial u}{\partial y} - \frac{K}{D} u + \frac{1}{D} \dot{u} \quad (5.42)$$

The function  $u$  is generally associated with the concentration of a substance dissolved in a solute. The above equation describes its transport and dispersion, which depend on the velocity field (with components  $v_x, v_y$ ) and the dispersion coefficient  $D$  (assuming the medium is homogeneous and isotropic). Coefficient  $K$  relates to a first-order chemical reaction. Equation (5.42) can also be used to describe the temperature field within a moving solid.

It is well-known that finite difference and finite element solutions of the convection-diffusion equation present numerical problems of oscillation and damping. On the other hand, boundary element solutions appear to be relatively free from these problems, as shown by Brebbia and Skerget [19] and others. These formulations employed the fundamental solution of the diffusion equation and included the convective effects by domain discretization and iteration.

Here, the problem will initially be treated by using the fundamental solution of Laplace's equation. This is done by combining the ideas previously derived in sections 4.2 and 5.2 for convective and diffusive problems, respectively. Thus, the DRM

formulation for equation (5.42) produces a matrix equation of the form

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{S} \left[ \left( \frac{v_x}{D} \frac{\partial \mathbf{F}}{\partial x} + \frac{v_y}{D} \frac{\partial \mathbf{F}}{\partial y} \right) \mathbf{F}^{-1} \mathbf{u} - \frac{K}{D} \mathbf{u} + \frac{1}{D} \dot{\mathbf{u}} \right] \quad (5.43)$$

in which

$$\mathbf{S} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1}$$

Calling, as before,

$$\mathbf{C} = -\frac{1}{D} \mathbf{S}$$

$$\mathbf{R} = \mathbf{S} \left( \frac{v_x}{D} \frac{\partial \mathbf{F}}{\partial x} + \frac{v_y}{D} \frac{\partial \mathbf{F}}{\partial y} \right) \mathbf{F}^{-1}$$

and substituting into (5.43), one obtains

$$\mathbf{C}\mathbf{u} + (\mathbf{H} - \mathbf{R} - K\mathbf{C})\mathbf{u} = \mathbf{G}\mathbf{q} \quad (5.44)$$

The above equation is similar in form to (5.12), and the same standard type of discrete time-marching procedure can be employed in its solution.

The present formulation was applied to a transient convection-diffusion problem in a rectangular region of cross-section dimensions  $6 \times 1.4$ , discretized with 38 linear boundary elements (figure 5.14). The initial  $u$  distribution is constant and equal to zero, and the boundary conditions are  $u = 300$  on the left face,  $q = 0$  on the right face and on the remaining faces parallel to the  $x$ -axis. The variation of boundary conditions at corners was taken into account by allowing the fluxes to be discontinuous as done in [7]. The values of  $D$ ,  $K$ ,  $v_x$  and  $v_y$  are 1.0, 1.0, 1.6 and 0, respectively.

Results are presented in figure 5.15 for the variation of  $u$  along  $x$ , obtained using  $\theta_u = \theta_q = 0.5$  and  $\Delta t = 0.025$ . This small time step value was necessary to get a better picture of the transient behaviour of  $u$  and not because of stability considerations. The results agree well with the analytical solution of [20]. An alternative DRM treatment of the problem which produces increased accuracy uses the fundamental solution of the steady-state convection-diffusion equation, as will be shown in chapter 6. However, the formulation presented in this section is of a general nature and can be used for problems with variable velocity fields and/or dispersion coefficient.

## 5.7 Non-Linear Problems

This section presents applications of the DRM to some non-linear problems. For simplicity, only non-linear features which arise in connection with the diffusion equation will be considered but the formulation is general enough to be extended to other types of problems.

Restricting the formulation to heat conduction problems, the most important types of non-linearities which appear in practical engineering situations can be classified as follows:

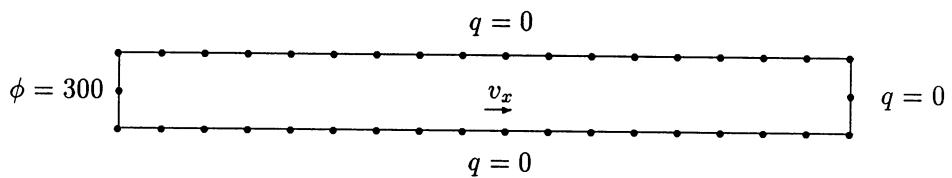
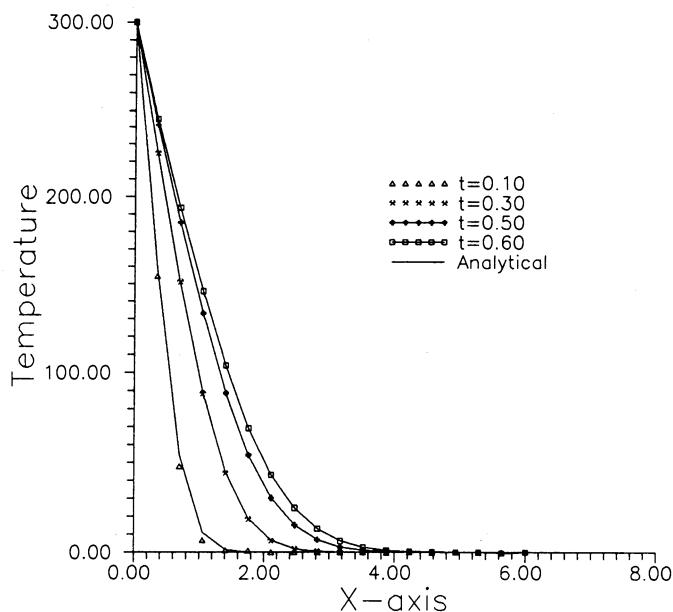


Figure 5.14: Transient Convection-Diffusion Problem: Geometry and Discretization

Figure 5.15: Transient Variation of  $u$  along the  $x$ -Axis

- Non-linear materials, *i.e.* temperature-dependent diffusivity coefficient;
- Non-linear boundary conditions, *e.g.* due to heat radiation;
- Non-linear distributed sources, as in the case of spontaneous ignition;
- Moving interface problems, *e.g.* due to phase change.

The first three situations will be considered in this section. The application of boundary elements to phase change problems has been dealt with before using time-dependent fundamental solutions (for instance in [21]) and is a very interesting area of research with the DRM.

### 5.7.1 Non-Linear Materials

The general form of the diffusion equation as applied to two-dimensional heat conduction in an isotropic medium without internal heat generation is

$$\frac{\partial}{\partial x} \left( K \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( K \frac{\partial u}{\partial y} \right) = \rho c \frac{\partial u}{\partial t} \quad (5.45)$$

in which  $u$  is the temperature,  $K$  the thermal conductivity,  $\rho$  is density and  $c$  the specific heat of the material. Assuming that  $K$ ,  $\rho$  and  $c$  are all temperature-dependent, the above equation can be expanded in the form

$$K \nabla^2 u - \rho c \frac{\partial u}{\partial t} = - \frac{dK}{du} \left[ \left( \frac{\partial u}{\partial x} \right)^2 + \left( \frac{\partial u}{\partial y} \right)^2 \right] \quad (5.46)$$

where the non-linear terms appear explicitly on the right side of the equation.

Steady-state non-linear problems with temperature-dependent conductivity have been treated by Onishi and Kuroki [22] in the above form by considering the right side of (5.46) as a non-linear source term which was accounted for by domain integration, in an iterative way.

A more elegant and efficient formulation can be derived by using Kirchhoff's transformation [5],[23], as described in section 4.4.3. Defining a new dependent variable  $U = U(u)$ , such that

$$\frac{dU}{du} = K(u) \quad (5.47)$$

or, in integral form,

$$U = \int_{u_a}^u K(u) du \quad (5.48)$$

equation (5.45) in the new variable becomes

$$\nabla^2 U = \frac{1}{k} \frac{\partial U}{\partial t} \quad (5.49)$$

in which  $k = k(u) = K/\rho c$  is the thermal diffusivity.

It can be noticed that equation (5.49) still contains a temperature-dependent diffusivity coefficient. Thus, a further transformation is needed. Following [6] and [24], one can write  $k = k(x, y, t)$  since  $u$  is a continuous function of the coordinates as well as time. A new variable  $\tau$  can then be defined by the relation

$$\tau = \int_0^t k(x, y, t) dt \quad (5.50)$$

The partial differentiation of  $\tau$  with respect to  $t$  gives

$$\frac{\partial \tau}{\partial t} = k \quad (5.51)$$

Substituting the above into (5.49), one finally obtains

$$\nabla^2 U = \frac{\partial U}{\partial \tau} \quad (5.52)$$

Equation (5.52) can be solved by the Dual Reciprocity BEM as described in section 5.2. However, since the modified time variable  $\tau$  is now a function of position (*i.e.* the problem is still non-linear), an iterative solution process has to be employed. A Newton-Raphson algorithm appropriate to the problem was developed by Wrobel and Brebbia and applied in conjunction with the DRM. The algorithm is described in detail in [6], and only the main ideas will be reviewed here.

The application of the DRM to equation (5.52) produces a system of equations similar to (5.17), which can be written as

$$(2\bar{C} + H)\mathbf{U}^{m+1} - 2G\mathbf{Q}^{m+1} = (2\bar{C} - H)\mathbf{U}^m \quad (5.53)$$

In the above equation,  $\mathbf{U}$  and  $\mathbf{Q}$  represent values in the transform space and matrix  $\bar{C}$  ( $\bar{C}_{ij} = C_{ij}/\Delta\tau_j$ ) contains step values of the modified time variable at each node, *i.e.*

$$\Delta\tau_j = k_j \Delta t \quad (5.54)$$

at node  $j$ .

The algorithm employed in the solution of the non-linear system (5.53) is of the Newton-Raphson type, following the idea of Wrobel and Azevedo [25] for steady-state problems.

Consider  $\psi(\mathbf{X}^{m+1})$  as a residual function which should be zero when  $\mathbf{X}^{m+1}$  is the exact solution at time  $(m+1)\Delta t$ :

$$\psi(\mathbf{X}^{m+1}) = (2\bar{C} + H)\mathbf{U}^{m+1} - 2G\mathbf{Q}^{m+1} - (2\bar{C} - H)\mathbf{U}^m \quad (5.55)$$

The Newton-Raphson scheme can be formulated by expanding the residual function as a Taylor series about an approximate solution:

$$\psi(\mathbf{X}) = \psi(\mathbf{X}_n) + \left[ \frac{\partial \psi}{\partial \mathbf{X}} \right]_n (\mathbf{X} - \mathbf{X}_n) \quad (5.56)$$

where second and higher-order terms have been neglected. The subscript  $n$  means the approximate solution at the  $n$ th iteration, and the superscript  $m+1$  was dropped for simplicity.

Considering that  $\mathbf{X}$  is the exact solution, i.e.  $\psi(\mathbf{X}) = \mathbf{0}$ , the following expression is obtained:

$$\mathbf{J}_n \Delta \mathbf{X}_{n+1} = -\psi(\mathbf{X}_n) \quad (5.57)$$

in which  $\mathbf{J} = \partial\psi/\partial\mathbf{X}$  is the tangent or Jacobian matrix and  $\Delta\mathbf{X}$  is the vector of increments.

Starting from the linear solution it is possible to determine, for each iteration, the updated solution through the incremental expression

$$\mathbf{X}_{n+1} = \mathbf{X}_n + \Delta\mathbf{X}_{n+1} \quad (5.58)$$

and the iteration process proceeds until the residual vector is sufficiently small.

The coefficients of the tangent matrix are computed by using the expression

$$J_{ij}^{m+1} = \frac{\partial\psi_i^{m+1}}{\partial X_j^{m+1}} \quad (5.59)$$

According to the boundary conditions of the problem, we have:

i. When  $X_{ij}^{m+1} = Q_{ij}^{m+1}$  :

$$J_{ij}^{m+1} = -2G_{ij} \quad (5.60)$$

ii. When  $X_{ij}^{m+1} = U_{ij}^{m+1}$  :

$$\begin{aligned} J_{ij}^{m+1} &= \frac{2}{\Delta\tau_j} C_{ij} + H_{ij} - 2G_{ij} \frac{\partial Q_j^{m+1}}{\partial U_j^{m+1}} \\ &\quad + 2C_{ij}(U_j^{m+1} - U_j^m) \frac{\partial}{\partial U_j^{m+1}} (\Delta\tau_j)^{-1} \end{aligned} \quad (5.61)$$

The expression for the derivative of  $(\Delta\tau_j)^{-1}$  with respect to  $U_j^{m+1}$  is given in [6]. The derivative of  $Q_j$  with respect to  $U_j$  depends on the type of boundary condition at node  $j$ , and its expression is discussed in the next subsection.

The iteration cycle will then consist of the following steps:

- i. Solve (5.52) for a constant  $\Delta\tau = \Delta t$ , finding the linear solution;
- ii. With the value of  $U$  at all points, an inverse transformation is performed to find values of  $u$ ;
- iii. The conductivity, density and specific heat curves provide, for each value of  $u$ , updated values of  $K$ ,  $\rho$  and  $c$ ;
- iv. Values of  $\Delta\tau$  can now be calculated at each point through expression (5.54);
- v. The tangent matrix and residual vector are updated, and a new solution carried out.

The iteration algorithm proceeds until a certain specified tolerance for the relative norm of increments  $\Delta \mathbf{X}$  is reached. With the converged solution for  $\mathbf{U}$  and  $\mathbf{Q}$ , the process can then be advanced to the next time step.

The above formulation has been employed to study several heat conduction problems [6]. In what follows, results are presented for the temperature distribution along the thickness of a wall with non-linear material properties. The wall is 20 cm long, 1 cm high and is initially at 100°C. The temperature of the left-end surface is suddenly raised to 200°C and kept at this value for 10 s, after which it is decreased to 100°C again; the temperature of the right-end surface is kept at 100°C, while the other surfaces are insulated. The thermal conductivity is  $K = 2 + 0.01u$  W/(cm °C) and heat capacity  $\rho c = 8$  J/(cm<sup>3</sup> °C).

DRM results using discretizations of 22 equal quadratic (labelled DRM-Q) or 44 equal constant boundary elements (DRM-C), taking into account symmetry with respect to the x-axis, are compared in tables 5.4 and 5.5 with a finite element solution obtained with 20 equal linear elements [26]. The agreement of results is very good. The average number of iterations of the boundary element solution was 4, compared with 3 using finite elements. The time step was 1 second in both cases.

x	FEM	DRM-Q	DRM-C
0	200.00	200.00	200.00
1	176.16	174.86	175.29
2	153.21	151.03	151.48
3	133.47	131.33	131.74
4	118.60	117.32	117.63
5	108.98	108.74	108.94
6	103.72	104.14	104.27
7	101.29	101.91	102.01
8	100.37	100.87	100.97
9	100.08	100.39	100.50
10	100.01	100.14	100.27

Table 5.4: Temperature Variation (°C) along Wall Thickness at Time t=10 s

### 5.7.2 Non-Linear Boundary Conditions

Another interesting case of non-linearity which can be treated with the present formulation is that of non-linear boundary conditions. The most common types of boundary conditions associated with the present problem are:

- i. Prescribed temperature :  $u = \bar{u}$
- ii. Prescribed flux :  $q = \bar{q}$
- iii. Convection :  $q = h(u - u_c)$

x	FEM	DRM-Q	DRM-C
0	100.00	100.00	100.00
1	128.53	130.46	130.15
2	139.97	138.70	138.95
3	136.95	132.01	132.47
4	124.72	121.29	121.71
5	114.40	112.37	112.64
6	107.18	106.56	106.71
7	103.24	103.27	103.36
8	101.29	101.56	101.62
9	100.45	100.72	100.77
10	100.13	100.33	100.36

Table 5.5: Temperature Variation ( $^{\circ}\text{C}$ ) along Wall Thickness at Time  $t=13$  s

iv. Radiation :  $q = \sigma\epsilon(u^4 - u_r^4)$

in which the heat flux is now defined as  $q = -K\partial u/\partial n$ ,  $n$  is the outward unit normal vector,  $h$  is the heat transfer coefficient,  $u_c$  is the temperature of the medium surrounding the convective boundaries,  $\sigma$  is the Stefan-Boltzmann constant and  $\epsilon$  is the radiative interchange factor between the surface and the exterior ambient at temperature  $u_r$ . After applying Kirchhoff's transformation, the following corresponding expressions are obtained:

- i. Prescribed temperature :  $U = \bar{U} = T[\bar{u}]$
- ii. Prescribed flux :  $Q = \frac{\partial U}{\partial n} = \bar{Q} = K\frac{\partial u}{\partial n} = -\bar{q}$
- iii. Convection :  $Q = -h(u - u_c)$
- iv. Radiation :  $Q = -\sigma\epsilon(u^4 - u_r^4)$

It can be noticed that the convection and radiation conditions are non-linear in the transform space, since  $u = T^{-1}[U]$  is no longer the primary unknown.

The expression of the derivative of  $Q_j$  with respect to  $U_j$  in equation (5.61) depends on the type of boundary condition at node  $j$ . For prescribed heat flux, the derivative is obviously zero since  $Q$  does not depend on  $U$ . Boundary conditions of the convective and radiative types, however, produce the following expressions:

- Convection :

$$\frac{\partial Q_j^{m+1}}{\partial U_j^{m+1}} = -h \frac{\partial u}{\partial U} = -\frac{h}{K_j}$$

- Radiation :

$$\frac{\partial Q_j^{m+1}}{\partial U_j^{m+1}} = -\frac{4\sigma\epsilon u_j^3}{K_j}$$

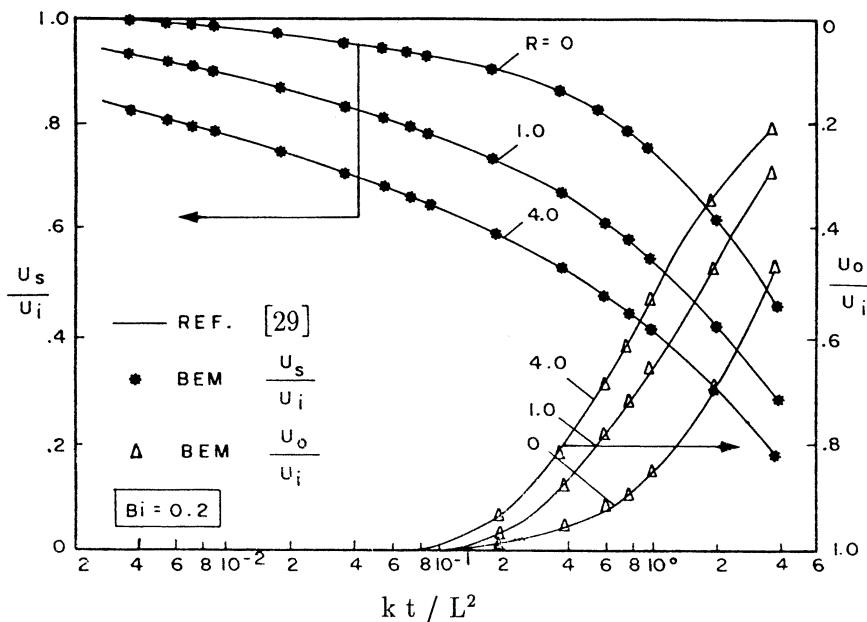


Figure 5.16: Transient Temperature Results for a Slab with Convection and Radiation at Surfaces

Another type of non-linearity may appear if the region under consideration is made up of piecewise homogeneous subregions of different materials. Enforcement of the continuity condition for the temperature along the interface between subregions produces a discontinuity in the transformed variable  $U$ , the values of which have to be adjusted so that the physical requirement of temperature continuity is satisfied at the final stage. This feature can also be treated by the same Newton-Raphson scheme, as discussed in [27].

Wrobel and Brebbia [28] employed the DRM to analyse a slab of width  $2L$ , initially at a uniform temperature  $u_i$ , the surfaces of which are suddenly exposed to simultaneous convection and radiation into a medium at  $0^\circ\text{F}$ . The slab has linear thermophysical properties, so the solution is carried out directly in the real space.

Results for the surface temperature  $u_s$  and the temperature  $u_0$  at the centre of the slab are presented in figure 5.16, in terms of dimensionless values. The parameters employed in the analysis are  $k = 1.0 \text{ in}^2/\text{h}$ , Biot number  $Bi = hL/K = 0.2$ , and  $R = \sigma\epsilon u_i^3 L/K = 0, 1$  and  $4$ ;  $R = 0$  corresponds to pure convection, representing a linear boundary condition.

The results obtained using DRM with 22 equal constant boundary elements compare well with those of Haji-Sheik and Sparrow [29], who used a Monte Carlo method.

### 5.7.3 Spontaneous Ignition: Transient Case

The problem of spontaneous ignition has already been discussed in section 4.4.2 for the steady-state case. Its mathematical description is given by a diffusion equation with a nonlinear reaction-heating (source) term which can be written, after some approximations, in the form

$$\nabla^2 u + \gamma e^u = \frac{1}{k} \frac{\partial u}{\partial t} \quad (5.62)$$

The so-called criticality problem, consisting of finding the value of  $\gamma$  above which spontaneous ignition will occur for a given body, was dealt with in section 4.4.2. This requires the solution of a steady-state version of equation (5.62) in which  $\partial u / \partial t$  is set to zero and the value of  $\gamma$  iterated until convergence is no longer achieved. Here, the formulation will be extended to analyse the full, transient problem, once the critical value of  $\gamma$  has been determined.

Thus, in equation (5.62), the value of  $\gamma$  is known and the solution provides the distribution of  $u$  (the temperature) all over the body.

Equation (5.62) can be interpreted as a Poisson's equation with the following non-homogeneous term

$$\nabla^2 u = b(x, y, u, t) = \frac{1}{k} \frac{\partial u}{\partial t} - \gamma e^u \quad (5.63)$$

Comparing the above with equation (5.5), it can be seen that the vector  $\alpha$  in (5.9) will now be given by

$$\alpha = \mathbf{F}^{-1} \left( \frac{1}{k} \dot{\mathbf{u}} - \gamma \mathbf{b}_1 \right) \quad (5.64)$$

in which  $\mathbf{b}_1$  is the vector of nodal values of  $e^u$ .

The system of equations resulting from the application of the DRM, equivalent to (5.10), becomes in the present case

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1} \left( \frac{1}{k} \dot{\mathbf{u}} - \gamma \mathbf{b}_1 \right) \quad (5.65)$$

Denoting, as before,

$$\mathbf{S} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{Q}})\mathbf{F}^{-1} \quad (5.66)$$

equation (5.65) can be written in the form

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{q} = \mathbf{S} \left( \frac{1}{k} \dot{\mathbf{u}} - \gamma \mathbf{b}_1 \right) \quad (5.67)$$

Employing the same two-level time integration scheme as in section 5.2 with  $\theta_u = 0.5$  and  $\theta_q = 1.0$ , an equation similar to (5.17) is obtained, i.e.

$$\left( \frac{2}{\Delta t} \mathbf{C} + \mathbf{H} \right) \mathbf{u}^{m+1} - 2\mathbf{G}\mathbf{q}^{m+1} = \left( \frac{2}{\Delta t} \mathbf{C} - \mathbf{H} \right) \mathbf{u}^m - 2\gamma \mathbf{S}\mathbf{b}_1 \quad (5.68)$$

in which  $\mathbf{C}$  is the capacity matrix,  $\mathbf{C} = -1/k \mathbf{S}$ . This non-linear, transient system can be solved for a given value of  $\gamma$  by marching in time, with iteration at each time step.

For the problem analysed, the boundary values of  $\mathbf{u}$  are all prescribed, so that system (5.68) is solved for the boundary values of  $\mathbf{q}$  and internal values of  $\mathbf{u}$ .

Within a time step  $(m + 1)\Delta t$ , at iteration  $n$ , vector  $\mathbf{u}_n^{m+1}$  is calculated and compared with  $\mathbf{u}_{n-1}^{m+1}$ . If they coincide to a given accuracy,  $\mathbf{u}_n^{m+1}$  is the solution for this time level and the process is repeated successively for all time levels. At iteration  $n$ , the nodal values of  $b_1$  are given by  $\exp[(\mathbf{u}_n^{m+1} + \mathbf{u}^m)/2]$  in which  $\mathbf{u}^m$  is the vector of converged values of  $\mathbf{u}$  at time  $m\Delta t$ .

Given the above, the right side of (5.68) is known for any iteration. Introducing the boundary conditions and reordering, the system can be solved by standard Gauss elimination.

The above technique was applied in [30] to study the problem of spontaneous ignition of a long cylinder of unit radius, initially at temperature  $T_0$  at all points, exposed at time  $t = 0$  to an environment at temperature  $T_a$ . The cylinder is of a uniform isotropic reactive material, the ignition temperature of which is  $T_m$ . The Dirichlet boundary condition  $T = T_a$  is imposed at all boundary nodes. The critical value of  $\gamma$  is  $\gamma_c = 2.0$  in this case [31].

The cylinder was discretized with 16 linear boundary elements and 19 internal nodes were spaced at intervals of 0.1m along a diagonal, as shown in figure 5.17. The numerical values of the physical parameters for RDX taken from [32] are

- $k = 7.778 \times 10^{-4} \text{ cm}^2/\text{s}$

- $T_m = 425 \text{ K}$

- $T_0 = 25^\circ\text{C} = 298 \text{ K}$

- $T_a = 400 \text{ K}$

- $E = 47500 \text{ kcal/M}$

- $R = 1.987 \text{ cal/(M K)}$

such that  $T = (u + 59.76)/0.1494$  [30]. A variable time step was used because this process has varying stability characteristics. Initially a large  $\Delta t$  may be used, but this must be rapidly reduced as ignition nears. The value of  $\Delta t$  was altered at each time step to maintain the average temperature change at all interior nodes between 5° and 20°.

Results are shown in figure 5.18 for the case  $\gamma = 1$ . Since this value of  $\gamma$  is smaller than  $\gamma_c$  the time step continually increases and the process converges to a steady-state solution with  $T_a < T < T_m$  at all internal points.

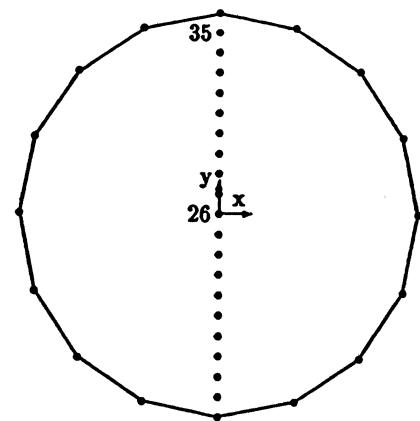


Figure 5.17: Discretization of Cylinder of Unit Radius

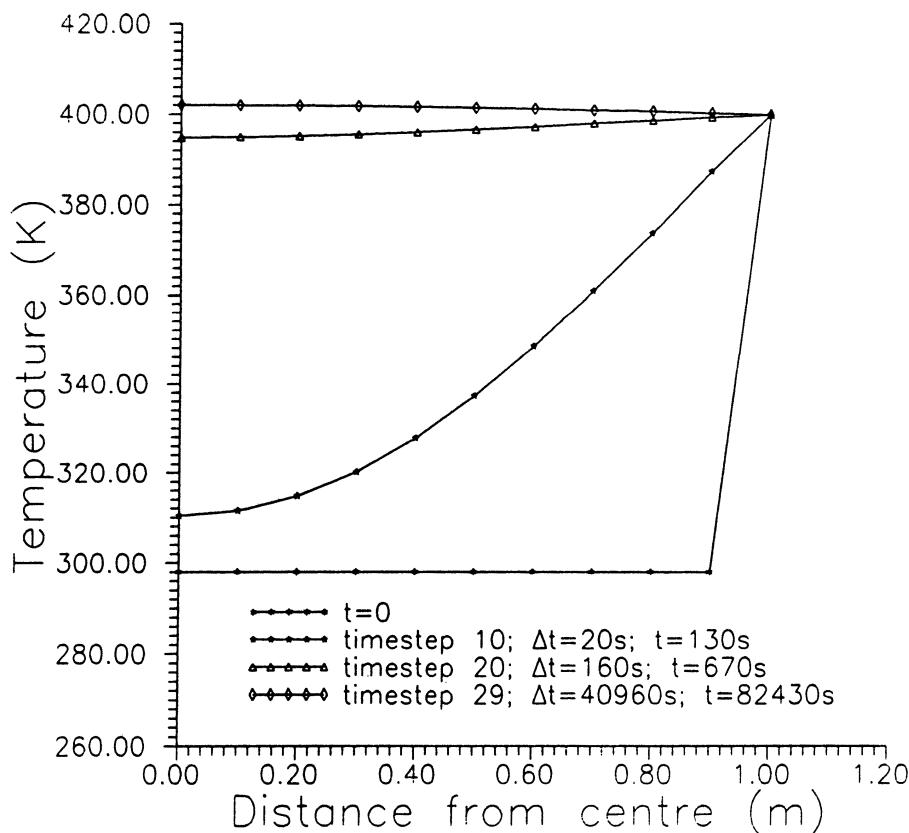


Figure 5.18: Temperature Distribution for  $\gamma = 1$

Results in figure 5.19 are for  $\gamma = 4$ . In this case it can be seen that ignition is first reached at the center of the cylinder.

For cases with higher values of  $\gamma$  the ignition point moves out from the center towards the outer surface and the elapsed time before ignition reduces. This situation is depicted in figure 5.20 for  $\gamma = 50$  and summarized in table 5.6. The results presented are very similar to those obtained using theoretical considerations in [32].

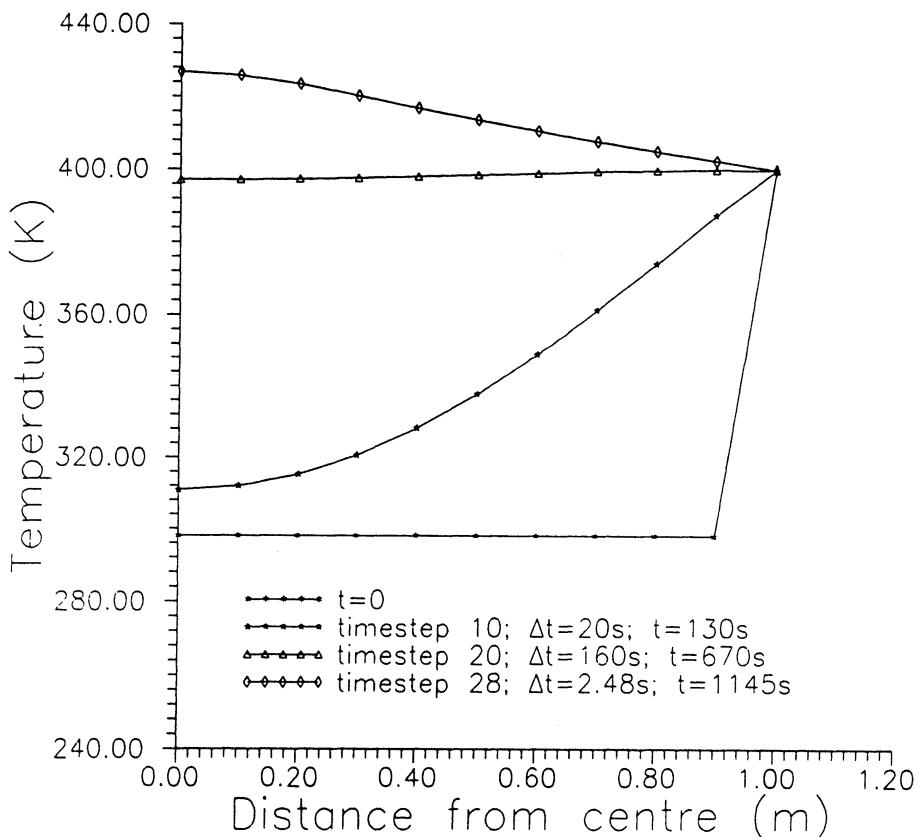
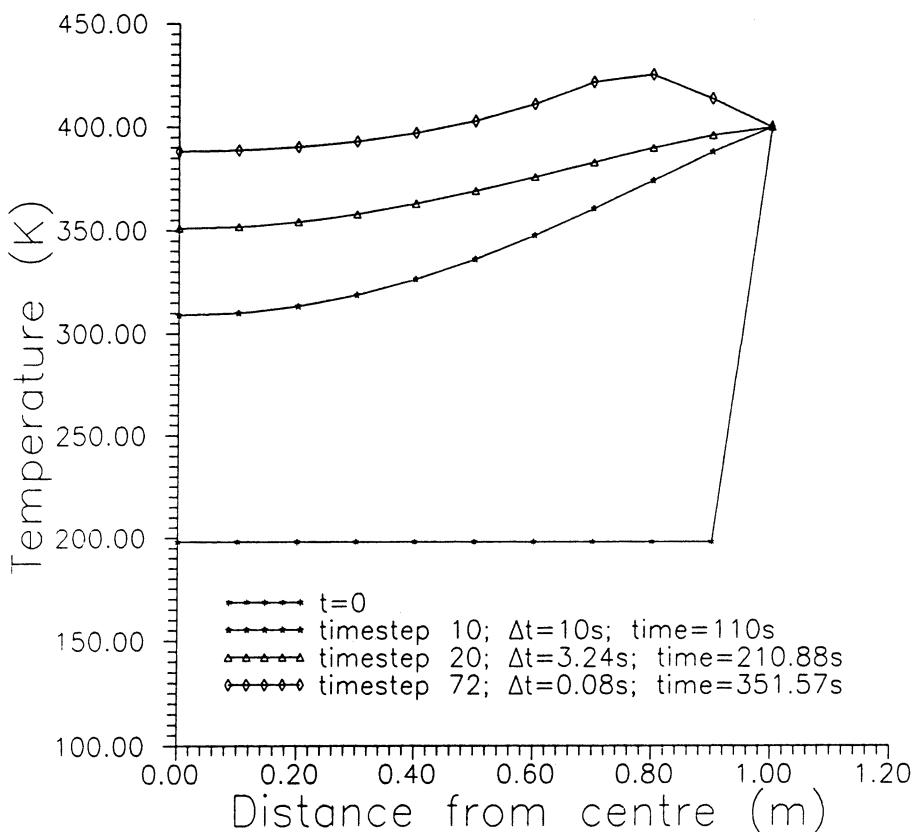


Figure 5.19: Temperature Distribution for  $\gamma = 4$

$\gamma$	ignition time (s)	ignition node	coordinate
1.0	—	—	max. temp. 402 K
1.9	—	—	max. temp. 405 K
2.1	5884	26	$y = 0$
4.0	1145	26	$y = 0$
10.0	723	26	$y = 0$
20.0	555	31	$y = 0.5$
50.0	351	34	$y = 0.8$
200.0	129	35	$y = 0.9$

Table 5.6: Ignition Time and Node for Increasing  $\gamma$ Figure 5.20: Temperature Distribution for  $\gamma = 50$

## 5.8 References

1. Wrobel, L. C., Brebbia, C. A. and Nardini, D.  
The Dual Reciprocity Boundary Element Formulation for Transient Heat Conduction, in *Finite Elements in Water Resources VI*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
2. Partridge, P. W. and Brebbia, C. A.  
The BEM Dual Reciprocity Method for Diffusion Problems, in *Computational Methods in Water Resources VIII*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1990.
3. Bruch, J. C. and Zyvoloski, G.  
Transient Two-dimensional Heat Conduction Problems Solved by the Finite Element Method, *International Journal for Numerical Methods in Engineering*, Vol. 8, pp. 481-494, 1974.
4. Brebbia, C. A., Telles, J. C. F. and Wrobel, L. C.  
Boundary Element Techniques, Springer-Verlag, Berlin and New York, 1984.
5. Carslaw, H. S. and Jaeger, J. C.  
Conduction of Heat in Solids, 2nd edn, Clarendon Press, Oxford, 1959.
6. Wrobel, L. C. and Brebbia, C. A.  
The Dual Reciprocity Boundary Element Formulation for Non-Linear Diffusion Problems, *Computer Methods in Applied Mechanics and Engineering*, Vol. 65, pp. 147-164, 1987.
7. Brebbia, C. A. and Dominguez, J.  
Boundary Elements : An Introductory Course, Computational Mechanics Publications, Southampton, and McGraw-Hill, New York, 1989.
8. Singh, K. M. and Kalra, M. S.  
A Least Squares Time Integration Scheme for the Dual Reciprocity BEM in Transient Heat Conduction, in *Numerical Methods in Thermal Problems*, Vol. VI, Part 1, Pineridge Press, Swansea, 1989.
9. Lahrmann, A. and Haberland, C.  
Comparative Application of Finite Element and Boundary Element Methods to Heat Transfer in Structures, in *Numerical Methods in Thermal Problems*, Vol. VII, Part 1, Pineridge Press, Swansea, 1991.
10. Wrobel, L. C., Telles, J. C. F. and Brebbia, C. A.  
A Dual Reciprocity Boundary Element Formulation for Axisymmetric Diffusion Problems, in *Boundary Elements VIII*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
11. Loeffler, C. F. and Mansur, W. J.  
Dual Reciprocity Boundary Element Formulation for Potential Problems in Infinite Domains, in *Boundary Elements X*, Vol. 2, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.

12. Haji-Sheik, A. and Sparrow, E. M.  
Transient Heat Conduction in a Prolate Spheroidal Solid, *Journal of Heat Transfer, ASME*, Vol. 88C, pp 331-333, 1966.
13. Zienkiewicz, O. C. and Parekh, C. J.  
Transient Field Problems: Two-Dimensional and Three-Dimensional Analysis by Isoparametric Finite Elements, *International Journal for Numerical Methods in Engineering*, Vol. 2, pp 61-71, 1970.
14. Nardini, D. and Brebbia, C. A.  
Boundary Integral Formulation of Mass Matrices for Dynamic Analysis, in *Topics in Boundary Element Research*, Vol. 2, Springer-Verlag, Berlin and New York, 1985.
15. Loeffler, C. F. and Mansur, W. J.  
Analysis of Time Integration Schemes for Boundary Element Applications to Transient Wave Propagation Problems, in *BETECH 87*, Computational Mechanics Publications, Southampton, 1987.
16. Bathe, K. J.  
Finite Element Procedures in Engineering Analysis, Prentice-Hall, New Jersey, 1982.
17. Miles, J. W.  
Modern Mathematics for the Engineer, Ed. E.F. Beckenbach, McGraw-Hill, London, 1961.
18. Dai, D. N.  
An Improved Boundary Element Formulation for Wave Propagation Problems, *Engineering Analysis*, in press.
19. Brebbia, C. A. and Skerget, L.  
Time Dependent Diffusion Convection Problems using Boundary Elements, in *Numerical Methods in Laminar and Turbulent Flow III*, Pineridge Press, Swansea, 1983.
20. Van Genuchten, M. Th. and Alves, W. J.  
Analytical Solutions of the One-dimensional Convective-Dispersive Solute Transport Equation, Technical Bulletin 1661, US Department of Agriculture, Riverside, California, USA, 1982.
21. Wrobel, L. C.  
A Boundary Element Solution to Stefan's Problem, in *Boundary Elements V*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1983.
22. Onishi, K. and Kuroki, T.  
Boundary Element Method in Singular and Nonlinear Heat Transfer, in *Boundary Element Methods in Engineering*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1982.
23. Ames, W. F.  
Nonlinear Partial Differential Equations in Engineering, Academic Press, New York, 1965.

24. Kadambi, V. and Dorri, B.  
Solution of Thermal Problems with Nonlinear Material Properties by the Boundary Element Method, in *BETECH 85*, Computational Mechanics Publications, Southampton, 1985.
25. Wrobel, L. C. and Azevedo, J. P. S.  
A Boundary Element Analysis of Nonlinear Heat Conduction, in *Numerical Methods in Thermal Problems IV*, Pineridge Press, Swansea, 1985.
26. Orivuori, S.  
Efficient Method for Solution of Nonlinear Heat Conduction Problems, *International Journal for Numerical Methods in Engineering*, Vol. 14, pp. 1461-1476, 1979.
27. Azevedo, J. P. S. and Wrobel, L. C.  
Non-linear Heat Conduction in Composite Bodies: A Boundary Element Formulation, *International Journal for Numerical Methods in Engineering*, Vol. 26, pp. 19-38, 1988.
28. Wrobel, L. C. and Brebbia, C. A.  
Boundary Elements for Non-linear Heat Conduction Problems, *Communications in Applied Numerical Methods*, Vol. 4, pp. 617-622, 1988.
29. Haji-Sheik, A. and Sparrow, E. M.  
The Solution of Heat Conduction Problems by Probability Methods, *Journal of Heat Transfer, ASME*, Vol. 89, pp. 121-131, 1967.
30. Partridge, P. W. and Wrobel, L. C.  
The Dual Reciprocity Boundary Element Method for Spontaneous Ignition, *International Journal for Numerical Methods in Engineering*, Vol. 30, pp. 953-963, 1990.
31. Zinn, J. and Mader, C. L.  
Thermal Initiation of Explosives, *Journal of Applied Physics*, Vol. 31, pp. 323-328, 1960.
32. Boddington, T., Gray, P. and Harvey, D. I.  
Thermal Theory of Spontaneous Ignition: Criticality in Bodies of Arbitrary Shape, *Phil. Trans. Royal Soc. London*, Vol. 270, pp. 467-506, 1971.

# Chapter 6

## Other Fundamental Solutions

### 6.1 Introduction

Up to now, all applications of the Dual Reciprocity Method considered in this book were related to the Laplace operator. This means that the governing partial differential equations were initially recast as some kind of Poisson's equation and the fundamental solution of Laplace's equation employed to obtain an equivalent integral formulation. The resulting domain integral was then approximated in a DRM fashion, and a boundary integral equation obtained and solved without resorting to internal discretization or integration.

Obviously, the DRM is not restricted to Laplacian operators. The method has previously been used for different problems and, in fact, the earlier applications of the DRM dealt with elastodynamics using Kelvin's fundamental solution of elastostatics.

This chapter discusses the application of the DRM to some problems for which a fundamental solution other than that of Laplace's equation is employed. The chapter opens considering elasticity problems. A brief review of the indicial notation generally utilized to formulate the mathematical model is included for those readers not familiar with it. Initially, the basic concepts of the BEM as applied to two-dimensional elasticity problems are described for completeness. It is discussed how the DRM can be applied to transform domain integrals resulting from body forces into equivalent boundary integrals for elastostatics problems. It is also explained how to treat the inertial effects in elastodynamics, for both harmonic and transient cases, and numerical results provided for these problems.

Next, plate bending problems are considered. The formulation is discussed within the framework of Kirchhoff's thin plate theory, and the corresponding fundamental solution is employed. Two cases where the DRM can be efficiently employed are included: plates on elastic foundations and non-linear bending using Berger's equation. In both situations, extra domain terms that appear in the formulation are dealt with using DRM approximations, leading to boundary-only integral equations.

Ending the section on elasticity, three-dimensional problems with body forces are considered with application to gravitational, centrifugal and thermal loadings.

Finally, transient convection-diffusion is dealt with. This problem has already been considered in chapter 5, using the fundamental solution of Laplace's equation. For the case of constant velocity fields, a fundamental solution of the steady-state convection-diffusion equation is available. Herein, this fundamental solution is employed to solve transient problems, in which case a Dual Reciprocity approximation is necessary only for the time-derivative term.

The concepts described in this chapter are rather general and permit the extension of the Dual Reciprocity Method to treat not only the problems mentioned above but others involving any type of fundamental solution.

## 6.2 Two-Dimensional Elasticity

In this section, some basic concepts of the linear theory of elasticity are reviewed in accordance with standard texts on the subject [1-3]. For convenience of presentation, the Cartesian tensor notation is adopted. Such notation makes use of subscript indices (1,2) to represent  $x$  and  $y$  and also renders summation symbols unnecessary when the same letter subscript appears twice in an equation. Hence, in two dimensions,

$$a_i a_i = a_1^2 + a_2^2$$

and

$$a_{kk} = a_{11} + a_{22}$$

Also, the Kronecker delta  $\delta_{ij}$

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

will be consistently used.

The static equilibrium of forces and moments in a body requires satisfaction of the equation

$$\sigma_{ij,j} + b_i = 0 \quad (6.1)$$

where  $\sigma_{ij}$  are the components of the stress tensor and  $b_i$  those of the body forces. The comma in the above equation indicates a space derivative, i.e.  $\sigma_{ij,j} = \partial \sigma_{ij} / \partial x_j$ . Furthermore, if no body moments are applied, the stress tensor is symmetric, i.e.

$$\sigma_{ij} = \sigma_{ji} \quad (6.2)$$

The relation between the stress tensor and surface forces (tractions) is given by

$$p_i = \sigma_{ij} n_j \quad (6.3)$$

where  $n_j$  represents the direction cosines of the outward normal to the surface.

Under the action of forces, a body is displaced from its original configuration. If the displacements  $u_i$  are small enough that the square and product of its partial derivatives are negligible, then strains can be represented by the Cauchy infinitesimal strain tensor:

$$\epsilon_{ij} = \frac{1}{2} (u_{i,j} + u_{j,i}) \quad (6.4)$$

For an isotropic elastic material undergoing no change in temperature, Hooke's law relating stresses and strains can be stated in the form

$$\sigma_{ij} = 2G\epsilon_{ij} + \frac{2G\nu}{1-2\nu}\epsilon_{kk}\delta_{ij} \quad (6.5)$$

or, inversely,

$$\epsilon_{ij} = \frac{1}{2G} \left( \sigma_{ij} - \frac{\nu}{1+\nu} \sigma_{kk} \delta_{ij} \right) \quad (6.6)$$

where  $\nu$  is Poisson's ratio and  $G$  the shear modulus, which can be related to the Young's modulus as follows:

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

Equations (6.1), (6.4) and (6.5) represent a set of 8 equations for 3 stresses, 3 strains and 2 displacements which can be further manipulated. A straightforward procedure is to substitute (6.4) into (6.5) to obtain stresses in terms of displacement gradients, and then substitute the result into (6.1) to obtain two second-order partial differential equations for the displacement components. The result of these operations is the well-known Navier equation which may be written in the form

$$Gu_{j,kk} + \frac{G}{1-2\nu}u_{k,kj} + b_j = 0 \quad (6.7)$$

This equation is particularly convenient when displacement boundary conditions are specified. By using (6.4) and (6.5) as before, but now substituting into (6.3) for boundary points, the traction boundary conditions are obtained, i.e.

$$\frac{2G\nu}{1-2\nu}u_{k,k}n_i + G(u_{i,j} + u_{j,i})n_j = p_i \quad (6.8)$$

It is interesting to note that since the equilibrium condition is now expressed in terms of displacements in equation (6.7), the compatibility equations are no longer required. The displacements  $u_i$  are solved from the Navier equations to satisfy the boundary conditions. After  $u_i$  are known throughout the body, the strains are obtained by equation (6.4), and the stresses are calculated by using Hooke's law.

### 6.2.1 Static Analysis

In what follows, the equilibrium equation (6.1) will be solved subject to the boundary conditions

$$u_i = \bar{u}_i \quad \text{on } \Gamma_1 \quad (6.9)$$

$$p_i = \sigma_{ij} n_j = \bar{p}_i \quad \text{on } \Gamma_2 \quad (6.10)$$

The starting weighted residual statement can be written as

$$\int_{\Omega} (\sigma_{kj,j} + b_k) u_k^* d\Omega = \int_{\Gamma_2} (p_k - \bar{p}_k) u_k^* d\Gamma - \int_{\Gamma_1} (u_k - \bar{u}_k) p_k^* d\Gamma \quad (6.11)$$

where  $u_k^*$  and  $p_k^*$  are the displacements and tractions corresponding to the weighting field, i.e.

$$p_k^* = \sigma_{kj}^* n_j \quad (6.12)$$

The strain-displacement relationship (6.4) and the constitutive equation (6.5) are assumed to apply for both the approximating and the weighting fields.

The first term in equation (6.11) can be integrated by parts, giving

$$-\int_{\Omega} \sigma_{kj} \epsilon_{kj}^* d\Omega + \int_{\Omega} b_k u_k^* d\Omega = -\int_{\Gamma_2} \bar{p}_k u_k^* d\Gamma - \int_{\Gamma_1} p_k u_k^* d\Gamma - \int_{\Gamma_1} (u_k - \bar{u}_k) p_k^* d\Gamma \quad (6.13)$$

Integrating by parts again the first term in the above equation, taking into consideration the reciprocity principle, i.e.

$$\int_{\Omega} \sigma_{kj} \epsilon_{kj}^* d\Omega = \int_{\Omega} \epsilon_{kj} \sigma_{kj}^* d\Omega \quad (6.14)$$

one obtains

$$\int_{\Omega} \sigma_{kj,j}^* u_k d\Omega + \int_{\Omega} b_k u_k^* d\Omega = -\int_{\Gamma_2} \bar{p}_k u_k^* d\Gamma - \int_{\Gamma_1} p_k u_k^* d\Gamma + \int_{\Gamma_1} \bar{u}_k p_k^* d\Gamma + \int_{\Gamma_2} u_k p_k^* d\Gamma \quad (6.15)$$

It is assumed that the stress field  $\sigma_{kj}^*$  satisfies the equilibrium equation

$$\sigma_{kj,j}^* + b_k^* = 0 \quad (6.16)$$

Equation (6.15) can be further modified by assuming that the body force components  $b_k^*$  correspond to positive unit point loads in each of the three orthogonal directions given by the unit vectors  $e_k$ , applied at a fixed point  $i$ . This can be represented as

$$b_k^* = \Delta^i e_k \quad (6.17)$$

where  $\Delta^i$  represents the Dirac delta function at  $i$ .

Therefore, the first integral in equation (6.15) will be given by

$$\int_{\Omega} \sigma_{kj,j}^* u_k d\Omega = - \int_{\Omega} b_k^* u_k d\Omega = - \int_{\Omega} \Delta^i u_k e_k d\Omega = - u_k^i e_k \quad (6.18)$$

where  $u_k^i$  represents the component  $k$  of the displacement at the point  $i$  of the application of the load. Furthermore, if each point load is taken as independent, the starred displacements and tractions can be written in the form

$$u_k^* = u_{lk}^* e_l \quad (6.19)$$

$$p_k^* = p_{lk}^* e_l \quad (6.20)$$

where  $u_{lk}^*$  and  $p_{lk}^*$  represent the displacements and tractions in direction  $k$  at a field point corresponding to a unit point force acting in direction  $l$ , applied at a load point.

From the above, it is seen that equation (6.15) can be rewritten to represent the three separate displacement components at the load point  $i$ , in the form

$$u_l^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma + \int_{\Omega} u_{lk}^* b_k d\Omega \quad (6.21)$$

Equation (6.21) is known as Somigliana's identity for displacements [4] and was here obtained by reciprocity with a singular solution of the Navier equation (6.7) satisfying

$$G u_{l,kk}^* + \frac{G}{1-2\nu} u_{k,kk}^* + \Delta^i e_l = 0 \quad (6.22)$$

which is the so-called *fundamental solution*.

The fundamental solution for the plane strain problem in an infinite elastic medium, known as Kelvin's solution, is of the form [5]

$$u_{ij}^* = \frac{1}{8\pi(1-\nu)G} \left[ (3-4\nu) \delta_{ij} \ln \left( \frac{1}{r} \right) + r_{,i} r_{,j} \right] \quad (6.23)$$

with the corresponding traction field

$$p_{ij}^* = \frac{-1}{4\pi(1-\nu)r} \left[ [(1-2\nu)\delta_{ij} + 2r_{,i}r_{,j}] \frac{\partial r}{\partial n} - (1-2\nu)(r_{,i}n_j - r_{,j}n_i) \right] \quad (6.24)$$

in which  $r_{,i} = \partial r / \partial x_i$ . These expressions are also valid for plane stress if  $\nu$  is replaced by  $\bar{\nu} = \nu/(1+\nu)$ .

In order to obtain a boundary integral equation, the source point in equation (6.21) is taken to the boundary and a limit analysis similar to that of chapter 2 carried out. The result can be written as follows

$$c_{lk}^i u_k^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma + \int_{\Omega} u_{lk}^* b_k d\Omega \quad (6.25)$$

As for potential problems, if the boundary  $\Gamma$  is smooth at the source point, one obtains  $c_{lk}^i = \delta_{lk}/2$ ; otherwise, closed form expressions for two- and three-dimensional cases have been presented in [6].

The numerical solution of the above equation follows the same steps as for potential problems, described in chapter 2. For the moment, it is assumed that body forces are absent in the analysis. Initially, the boundary  $\Gamma$  is discretized into a series of elements; then, the variation of  $u_k$  and  $p_k$  within each element is approximated using interpolation functions and nodal values; finally, the discretized equation is satisfied at a finite number of points along the boundary (the collocation points or boundary nodes). The final result is a system of algebraic equations which, after imposition of the boundary conditions, can be solved to provide the remaining boundary unknowns.

In order to show the above steps in more detail, it is now more convenient to work with matrices rather than the indicial notation. To this effect, equation (6.25) is rewritten in the form below in which body forces have been omitted for simplicity,

$$\mathbf{c}^i \mathbf{u}^i = \int_{\Gamma} \mathbf{u}^* \mathbf{p} d\Gamma - \int_{\Gamma} \mathbf{p}^* \mathbf{u} d\Gamma \quad (6.26)$$

where the displacements and tractions vectors have the components

$$\mathbf{u} = \begin{Bmatrix} u_1 \\ u_2 \end{Bmatrix}; \quad \mathbf{p} = \begin{Bmatrix} p_1 \\ p_2 \end{Bmatrix} \quad (6.27)$$

and the following two matrices of fundamental solutions have also been defined:

$$\mathbf{u}^* = \begin{bmatrix} u_{11}^* & u_{12}^* \\ u_{21}^* & u_{22}^* \end{bmatrix}; \quad \mathbf{p}^* = \begin{bmatrix} p_{11}^* & p_{12}^* \\ p_{21}^* & p_{22}^* \end{bmatrix} \quad (6.28)$$

It is important to notice that the free term  $\mathbf{c}^i$  is now also a  $2 \times 2$  matrix

$$\mathbf{c} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \quad (6.29)$$

After equation (6.26) has been discretized into a number of boundary elements, the values of  $\mathbf{u}$  and  $\mathbf{p}$  within each element can be approximated using interpolation functions, *i.e.*

$$\mathbf{u} = \boldsymbol{\phi}^T \mathbf{u}^n \quad (6.30)$$

$$\mathbf{p} = \boldsymbol{\phi}^T \mathbf{p}^n \quad (6.31)$$

in which  $\mathbf{u}^n$  and  $\mathbf{p}^n$  refer to the nodal displacements and tractions, respectively, and  $\boldsymbol{\phi}$  is the vector of interpolation functions, *i.e.*

$$\boldsymbol{\phi} = \begin{Bmatrix} \phi_1 \\ \phi_2 \end{Bmatrix} \quad (6.32)$$

Substituting the above approximations into the discretized version of equation (6.26), the following equation is obtained

$$\mathbf{c}^i \mathbf{u}^i = \sum_{j=1}^N \left( \int_{\Gamma_j} \mathbf{u}^* \boldsymbol{\phi}^T d\Gamma \right) \mathbf{p}^n - \sum_{j=1}^N \left( \int_{\Gamma_j} \mathbf{p}^* \boldsymbol{\phi}^T d\Gamma \right) \mathbf{u}^n \quad (6.33)$$

where the summation from 1 to  $N$  indicates summation over the  $N$  elements on the boundary and  $\Gamma_j$  is the surface of element  $j$ . Calling, as in chapter 2,

$$\mathbf{h}_{ij}^1 = \int_{\Gamma_j} \phi_1 \mathbf{p}^* d\Gamma \quad (6.34)$$

$$\mathbf{h}_{ij}^2 = \int_{\Gamma_j} \phi_2 \mathbf{p}^* d\Gamma \quad (6.35)$$

$$\mathbf{g}_{ij}^1 = \int_{\Gamma_j} \phi_1 \mathbf{u}^* d\Gamma \quad (6.36)$$

and

$$\mathbf{g}_{ij}^2 = \int_{\Gamma_j} \phi_2 \mathbf{u}^* d\Gamma \quad (6.37)$$

in which  $\mathbf{h}_{ij}^k$  and  $\mathbf{g}_{ij}^k$  are now  $2 \times 2$  matrices, and substituting into equation (6.33), one obtains the following equation for node  $i$

$$\begin{aligned} \mathbf{c}^i \mathbf{u}^i + \left[ \begin{array}{cccc} \bar{\mathbf{H}}_{i1} & \bar{\mathbf{H}}_{i2} & \cdots & \bar{\mathbf{H}}_{iN} \end{array} \right] \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_N \end{bmatrix} = \\ \left[ \begin{array}{cccc} \mathbf{G}_{i1} & \mathbf{G}_{i2} & \cdots & \mathbf{G}_{iN} \end{array} \right] \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_N \end{bmatrix} \end{aligned} \quad (6.38)$$

where  $\bar{\mathbf{H}}_{ij}$  is the summation of the submatrix  $\mathbf{h}_{ij}^1$  of element  $j$  with the submatrix  $\mathbf{h}_{i,j-1}^2$  of element  $j - 1$ , the same applying for  $\mathbf{G}_{ij}$ . Hence, formula (6.38) represents the assembled equation for node  $i$ . The above matrix equation can be written in the more concise form

$$\mathbf{c}^i \mathbf{u}^i + \sum_{j=1}^N \bar{\mathbf{H}}_{ij} \mathbf{u}_j = \sum_{j=1}^N \mathbf{G}_{ij} \mathbf{p}_j \quad (6.39)$$

Incorporating the term  $\mathbf{c}^i$  into matrix  $\mathbf{H}$ , equation (6.39) becomes

$$\sum_{j=1}^N \mathbf{H}_{ij} \mathbf{u}_j = \sum_{j=1}^N \mathbf{G}_{ij} \mathbf{p}_j \quad (6.40)$$

The final step is to apply equation (6.40) to each boundary node successively, generating a system of  $2N$  equations which can be expressed as

$$\mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{p} \quad (6.41)$$

After incorporating the boundary conditions of the problem, this system can be solved by Gauss elimination.

### 6.2.2 Treatment of Body Forces

The Dual Reciprocity Method can be used in elastostatics to transform the domain integral arising from the application of body forces into equivalent boundary integrals. The formulation closely follows that for Poisson's equation, described in Chapter 3.

Assuming that body forces are now present in the definition of the problem, the boundary integral equation to be solved takes the form of equation (6.25). The following approximation can then be proposed for the term  $b_k (k = 1, 2)$ :

$$b_k \simeq \sum_{j=1}^{N+L} f^j \alpha_k^j \quad (6.42)$$

where, as before, the  $\alpha_k^j$  are a set of initially unknown coefficients and the  $f^j$  are approximating functions. If particular solutions  $\hat{u}_{mk}^j$  can be found satisfying equation (6.7), *i.e.*

$$G\hat{u}_{mk, ll}^j + \frac{G}{1 - 2\nu} \hat{u}_{lk, lm}^j = \delta_{mk} f^j \quad (6.43)$$

then, upon substitution of (6.42) and (6.43) into equation (6.25), one obtains from the application of the reciprocity principle to the resulting domain integral

$$c_{lk}^i u_k^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma + \sum_{j=1}^{N+L} \left( c_{lk}^i \hat{u}_{mk}^{ij} + \int_{\Gamma} p_{lk}^* \hat{u}_{mk}^j d\Gamma - \int_{\Gamma} u_{lk}^* \hat{p}_{mk}^j d\Gamma \right) \alpha_m^j \quad (6.44)$$

where  $\hat{p}_{mk}^j$  are the tractions corresponding to the particular solutions  $\hat{u}_{mk}^j$ . It can be noted that equation (6.44) is equivalent to (3.10) for Poisson's equation.

The procedure for numerical solution of the above equation follows that described in section 3.1.2. Equation (6.44) can be written using matricial rather than indicial notation as

$$\mathbf{c}^i \mathbf{u}^i = \int_{\Gamma} \mathbf{u}^* \mathbf{p} d\Gamma - \int_{\Gamma} \mathbf{p}^* \mathbf{u} d\Gamma + \sum_{j=1}^{N+L} \left( \mathbf{c}^i \hat{\mathbf{u}}^{ij} + \int_{\Gamma} \mathbf{p}^* \hat{\mathbf{u}}^j d\Gamma - \int_{\Gamma} \mathbf{u}^* \hat{\mathbf{p}}^j d\Gamma \right) \boldsymbol{\alpha}^j \quad (6.45)$$

After discretization and approximation of the variation of  $\mathbf{u}, \mathbf{p}, \hat{\mathbf{u}}, \hat{\mathbf{p}}$  over each element using their nodal values and the same set of interpolation functions, equation (6.45) becomes

$$\mathbf{c}^i \mathbf{u}^i + \sum_{k=1}^N \overline{\mathbf{H}}_{ik} \mathbf{u}_k - \sum_{k=1}^N \mathbf{G}_{ik} \mathbf{p}_k = \sum_{j=1}^{N+L} \left( \mathbf{c}^i \hat{\mathbf{u}}^{ij} + \sum_{k=1}^N \overline{\mathbf{H}}_{ik} \hat{\mathbf{u}}_k^j - \sum_{k=1}^N \mathbf{G}_{ik} \hat{\mathbf{p}}_k^j \right) \boldsymbol{\alpha}^j \quad (6.46)$$

Finally, applying the above equation to all boundary nodes, the following system of equations is obtained

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{p} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{P}})\boldsymbol{\alpha} \quad (6.47)$$

or, substituting  $\alpha = \mathbf{F}^{-1}\mathbf{b}$ ,

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{p} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{P}})\mathbf{F}^{-1}\mathbf{b} \quad (6.48)$$

If the expression for the approximating function  $f$  is of the same form as used in the previous chapters, i.e.

$$f = 1 + r \quad (6.49)$$

the particular solution  $\hat{u}$  satisfying equation (6.43) is given by

$$\begin{aligned} \hat{u}_{mk} &= \frac{1 - 2\nu}{(5 - 4\nu)G} r_{,m} r_{,k} r^2 + \\ &\quad \frac{1}{30(1 - \nu)G} \left[ \left( 3 - \frac{10\nu}{3} \right) \delta_{mk} - r_{,m} r_{,k} \right] r^3 \end{aligned} \quad (6.50)$$

The corresponding expression for the traction  $\hat{p}$  is

$$\begin{aligned} \hat{p}_{mk} &= \frac{2(1 - 2\nu)}{5 - 4\nu} \left[ \frac{1 + \nu}{1 - 2\nu} r_{,m} n_k + \frac{1}{2} r_{,k} n_m + \frac{1}{2} \delta_{mk} \frac{\partial r}{\partial n} \right] r + \\ &\quad \frac{1}{15(1 - \nu)} \left[ (4 - 5\nu) r_{,k} n_m - (1 - 5\nu) r_{,m} n_k + [(4 - 5\nu) \delta_{mk} - r_{,m} r_{,k}] \frac{\partial r}{\partial n} \right] r^2 \end{aligned} \quad (6.51)$$

The above expressions can be checked by the reader. To verify the first, one initially differentiates (6.50) twice and substitutes the result into expression (6.43) for  $f$  defined in (6.49). For the second, the following expressions should be used:

$$\hat{\epsilon}_{mkl} = \frac{1}{2}(\hat{u}_{mk,l} + \hat{u}_{ml,k}) \quad (6.52)$$

$$\hat{\sigma}_{mkl} = 2G\hat{\epsilon}_{mkl} + \frac{2G\nu}{1 - 2\nu} \hat{\epsilon}_{mj,j} \delta_{kl} \quad (6.53)$$

for the strain and stress components  $kl$  at any point, due to a unit point load in direction  $m$  applied at a fixed point. The above tensors are of the form

$$\begin{aligned} \hat{\epsilon}_{mkl} &= \frac{1 - 2\nu}{(5 - 4\nu)G} \left[ \delta_{kl} r_{,m} + \frac{1}{2}(\delta_{mk} r_{,l} + \delta_{ml} r_{,k}) \right] r + \\ &\quad \frac{1}{30(1 - \nu)G} [(4 - 5\nu)(\delta_{mk} r_{,l} + \delta_{ml} r_{,k}) - \delta_{kl} r_{,m} - r_{,m} r_{,k} r_{,l}] r^2 \end{aligned} \quad (6.54)$$

$$\begin{aligned} \hat{\sigma}_{mkl} &= \frac{2(1 - 2\nu)}{5 - 4\nu} \left[ \frac{1 + \nu}{1 - 2\nu} \delta_{kl} r_{,m} + \frac{1}{2}(\delta_{mk} r_{,l} + \delta_{ml} r_{,k}) \right] r + \\ &\quad \frac{1}{15(1 - \nu)} [(4 - 5\nu)(\delta_{mk} r_{,l} + \delta_{ml} r_{,k}) - (1 - 5\nu)\delta_{kl} r_{,m} - r_{,m} r_{,k} r_{,l}] r^2 \end{aligned} \quad (6.55)$$

Expression (6.51) for  $\hat{p}_{mk}$  is obtained by writing

$$\hat{p}_{mk} = \hat{\sigma}_{mkl} n_l \quad (6.56)$$

The most common types of body forces that appear in elasticity problems are those due to self-weight (gravitational), centrifugal and thermal loadings. Although they can be dealt with by using the Galerkin-tensor formulation described in chapter 2 (see [5] for details), this technique is restricted to these cases only. The DRM formulation presented here can equally be applied to these situations but, due to its approximating character, presents no restriction whatsoever regarding more general types of body forces.

Some DRM results for elasticity problems with body forces are presented for the three-dimensional case in section 6.4.

### 6.2.3 Dynamic Analysis

The conditions of dynamical equilibrium of a body are expressed by the equation

$$\sigma_{kj,j} + b_k = \rho \ddot{u}_k \quad (6.57)$$

where  $\rho$  is the mass density and the dots denote differentiation with respect to time. This equation can also be expressed in terms of the displacement field as

$$G u_{k,jj} + \frac{G}{1 - 2\nu} u_{j,jk} + b_k = \rho \ddot{u}_k \quad (6.58)$$

The above form is compatible with the Navier equation of elasticity (6.7).

In order to formulate uniquely the dynamic problem, one has to impose boundary conditions of the types (6.9) and (6.10) and initial conditions which specify the state of displacements and velocities at time  $t_0$ , i.e.

$$u_k = u_k^0 \quad (6.59)$$

$$\dot{u}_k = \dot{u}_k^0 \quad (6.60)$$

The application of the Dual Reciprocity Method to the dynamic problem initially requires a boundary integral equation which is obtained by using the static fundamental solutions (6.23) and (6.24) for displacements and tractions. This integral equation is similar in form to (6.25) and, in the absence of body forces, it reads

$$c_{lk}^i u_k^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma - \rho \int_{\Omega} u_{lk}^* \ddot{u}_k d\Omega \quad (6.61)$$

The next step in the formulation is the introduction of the approximation for the accelerations  $\ddot{u}_k$ , similarly to what was done for the scalar wave equation in section 5.5. Writing (see expression 5.7),

$$\ddot{u}_k(x, y, t) \simeq \sum_{j=1}^{N+L} f^j(x, y) \alpha_k^j(t) \quad (6.62)$$

and using the particular solution  $\hat{u}^j$  related to  $f^j$  through expression (6.43), as in the case of body forces, an equation similar in form to (6.44) is obtained, noting that the parameters  $\alpha^j$  are now time-dependent. The discretization and application of this boundary integral equation to all boundary nodes generates the system

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{p} = -\rho(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{P}})\boldsymbol{\alpha} \quad (6.63)$$

or, substituting  $\boldsymbol{\alpha} = \mathbf{F}^{-1}\ddot{\mathbf{u}}$ ,

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{p} = -\rho(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{P}})\mathbf{F}^{-1}\ddot{\mathbf{u}} \quad (6.64)$$

This equation can be rewritten in the form

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{H}\mathbf{u} = \mathbf{G}\mathbf{p} \quad (6.65)$$

where the generalized mass matrix  $\mathbf{M}$  is given by

$$\mathbf{M} = \rho(\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{P}})\mathbf{F}^{-1} \quad (6.66)$$

The previous DRM formulation for dynamic analysis was originally derived by Brebbia and Nardini [7],[8] and applied to a series of harmonic and transient problems. In what follows, some of their numerical results are reproduced and discussed.

The system (6.65) was initially partitioned according to the type of applied boundary condition, and then statically condensed in such a way that the final system could be solved for the unknown displacements only. This procedure is similar to that explained for the Helmholtz equation in chapter 4, and is possible in this case due to the absence of time derivatives of tractions in the present formulation. Denoting the variables in parts  $\Gamma_1$  and  $\Gamma_2$  of the boundary by the subscripts 1 and 2, i.e.

$$\mathbf{u} = \begin{Bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{Bmatrix}; \quad \mathbf{p} = \begin{Bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \end{Bmatrix} \quad (6.67)$$

the global system (6.65) can be partitioned as follows,

$$\mathbf{M}_{11}\ddot{\mathbf{u}}_1 + \mathbf{M}_{12}\ddot{\mathbf{u}}_2 + \mathbf{H}_{11}\mathbf{u}_1 + \mathbf{H}_{12}\mathbf{u}_2 = \mathbf{G}_{11}\mathbf{p}_1 + \mathbf{G}_{12}\mathbf{p}_2 \quad (6.68)$$

$$\mathbf{M}_{21}\ddot{\mathbf{u}}_1 + \mathbf{M}_{22}\ddot{\mathbf{u}}_2 + \mathbf{H}_{21}\mathbf{u}_1 + \mathbf{H}_{22}\mathbf{u}_2 = \mathbf{G}_{21}\mathbf{p}_1 + \mathbf{G}_{22}\mathbf{p}_2 \quad (6.69)$$

Inverting  $\mathbf{G}_{11}$  from the first equation above, the tractions  $\mathbf{p}_1$  can be expressed in terms of the other variables in the form

$$\mathbf{p}_1 = \mathbf{G}_{11}^{-1}(\mathbf{M}_{11}\ddot{\mathbf{u}}_1 + \mathbf{M}_{12}\ddot{\mathbf{u}}_2 + \mathbf{H}_{11}\mathbf{u}_1 + \mathbf{H}_{12}\mathbf{u}_2 - \mathbf{G}_{12}\mathbf{p}_2) \quad (6.70)$$

Substituting this expression for  $\mathbf{p}_1$  into equation (6.69), one finally obtains

$$\hat{\mathbf{M}}\ddot{\mathbf{u}}_2 + \hat{\mathbf{H}}\mathbf{u}_2 = \check{\mathbf{M}}\ddot{\mathbf{u}}_1 + \check{\mathbf{H}}\mathbf{u}_1 + \hat{\mathbf{G}}\mathbf{p}_2 \quad (6.71)$$

with the modified matrices defined as follows:

$$\hat{\mathbf{M}} = \mathbf{M}_{22} - \mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{M}_{12}$$

$$\hat{\mathbf{H}} = \mathbf{H}_{22} - \mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{H}_{12}$$

$$\hat{\mathbf{G}} = \mathbf{G}_{22} - \mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{G}_{12}$$

$$\check{\mathbf{M}} = \mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{M}_{11} - \mathbf{M}_{21}$$

$$\check{\mathbf{H}} = \mathbf{G}_{21}\mathbf{G}_{11}^{-1}\mathbf{H}_{11} - \mathbf{H}_{21}$$

The right-hand side of equation (6.71) contains only the externally applied displacements  $\mathbf{u}_1$  and tractions  $\mathbf{p}_2$ , so the system can be solved for  $\mathbf{u}_2$  using a direct time integration procedure.

The most common types of dynamical problems can be summarized as follows:

### Free Vibrations

The analysis of natural modes and frequencies can be deduced from the general transient system (6.71) by setting the external loads to zero, obtaining

$$\hat{\mathbf{M}}\ddot{\mathbf{u}}_2 + \hat{\mathbf{H}}\mathbf{u}_2 = 0 \quad (6.72)$$

Assuming that the displacements are harmonic functions of time, they can be expanded in the form

$$\ddot{\mathbf{u}}_2 = -\omega^2\mathbf{u}_2 \quad (6.73)$$

with  $\omega$  being the natural circular frequency. Equation (6.71) is then reduced to

$$\hat{\mathbf{H}}\mathbf{u}_2 = \omega^2\hat{\mathbf{M}}\mathbf{u}_2 \quad (6.74)$$

which represents the generalized algebraic eigenvalue problem. It should be pointed out that neither  $\hat{\mathbf{H}}$  nor  $\hat{\mathbf{M}}$  are symmetric or positive definite, so that care should be taken in the choice of the appropriate eigenvalue solution algorithm.

The method employed by Nardini and Brebbia [8] reduced the generalized eigenvalue problem to a standard one by the inversion of matrix  $\hat{\mathbf{H}}$ , obtaining the system

$$\mathbf{A}\mathbf{u}_2 = \lambda\mathbf{u}_2 \quad (6.75)$$

with

$$\mathbf{A} = \hat{\mathbf{H}}^{-1}\hat{\mathbf{M}}$$

$$\lambda = \frac{1}{\omega^2}$$

Matrix  $\mathbf{A}$  is then transformed into tridiagonal form by the Householder algorithm, and the eigenvalues and eigenvectors of the transformed matrix are calculated by the Q-R algorithm [9]. For very large systems, Nardini and Brebbia [10] recommended the use of a variant of the subspace iteration method adapted by Dong [11] for non-symmetric matrices.

## Forced Vibrations

It is seldom the case in practical applications to have simultaneously both external tractions and support movements applied to the system. Therefore, the two cases can be considered separately, although the two effects could be taken into account simultaneously if required.

In the absence of support excitations, equation (6.71) is simplified in the form

$$\hat{\mathbf{M}}\ddot{\mathbf{u}}_2 + \hat{\mathbf{H}}\mathbf{u}_2 = \hat{\mathbf{G}}\mathbf{p}_2 \quad (6.76)$$

where  $\mathbf{p}_2$  is a vector of externally applied, time-dependent tractions.

When support excitations are given as a general function of time, equation (6.71) reduces to

$$\hat{\mathbf{M}}\ddot{\mathbf{u}}_2 + \hat{\mathbf{H}}\mathbf{u}_2 = \hat{\mathbf{M}}\ddot{\mathbf{u}}_1 + \hat{\mathbf{H}}\mathbf{u}_1 \quad (6.77)$$

This expression employs the enforced displacements  $\mathbf{u}_1$  and accelerations  $\ddot{\mathbf{u}}_1$  at part  $\Gamma_1$  of the boundary, and is valid for any compatible variation of  $\mathbf{u}_1$  with time.

The solution of equation (6.76) or (6.77) may be obtained by a direct time integration procedure. As mentioned in section 5.5, the Houbolt scheme [12] has shown to provide good resolution for DRM formulations due to its capability of introducing numerical damping for the high frequencies of the problem.

Alternatively, Nardini and Brebbia [10] decoupled the system response into independent modes and solved the transient problem using modal superposition. Since matrix  $\mathbf{A}$  in (6.75) is non-symmetric, one has to work with two bases, one corresponding to the original eigenvalue problem (6.75) and the other corresponding to its adjoint problem, *i.e.*

$$\mathbf{A}^T \mathbf{v}_2 = \lambda \mathbf{v}_2 \quad (6.78)$$

with  $\mathbf{A}^T$  being the transpose of  $\mathbf{A}$  and  $\mathbf{v}_2$  the alternative basis. The set of eigenvalues  $\lambda$  is the same in both cases.

Using the notation  $\mathbf{A} = \hat{\mathbf{H}}^{-1}\hat{\mathbf{M}}$ ,  $\mathbf{x} = \mathbf{u}_2$  and  $\mathbf{b}(t) = \hat{\mathbf{H}}^{-1}\mathbf{f}(t)$ , with  $\mathbf{f}(t)$  representing the right-hand side of equation (6.76) or (6.77) ( $\mathbf{f}(t)$  is a known function of time), it is possible to rewrite the transient problem (6.76) or (6.77) in the form

$$\mathbf{A}\ddot{\mathbf{x}} + \mathbf{x} = \mathbf{b}(t) \quad (6.79)$$

As for a symmetrical system,  $\mathbf{x}$  can be represented in the original basis  $\mathbf{u}_2$  as

$$\mathbf{x} = \sum_{j=1}^n X_j \mathbf{u}_{2j} \quad (6.80)$$

where  $X_j(t)$  are the modal contributions,  $\mathbf{u}_{2j}$  are the modal shapes and  $n$  is the number of modes considered. It is noted that  $X_j(t)$  are unknown functions. Substituting the above into equation (6.79), one obtains

$$\mathbf{A} \sum_{j=1}^n \ddot{X}_j \mathbf{u}_{2j} + \sum_{j=1}^n X_j \mathbf{u}_{2j} = \mathbf{b} \quad (6.81)$$

Pre-multiplying this expression by vector  $\mathbf{v}_{2i}^T$  of the dual basis the system of equations can be decoupled, due to the orthogonality of  $\mathbf{v}_2$  and  $\mathbf{u}_2$  with respect to matrix  $\mathbf{A}$  and the identity matrix, into the form

$$(\mathbf{v}_{2i}^T \mathbf{A} \mathbf{u}_{2i}) \ddot{X}_i + (\mathbf{v}_{2i}^T \mathbf{u}_{2i}) X_i = \mathbf{v}_{2i}^T \mathbf{b} \quad (6.82)$$

or

$$\ddot{X}_i + \omega_i^2 X_i = \frac{\mathbf{v}_{2i}^T \mathbf{b}}{\mathbf{v}_{2i}^T \mathbf{A} \mathbf{u}_{2i}} \quad (6.83)$$

which represents  $n$  independent one-degree-of-freedom systems that can be solved by a time-stepping technique or using Duhamel's integral formulation.

The previous formulation for dynamic analysis was successfully applied to a number of problems [7],[8],[10]. Figure 6.1 shows the boundary element and the finite element models employed in the study of the dynamic properties of a shear wall with four openings. The BEM model consists of 29 quadratic elements with 58 nodes, while the FEM one comprises 559 nodes and was used for the purpose of comparison. The results for the period of free vibrations for the first eight natural modes are given in table 6.1. In spite of the complicated geometry and the relatively small number of boundary elements used, the agreement is excellent.

Mode	1	2	3	4	5	6	7	8
BEM	3.02	0.875	0.822	0.531	0.394	0.337	0.310	0.276
FEM	3.03	0.885	0.824	0.526	0.409	0.342	0.316	0.283

Table 6.1: Free Vibration Periods for Shear Wall

Another set of results for free vibration problems is shown in figure 6.3, for the arch depicted in figure 6.2. The geometrical dimensions and physical constants of the arch are  $r_0 = 2.5$ ,  $r_1 = 5.0$ ,  $E/\rho = 10^4$  and  $\nu = 0.25$ . Again, good agreement between the natural frequencies obtained by the BEM and FEM models can be noted, with a small number of nodes in the BEM model.

Figure 6.4 shows the horizontal and vertical displacements history at points  $A$  and  $B$  of a simple rectangular cantilever subject to a uniform shear impulse. The material constants for the problem were taken as  $E = 10^5$ ,  $\rho = 1$  and  $\nu = 0.25$ . The agreement with a FEM solution is excellent although only twelve boundary elements were used in the BEM discretization.

Finally, figure 6.5 shows the time variation of the horizontal displacement at the crest of a dam-like structure subject to a sinusoidal excitation at its base. The forcing frequency was chosen to be 16 Hz, which is roughly an average between the first and

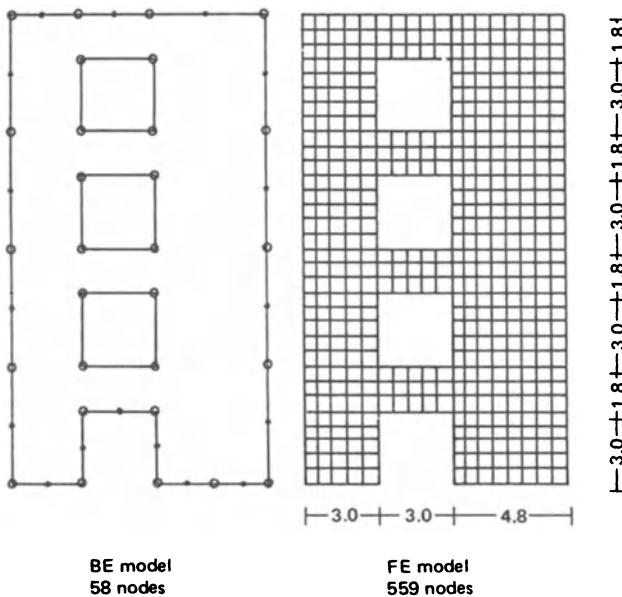


Figure 6.1: BEM and FEM Models for Shear Wall

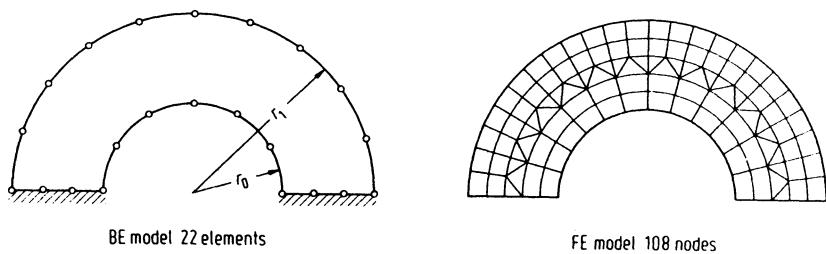


Figure 6.2: BEM and FEM Models for Arch

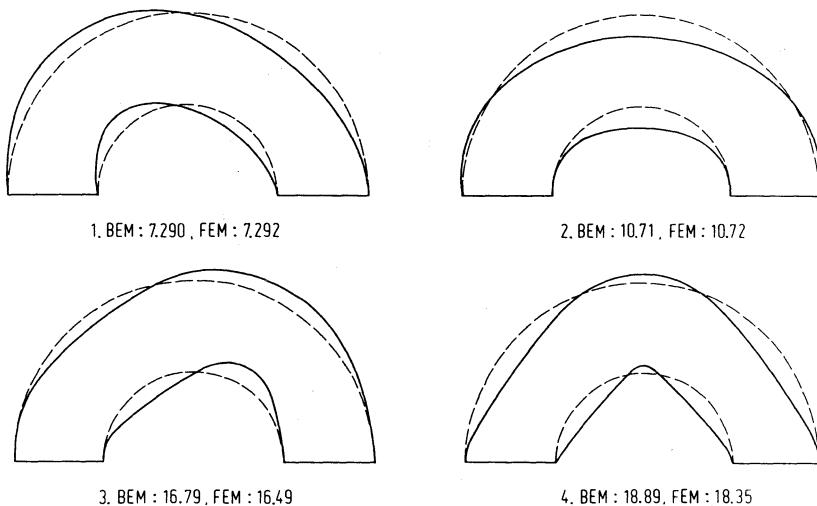


Figure 6.3: Results for Free Vibration Modes and Natural Frequencies (in Hz)

second natural frequencies of the structure, respectively 11.04 and 20.72 Hz. Again, excellent accuracy is obtained with a coarse BEM discretization.

### 6.3 Plate Bending

The dual reciprocity method has also been applied to study the deflection of thin elastic plates resting on elastic foundations supported by Winkler-type springs [13],[14]. The governing differential equation for a plate of constant thickness  $h$  resting on a foundation with stiffness  $k$  and subjected to an arbitrary lateral load  $q(x, y)$  is of the form

$$D\nabla^4 w + kw = q \quad (6.84)$$

where  $w$  is the deflection,  $\nabla^4$  is the bi-harmonic operator and  $D$  the bending rigidity of the plate,  $D = Eh^3/12(1 - \nu^2)$ . It is assumed that the mid-plane of the plate occupies a two-dimensional domain  $\Omega$  bounded by a curve  $\Gamma$ .

Although a fundamental solution to the homogeneous counterpart of equation (6.84) exists [15], its form is rather complicated; thus, some authors prefer to formulate the problem employing the fundamental solution of the bi-harmonic equation, *i.e.* the solution of

$$D\nabla^4 w^* = \Delta^i \quad (6.85)$$

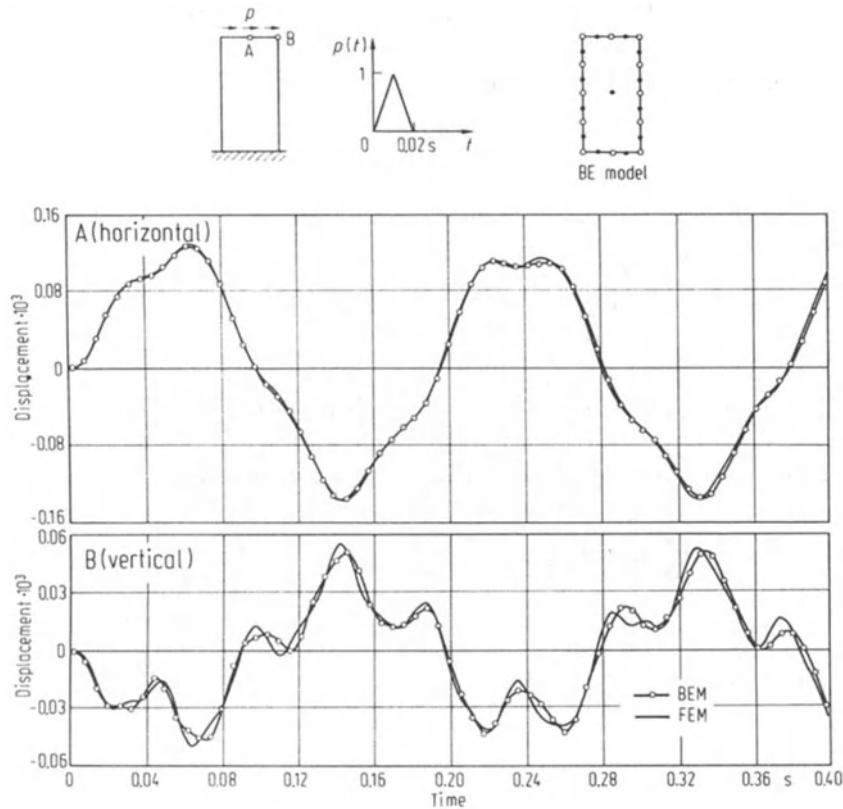


Figure 6.4: Time History of Displacement at Two Points of Rectangular Cantilever

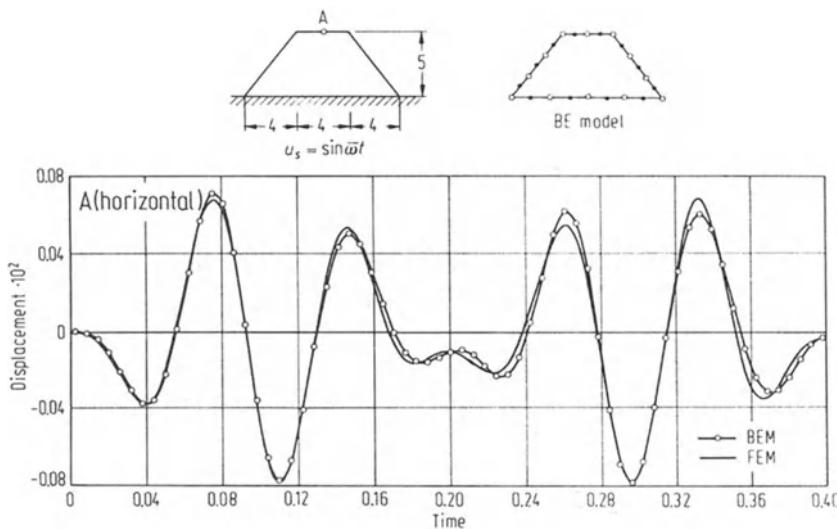


Figure 6.5: Time History of Displacement at Dam Crest

with  $\Delta^i$  representing the Dirac delta function at a point  $i$ , and treating the foundation reaction term  $kw$ , as well as the lateral load  $q$ , by using cell integration or a dual reciprocity approximation. The fundamental solution  $w^*$  of equation (6.85) is

$$w^* = -\frac{r^2}{8\pi D} \ln r \quad (6.86)$$

For a smooth boundary curve  $\Gamma$ , the reciprocity theorem for bi-harmonic operators can be written in the form:

$$\begin{aligned} D \int_{\Omega} (w \nabla^4 w^* - w^* \nabla^4 w) d\Omega = \\ \int_{\Gamma} \left[ w^* K(w) - \frac{\partial w^*}{\partial n} M(w) + \frac{\partial w}{\partial n} M(w^*) - w K(w^*) \right] d\Gamma \end{aligned} \quad (6.87)$$

where the differential operators  $K$  and  $M$  are defined as follows:

$$K = D(1-\nu) \frac{\partial}{\partial s} \left[ n_x n_y \left( \frac{\partial^2}{\partial x^2} - \frac{\partial^2}{\partial y^2} \right) - (n_x^2 - n_y^2) \frac{\partial^2}{\partial x \partial y} \right] - \frac{\partial}{\partial n} \nabla^2$$

$$M = -D \left[ \nu \nabla^2 + (1-\nu) \left( n_x^2 \frac{\partial^2}{\partial x^2} + n_y^2 \frac{\partial^2}{\partial y^2} + 2 n_x n_y \frac{\partial^2}{\partial x \partial y} \right) \right]$$

In the above,  $n_x$  and  $n_y$  are the direction cosines of the outward normal  $n$ ,  $\partial/\partial s$  and  $\partial/\partial n$  denote directional derivatives along the tangent and normal to the boundary  $\Gamma$ , respectively.

Substituting  $D \nabla^4 w$  as given by (6.84) and  $D \nabla^4 w^*$  as given by (6.85) into equation (6.87), the following integral equation is obtained

$$\begin{aligned} c^i w^i = \int_{\Gamma} \left[ w^* K(w) - \frac{\partial w^*}{\partial n} M(w) + \frac{\partial w}{\partial n} M(w^*) - w K(w^*) \right] d\Gamma \\ + \int_{\Omega} q w^* d\Omega - \int_{\Omega} k w w^* d\Omega \end{aligned} \quad (6.88)$$

where  $c^i = 1$  for internal points and  $c^i = 1/2$  for boundary points.

Differentiating equation (6.88) with respect to the outward normal  $n_0$  at source boundary points  $i$ , and replacing  $w$  at field boundary points by  $w - w^i$ , one arrives at the expression

$$\begin{aligned} \frac{1}{2} \frac{\partial w^i}{\partial n_0} = \int_{\Gamma} \left[ \frac{\partial w^*}{\partial n_0} K(w) - \frac{\partial^2 w^*}{\partial n_0 \partial n} M(w) + \frac{\partial w}{\partial n} \frac{\partial M(w^*)}{\partial n_0} - (w - w^i) \frac{\partial K(w^*)}{\partial n_0} \right] d\Gamma \\ + \int_{\Omega} q \frac{\partial w^*}{\partial n_0} d\Omega - \int_{\Omega} k w \frac{\partial w^*}{\partial n_0} d\Omega \end{aligned} \quad (6.89)$$

Equations (6.88) and (6.89) constitute a set of integral equations in terms of the four basic boundary values, i.e. deflection  $w$ , normal slope  $\partial w/\partial n$ , normal bending

moment  $M(w)$  and equivalent shear force  $K(w)$ . The domain integrals include the effects of lateral loads and the foundation reaction. For relatively simple lateral load distributions, it is possible to transform its domain integral into equivalent boundary integrals without resorting to any approximation [16]. In what follows, it is shown that a DRM approximation can be used to find equivalent boundary integrals also for the foundation reaction term, giving rise to a boundary-only formulation of the problem.

Expanding the value of the deflection  $w$  at a general point in the usual DRM form, i.e.

$$w \simeq \sum_{j=1}^{N+L} f^j \alpha^j \quad (6.90)$$

where  $f^j$  represents a set of approximating functions and  $\alpha^j$  is a set of unknown coefficients, and assuming that for each  $f^j$  a corresponding function  $\hat{w}^j$  can be found such that

$$\nabla^4 \hat{w}^j = f^j \quad (6.91)$$

then the second domain integral in equation (6.88) can be written as

$$\int_{\Omega} k w w^* d\Omega = \sum_{j=1}^{N+L} k \alpha^j \int_{\Omega} \nabla^4 \hat{w}^j w^* d\Omega \quad (6.92)$$

With the help of the reciprocity theorem (6.87), the above domain integral may be transformed into the following boundary integral:

$$\begin{aligned} \int_{\Omega} k w w^* d\Omega &= \sum_{j=1}^{N+L} k \alpha^j \left[ c^i \hat{w}^{ij} + \right. \\ &\quad \left. \int_{\Gamma} \left( w^* \frac{\partial}{\partial n} \nabla^2 \hat{w}^j - \frac{\partial w^*}{\partial n} \nabla^2 \hat{w}^j + \frac{\partial \hat{w}^j}{\partial n} \nabla^2 w^* - \hat{w}^j \frac{\partial}{\partial n} \nabla^2 w^* \right) d\Gamma \right] \end{aligned} \quad (6.93)$$

Similarly, the second domain integral in equation (6.89) becomes

$$\begin{aligned} \int_{\Omega} k w \frac{\partial w^*}{\partial n_0} d\Omega &= \sum_{j=1}^{N+L} k \alpha^j \left[ c^i \frac{\partial \hat{w}^{ij}}{\partial n_0} + \right. \\ &\quad \left. \int_{\Gamma} \left( \frac{\partial w^*}{\partial n_0} \frac{\partial}{\partial n} \nabla^2 \hat{w}^j - \frac{\partial^2 w^*}{\partial n_0 \partial n} \nabla^2 \hat{w}^j + \frac{\partial \hat{w}^j}{\partial n} \frac{\partial}{\partial n_0} \nabla^2 w^* - (\hat{w}^j - \hat{w}^{ij}) \frac{\partial^2}{\partial n_0 \partial n} \nabla^2 w^* \right) d\Gamma \right] \end{aligned} \quad (6.94)$$

It should be noted that the terms between brackets in expressions (6.93) and (6.94) involve only known values.

The approximating function  $f^j$  employed by Kamiya and Sawaki [13] is of the form

$$f = 1 - r - r^2 \quad (6.95)$$

with the corresponding  $\hat{w}^j$  set given by

$$\hat{w} = \frac{1}{64} r^4 - \frac{1}{225} r^5 - \frac{1}{576} r^6 \quad (6.96)$$

while Silva and Venturini [14] used the more common form  $f = 1 + r$ .

For the numerical solution of the system of equations obtained through the satisfaction of equations (6.88) and (6.89) at all boundary nodes, the contribution of the foundation reaction terms can be written as a product of known matrices  $\mathbf{S}$  and  $\bar{\mathbf{S}}$  by a vector  $\boldsymbol{\alpha}$  of unknown coefficients. However, applying expression (6.90) to the boundary nodes and inverting, one obtains

$$\boldsymbol{\alpha} = \mathbf{F}^{-1} \mathbf{W} \quad (6.97)$$

in which  $\mathbf{W}$  is the vector of nodal deflections. The terms  $\mathbf{SF}^{-1}\mathbf{W}$  and  $\bar{\mathbf{SF}}^{-1}\mathbf{W}$  can then be added to the ones resulting from the boundary integrals in (6.88) and (6.89), respectively, and the final system of equations solved by a direct method after incorporation of the boundary conditions.

Kamiya and Sawaki [13] employed this formulation to analyse a clamped circular plate of radius  $a$  resting on an elastic foundation, subjected to a concentrated load  $P$  at its centre. Table 6.2 shows results for the centre deflection for various magnitudes of the dimensionless parameter  $\lambda (= k a^4 / D)$ , obtained with different numbers of constant boundary elements and 25 internal points. It can be seen that the accuracy of the BEM solution improves as the discretization is refined. Table 6.3 also shows results for the centre deflection, for a fixed discretization with 48 constant boundary elements, varying the number of internal points according to the distribution depicted in figure 6.6. In this case, the solution seems to converge to values which are slightly lower than the exact ones (around 1.5 % error).

$\lambda$	Exact Solution	Number of Boundary Elements		
		48	36	24
0	0.1989	0.1982	0.1977	0.1961
1	0.1973	0.1963	0.1957	0.1940
2	0.1958	0.1945	0.1938	0.1920
3	0.1942	0.1927	0.1919	0.1900
4	0.1927	0.1909	0.1901	0.1881
5	0.1912	0.1891	0.1883	0.1862
6	0.1897	0.1874	0.1865	0.1843
7	0.1882	0.1858	0.1848	0.1825
8	0.1868	0.1841	0.1831	0.1808
9	0.1854	0.1825	0.1814	0.1780
10	0.1839	0.1810	0.1798	0.1773

Table 6.2: Centre Deflection  $D w_c / P a^2 (\times 10)$

		Number of Internal Points			
$\lambda$	Exact Solution	49	25	13	7
0	0.1989	0.1982	0.1982	0.1982	0.1982
1	0.1973	0.1963	0.1963	0.1964	0.1966
2	0.1958	0.1944	0.1945	0.1946	0.1950
3	0.1942	0.1925	0.1927	0.1929	0.1934
4	0.1927	0.1906	0.1909	0.1912	0.1918
5	0.1912	0.1889	0.1891	0.1895	0.1903
6	0.1897	0.1871	0.1874	0.1878	0.1888
7	0.1882	0.1854	0.1858	0.1862	0.1874
8	0.1868	0.1837	0.1841	0.1846	0.1859
9	0.1854		0.1825	0.1831	0.1845
10	0.1839		0.1810	0.1816	0.1831

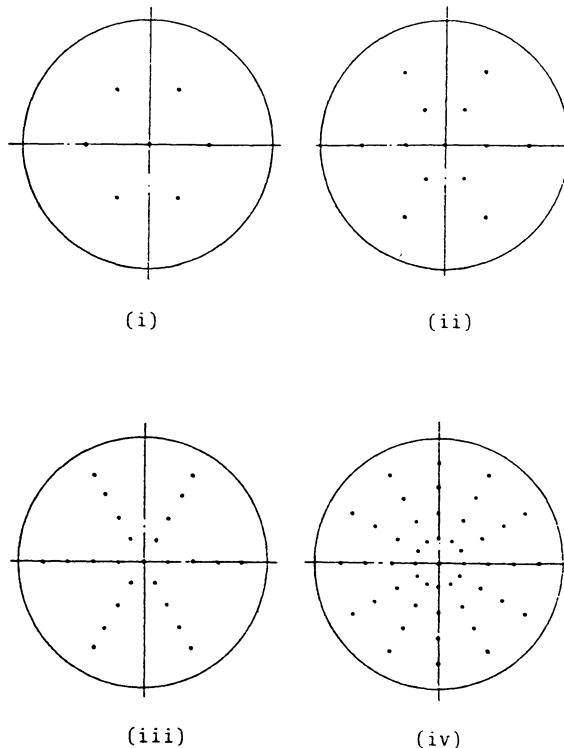
Table 6.3: Centre Deflection  $D w_c / P a^2 (\times 10)$ 

Figure 6.6: Distribution of Internal Points for Circular Plate: (i)49, (ii)25, (iii)13, (iv)7

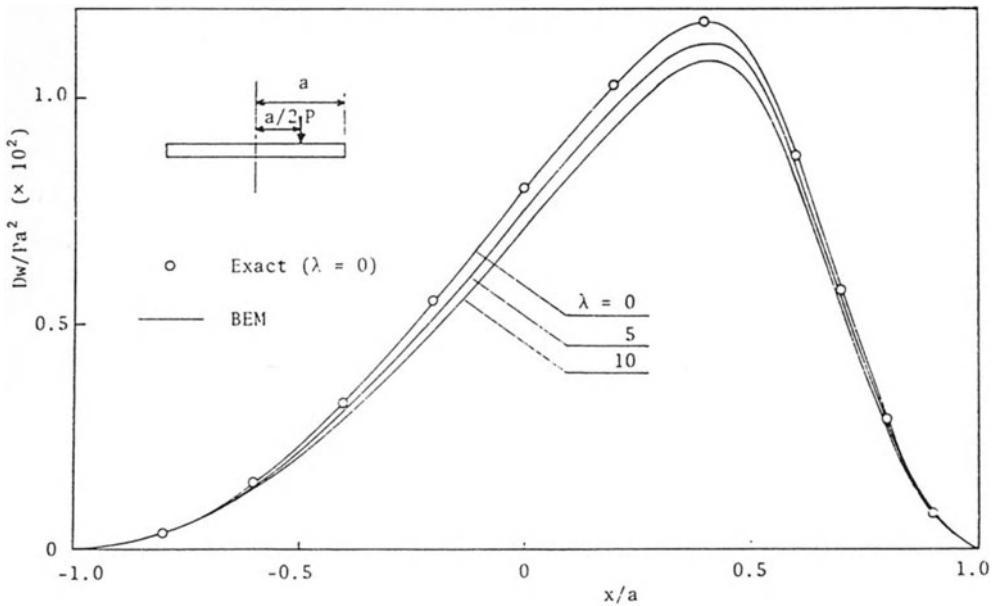


Figure 6.7: Deflection Variation for Circular Plate with Eccentric Load

Another problem studied in [13] was that of a clamped circular plate subjected to an eccentric concentrated load. The deflection variation along the plate diameter, including the load point, is shown in figure 6.7 for three different values of the parameter  $\lambda$ . The DRM discretization employed 48 constant boundary elements and 25 internal points.

The previous formulation for plate bending analysis was extended by Sawaki *et al.* [17] to deal with geometrically non-linear problems. The governing differential equation in this case is the Berger equation for non-linear bending [18]

$$D(\nabla^4 w - k^2 \nabla^2 w) = q \quad (6.98)$$

with the identity

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{1}{2} \left[ \left( \frac{\partial w}{\partial x} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \right] = \frac{k^2 h^2}{12} = \text{constant} \quad (6.99)$$

where  $u, v, w$  correspond to the displacements in the cartesian coordinates directions. Parameter  $k^2$ , sometimes called the Berger parameter, may be estimated using the recognized fact that the Berger equation can produce a fairly good approximation of the problem only when the in-plane displacements are restricted along the plate edges. Thus, one can write from (6.99)

$$k^2 = \frac{6}{A h^2} \int_{\Omega} \left[ \left( \frac{\partial w}{\partial x} \right)^2 + \left( \frac{\partial w}{\partial y} \right)^2 \right] d\Omega \quad (6.100)$$

in which  $A$  is the plate area.

As for the case of linear bending, a fundamental solution is available for the Berger equation, permitting a direct boundary element formulation of the problem in terms of boundary integrals only [19]. However, due to the complexity of this fundamental solution, an alternative formulation has been proposed which makes use of the fundamental solution of the bi-harmonic equation and treats the Laplacian term by the DRM [17]. It follows the same line as for the previous linear problem, and produces a set of integral equations similar to (6.88) and (6.89), with the last domain integral replaced by

$$+ \int_{\Omega} D k^2 \nabla^2 w w^* d\Omega \quad (6.101)$$

and

$$+ \int_{\Omega} D k^2 \nabla^2 w \frac{\partial w^*}{\partial n} d\Omega \quad (6.102)$$

respectively.

In order to transform the above into equivalent boundary integrals, expression (6.101) is initially integrated by parts twice to produce

$$\begin{aligned} D k^2 \int_{\Omega} \nabla^2 w w^* d\Omega &= \\ D k^2 \left[ \int_{\Omega} w \nabla^2 w^* d\Omega + \int_{\Gamma} \left( w^* \frac{\partial w}{\partial n} - w \frac{\partial w^*}{\partial n} \right) d\Gamma \right] \end{aligned} \quad (6.103)$$

Next, the deflection  $w$  at a general point is expressed in the usual DRM form (6.90). By substituting this expression into the domain integral of (6.103), making use of particular solutions  $\hat{w}^j$  of the equation

$$\nabla^2 \hat{w}^j = f^j \quad (6.104)$$

and employing the reciprocity theorem for the Laplacian operator, the domain integral (6.101) becomes

$$\begin{aligned} D k^2 \int_{\Omega} \nabla^2 w w^* d\Omega &= D k^2 \left\{ \sum_{j=1}^{N+L} \alpha^j \left[ -c^i \hat{w}^{ij} \right. \right. \\ &\quad \left. \left. + \int_{\Gamma} \left( \nabla^2 w^* \frac{\partial \hat{w}^j}{\partial n} - \hat{w}^j \frac{\partial}{\partial n} \nabla^2 w^* \right) d\Gamma \right] \right. \\ &\quad \left. + \int_{\Gamma} \left( w^* \frac{\partial w}{\partial n} - w \frac{\partial w^*}{\partial n} \right) d\Gamma \right\} \end{aligned} \quad (6.105)$$

Through a similar procedure, integral (6.102) becomes

$$\begin{aligned}
 D k^2 \int_{\Omega} \nabla^2 w \frac{\partial w^*}{\partial n_0} d\Omega &= D k^2 \left\{ \sum_{j=1}^{N+L} \alpha^j \left[ -c^i \frac{\partial \hat{w}^{ij}}{\partial n_0} \right. \right. \\
 &\quad + \int_{\Gamma} \left( \frac{\partial \nabla^2 w^*}{\partial n_0} \frac{\partial \hat{w}^j}{\partial n} - \hat{w}^j \frac{\partial^2}{\partial n_0 \partial n} \nabla^2 w^* \right) d\Gamma \Big] \\
 &\quad \left. + \int_{\Gamma} \left( \frac{\partial w^*}{\partial n_0} \frac{\partial w}{\partial n} - w \frac{\partial^2 w^*}{\partial n_0 \partial n} \right) d\Gamma \right\}
 \end{aligned} \tag{6.106}$$

The approximating function  $f^j$  adopted in this case was

$$f = 1 - r - r^2 \tag{6.107}$$

with the corresponding particular solution  $\hat{w}^j$

$$\hat{w} = \frac{1}{4}r^2 - \frac{1}{9}r^3 - \frac{1}{16}r^4 \tag{6.108}$$

The numerical solution of the non-linear problem is effected in an iterative way. Starting from the linear solution ( $k^2 = 0$ ), an initial variation of  $w$  is computed; then, the value of the Berger parameter  $k^2$  is estimated and a new iteration carried out, now including the non-linear term. This procedure is repeated until a specified convergence criterion is reached.

Sawaki *et al.* [17] employed this DRM formulation in the analysis of thin elastic plates subjected to a uniform lateral load. Their numerical results were obtained using 36 constant boundary elements and 61 internal points, and are reproduced in figures 6.8 and 6.9. These figures show the maximum deflection of a clamped and a simply-supported plate, respectively. It can be seen that the DRM solutions agree well with the corresponding RKG (Runge-Kutta-Gill) solutions, which are known to be effective for axisymmetric problems.

## 6.4 Three-Dimensional Elasticity

### 6.4.1 Computational Formulation

The equations given in section 6.2 for two-dimensional analysis of elasticity problems are also valid in three dimensions except that the subscript indices now take the values 1,2,3.

The three-dimensional fundamental solution for isotropic problems is [5]

$$u_{ij}^* = \frac{1}{16\pi(1-\nu)Gr} [(3-4\nu)\delta_{ij} + r_{,i}r_{,j}] \tag{6.109}$$

in the case of displacements, with the corresponding surface traction field given by

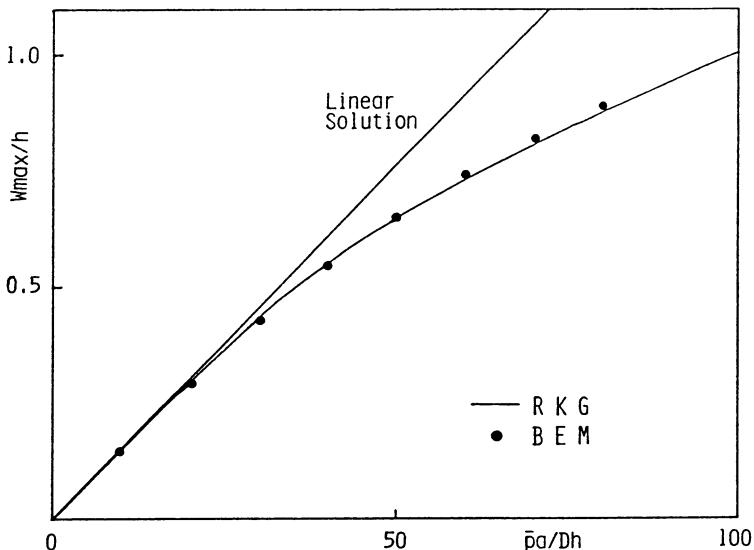


Figure 6.8: Maximum Deflection of Clamped Circular Plate

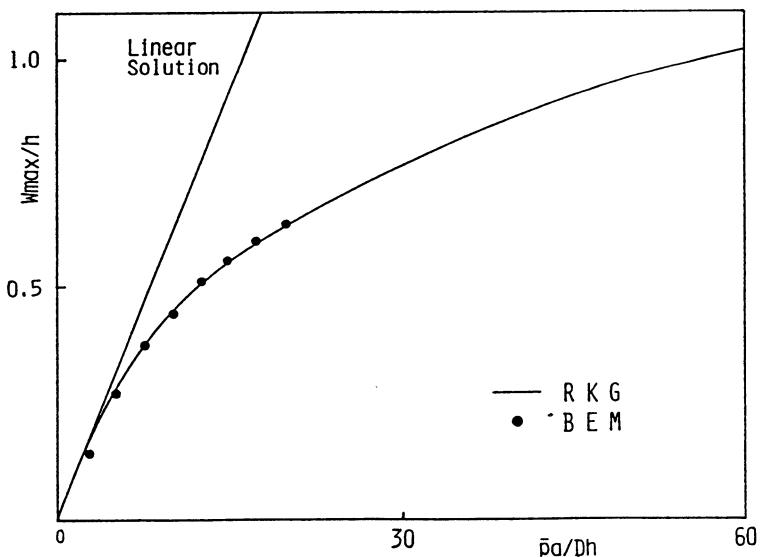


Figure 6.9: Maximum Deflection of Simply-Supported Circular Plate

$$p_{ij}^* = \frac{-1}{8\pi(1-\nu)r^2} \left[ [(1-2\nu)\delta_{ij} + 3r_{,i}r_{,j}] \frac{\partial r}{\partial n} - (1-2\nu)(r_{,i}n_j - r_{,j}n_i) \right] \quad (6.110)$$

The DRM particular solution for displacements in three dimensions is given by

$$\begin{aligned} \hat{u}_{mk} = & \frac{1-2\nu}{(6-4\nu)G} r_{,m} r_{,k} r^2 + \\ & \frac{1}{48(1-\nu)G} \left[ \left( \frac{11}{3} - 4\nu \right) \delta_{mk} - r_{,m} r_{,k} \right] r^3 \end{aligned} \quad (6.111)$$

from which the particular solution for surface tractions may be derived, as

$$\begin{aligned} \hat{p}_{mk} = & \frac{1-2\nu}{3-2\nu} \left[ \frac{1+2\nu}{1-2\nu} r_{,m} n_k + \frac{1}{2} r_{,k} n_m + \frac{1}{2} \delta_{mk} \frac{\partial r}{\partial n} \right] r + \\ & \frac{1}{24(1-\nu)} \left[ (5-6\nu) r_{,k} n_m - (1-6\nu) r_{,m} n_k + [(5-6\nu) \delta_{mk} - r_{,m} r_{,k}] \frac{\partial r}{\partial n} \right] r^2 \end{aligned} \quad (6.112)$$

Equation (6.48), repeated below for convenience,

$$\mathbf{H}\mathbf{u} - \mathbf{G}\mathbf{p} = (\mathbf{H}\hat{\mathbf{U}} - \mathbf{G}\hat{\mathbf{P}})\mathbf{F}^{-1}\mathbf{b} \quad (6.113)$$

is also valid for three-dimensional analysis provided that matrices  $\mathbf{H}$ ,  $\mathbf{G}$ ,  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{P}}$  are evaluated using the preceding four expressions (6.109) to (6.112).

It is convenient to say a few words about indices in order to permit a correct assemblage of the matrix equations. Each term of the matrices  $\mathbf{H}$  and  $\mathbf{G}$  has four indices: two global referring to the source and field nodes, and two local. Thus, each pair of global indices refers to a  $3 \times 3$  submatrix, the indices of which being those used in equations (6.109) and (6.110), which vary from 1 to 3 and refer to the degrees of freedom at each point. Similar considerations apply to the matrices  $\hat{\mathbf{U}}$  and  $\hat{\mathbf{P}}$ . The two global indices in this case refer to field node and DRM collocation point (figure 3.3).

It is important to note that the DRM matrices have their indices transposed in relation to the BEM matrices, as shown in figure 3.3. This refers to both global indices and those of local degrees of freedom at points.

The  $3 \times 3$  submatrices of  $\mathbf{F}$  in equation (6.113) contain non-zero terms only on the leading diagonal. Vector  $\boldsymbol{\alpha} = \mathbf{F}^{-1}\mathbf{b}$  is of size  $3 \times (N + L)$  where  $N$  is the number of boundary nodes and  $L$  the number of internal nodes. Vector  $\boldsymbol{\alpha}$  is calculated using Gauss elimination since  $\mathbf{b}$  is known and no matrix inversions are necessary.

Another important point to observe which refers to BEM elasticity analyses in general, particularly in three dimensions, and not specifically to DRM, is that as surface tractions are discontinuous along edges and at corners where three edges meet, it is necessary to store them referred to elements and not to nodes. Thus, when a surface traction is calculated at a given node in a given direction, it must be associated only with the face which has a restriction in that direction [31].

Given these considerations, unknown displacements and surface tractions may be obtained for a three-dimensional problem with body forces using equation (6.113).

The free terms  $c^i$  (see equation 6.46), now  $3 \times 3$  submatrices, have been incorporated onto the principal diagonal of  $\mathbf{H}$ . Element  $m, n$  of a given  $c^i$  submatrix is the sum of the same term in each of the other submatrices on that row with a change of sign. This can be deduced from rigid body movements in a similar way as described in chapter 2 for potential problems.

Displacements at internal nodes are obtained writing equation (6.44) at these points, noting that the free terms  $c_{lk}$  will now generate unit submatrices. The equation obtained is as follows:

$$u_l^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma + \sum_{j=1}^{N+L} \left( c_{lk}^i \hat{u}_{mk}^{ij} + \int_{\Gamma} p_{lk}^* \hat{u}_{mk}^j d\Gamma - \int_{\Gamma} u_{lk}^* \hat{p}_{mk}^j d\Gamma \right) \alpha_m^j \quad (6.114)$$

In order to obtain the internal stresses equation (6.114) is differentiated to produce the strain tensor, and the result substituted into Hooke's law (equation 6.53),

$$\sigma_{nl}^i = \int_{\Gamma} D_{nlk}^* p_k d\Gamma - \int_{\Gamma} S_{nlk}^* u_k d\Gamma + \sum_{j=1}^{N+L} \left( \hat{D}_{nlm}^{ij} + \int_{\Gamma} S_{nlk}^* \hat{u}_{mk}^j d\Gamma - \int_{\Gamma} D_{nlk}^* \hat{p}_{mk}^j d\Gamma \right) \alpha_m^j \quad (6.115)$$

in which tensors  $D^*$  and  $S^*$  are given by

$$D_{ijk}^* = \frac{1}{8\pi(1-\nu)r^2} [(1-2\nu)(r_{,j}\delta_{ik} + r_{,i}\delta_{jk} - r_{,k}\delta_{ij}) + 5r_{,i}r_{,j}r_{,k}] \quad (6.116)$$

$$S_{ijk}^* = \frac{G}{4\pi(1-\nu)r^3} \{ 3[(1-2\nu)r_{,k}\delta_{ij} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 5r_{,i}r_{,j}r_{,k}] \frac{\partial r}{\partial n} +$$

$$3\nu(r_{,j}r_{,k}n_i + r_{,i}r_{,k}n_j) + (1-2\nu)(3r_{,i}r_{,j}n_k + n_j\delta_{ik} + n_i\delta_{jk}) - (1-4\nu)n_k\delta_{ij} \} \quad (6.117)$$

The signs on expressions (6.116) and (6.117) have been changed due to the fact that at internal points the derivatives of  $r$  are taken with respect to the source point [5,31]. Tensor  $\hat{D}$ , calculated using DRM indices, does not present this feature, and is given by

$$\hat{D}_{ijk} = \frac{(1-2\nu)r}{3-2\nu} \left[ \frac{1-2\nu}{1+2\nu} r_{,k}\delta_{ij} + \frac{1}{2}(r_{,i}\delta_{jk} + r_{,j}\delta_{ik}) \right] +$$

$$\frac{r^2}{24(1-\nu)} [(5-6\nu)(r_{,i}\delta_{kj} + r_{,j}\delta_{ik}) - (1-6\nu)r_{,k}\delta_{ij} - r_{,i}r_{,j}r_{,k}] \quad (6.118)$$

It is noted that tensor  $\hat{D}$  is derived from the free term in equation (6.114) and is therefore not integrated.

In the next sections, results using the above DRM formulation will be presented for problems involving gravitational, centrifugal and thermal loadings. With the appropriate modifications, the formulation may also be extended to anisotropic problems, dynamic analysis and other cases.

### 6.4.2 Gravitational Load

In the case of a structural component loaded only by its self-weight the body force in equation (6.1) is given by

$$b_1 = b_2 = 0$$

$$b_3 = -\gamma g$$

where  $g$  is the acceleration due to gravity and  $\gamma$  is the specific weight of the material. In the case of homogeneous bodies this body force will be a constant acting in the negative  $z$  direction.

As an example, the case of a homogeneous prismatic bar was considered. The bar has unit cross-section and is 2 meters long. The discretization into 28 quadratic eight-node surface elements is shown in figure 6.10.

The surface  $z = 0$  was constrained along its normal direction. In addition to the 86 boundary nodes shown in the figure, results were calculated at seven internal nodes evenly spaced along the negative  $z$ -axis. The physical parameters for the problem are  $E = 10000$ ,  $\nu = 0.3$ ,  $\gamma g = 1$ .

A Gaussian numerical integration scheme with  $10 \times 10$  integration points was used for the evaluation of the regular integrals, together with the Telles' transformation (Appendix 2) for the singular integrals.

The results obtained are shown in table 6.4 together with the exact values. The largest errors are obtained, as expected, near the extremities of the bar.

$z$	Displacements		Normal Stress $\sigma_{zz}$	
	DRM	Exact	DRM	Exact
-0.25	.437 E-4	.469 E-4	1.78	1.78
-0.50	.886 E-4	.875 E-4	1.51	1.50
-0.75	.123 E-3	.122 E-3	1.25	1.25
-1.00	.151 E-3	.150 E-3	1.00	1.00
-1.25	.173 E-3	.172 E-3	0.75	0.75
-1.50	.188 E-3	.183 E-3	0.50	0.50
-1.75	.198 E-3	.197 E-3	0.23	0.25

Table 6.4: Displacements and Stresses for Gravitational Load

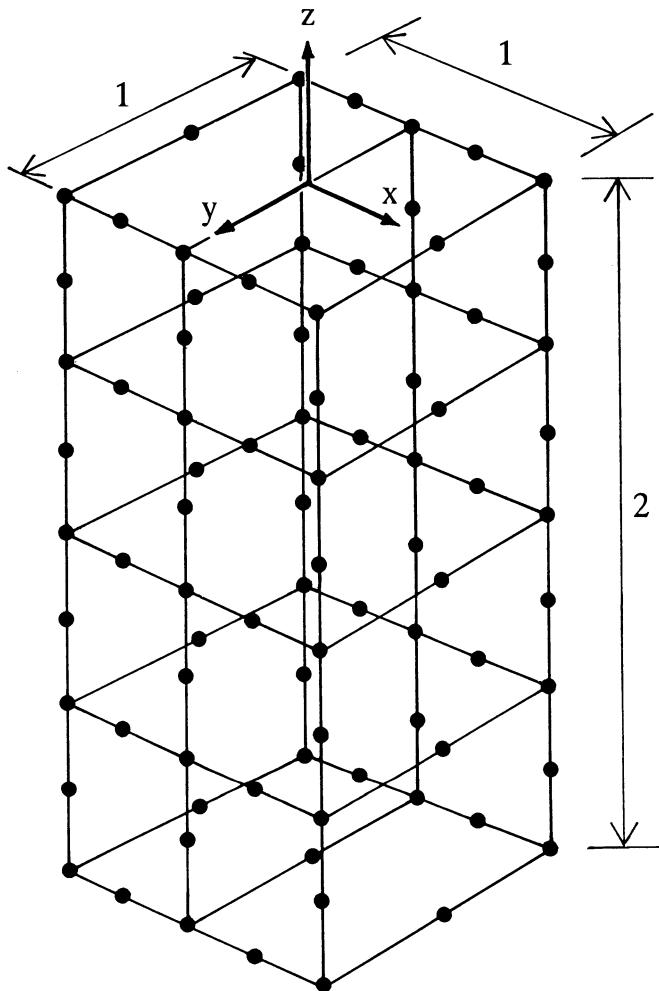


Figure 6.10: Homogeneous Prismatic Bar

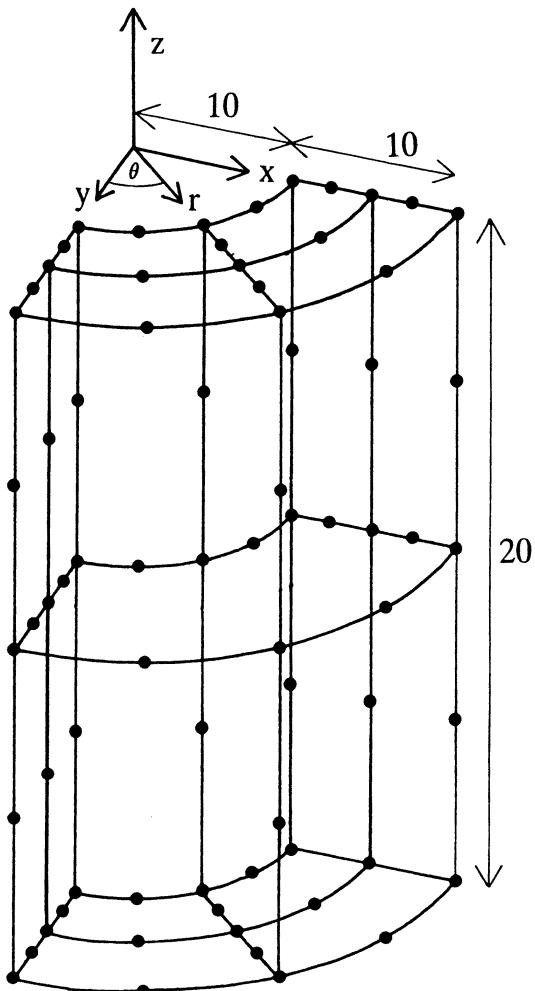


Figure 6.11: One-Quarter of Thick Cylinder

### 6.4.3 Centrifugal Load

In this case a thick cylinder was considered with its axis of rotation coinciding with \$z\$, as shown in figure 6.11.

The faces \$x = 0\$ and \$y = 0\$ were constrained in the direction of their normals. The cylinder was considered to rotate about its axis with an angular velocity \$\omega = 10\$ rad/sec. The body forces are thus given by

$$b_1 = \gamma\omega^2 x$$

$$b_2 = \gamma\omega^2 y$$

$$b_3 = 0$$

where  $(x, y)$  are the coordinates of a given point, as shown in figure 6.11, and  $\gamma = 1$  was assumed.

The cylinder was discretized into 24 quadratic eight-node surface elements, with a total of 74 boundary nodes; 3 internal nodes evenly spaced along the cylinder radius at  $z = -10$ ,  $\theta = 45^\circ$  were considered. Values of the physical constants were taken as  $E = 210000$  and  $\nu = 0.3$ .

Results for displacements and stresses at the three internal nodes are shown in table 6.5, compared with the exact values.

$r$	u		$\sigma_t \times 10^3$		$\sigma_r \times 10^3$	
	Exact	DRM	Exact	DRM	Exact	DRM
12.5	1.57	1.59	27.4	28.0	3.6	3.7
15.0	1.52	1.54	22.6	22.9	4.0	4.0
17.5	1.49	1.51	18.7	18.7	2.6	2.7

Table 6.5: Displacements and Stresses for Centrifugal Load

#### 6.4.4 Thermal Load

The effect of a temperature variation on an elastic material is equivalent to the introduction of an initial strain [31]

$$\epsilon_{jk}^0 = \alpha\theta\delta_{jk}$$

where  $\alpha$  is the coefficient of thermal expansion and  $\theta$  is the change in temperature. The equivalent initial stress may be obtained from Hooke's Law

$$\sigma_{jk}^0 = - \left( \frac{2G\nu}{1-2\nu} \epsilon_{ii}^0 \delta_{jk} + 2G\epsilon_{jk}^0 \right)$$

such that, in three dimensions,

$$\sigma_{jk}^0 = -2G \frac{1+\nu}{1-2\nu} \alpha\theta\delta_{jk} = \chi\theta\delta_{jk}$$

where

$$\chi = -2G \frac{1+\nu}{1-2\nu} \alpha$$

If no other body forces exist, the boundary integral expression including initial strains is [31]

$$c_{lk}^i u_k^i = \int_{\Gamma} u_{lk}^* p_k d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma - \int_{\Omega} \sigma_{jk}^0 \epsilon_{ijk}^* d\Omega \quad (6.119)$$

where  $\epsilon_{ijk}^*$  may be found using the procedure given in section 6.2.2.

Substituting for the initial stress in the domain integral, several different but equivalent expressions may be found

$$\int_{\Omega} \sigma_{jk}^0 \epsilon_{ijk}^* d\Omega = \int_{\Omega} \chi \theta \delta_{jk} \epsilon_{ijk}^* d\Omega$$

$$= \int_{\Omega} \chi \theta u_{lk,k}^* d\Omega$$

$$= \int_{\Omega} \chi \theta_{,k} u_{lk}^* d\Omega - \int_{\Gamma} \chi \theta u_{lk}^* n_k d\Gamma$$

any of which may be used in BEM analyses.

For the DRM the third above expression is more convenient as the domain integral is of a similar form to that of equation (6.25), which was used in the derivation of equation (6.113).

If the first or second expressions above are used a DRM formulation is still possible; however, matrices  $\mathbf{H}$  and  $\mathbf{G}$  on the right side of (6.113) will have to be replaced by matrices calculated starting from  $u_{lk,k}^*$ . This will also lead to modifications in expressions (6.114) and (6.115).

As can be seen in the above equation, the formulation leads to an additional boundary integral. This term can, however, be combined with that of equation (6.119) in the form

$$c_{lk}^i u_k^i = \int_{\Gamma} u_{lk}^* (p_k + \chi \theta n_k) d\Gamma - \int_{\Gamma} p_{lk}^* u_k d\Gamma - \int_{\Omega} \chi \theta_{,k} u_{lk}^* d\Omega \quad (6.120)$$

The above formulation was applied to the bar shown in figure 6.10, which was subjected to a temperature field such that the face  $x = -0.5$  was kept at  $\theta = 50^\circ$  and the face  $x = 0.5$  was kept at  $\theta = -50^\circ$ . The temperature at any point is thus given by

$$\theta = -100x$$

If the bar is fixed at its extremities the normal stresses are given by [32]

$$\sigma_{zz} = -100E\alpha x$$

For the case considered  $E = 10000$ ,  $\alpha = 0.00001$ ,  $\nu = 0.3$  producing  $\chi = 0.25$ .

Results are given in table 6.6 for internal nodes located on an axis parallel to  $x$  passing through the baricenter of the beam. Results obtained for internal points located near the boundary showed some loss of accuracy as expected.

$x$	$\sigma_{zz}$	Exact
0	0.0	0.0
0.1	-1.09	-1.0
0.2	-2.18	-2.0
0.3	-3.30	-3.0

Table 6.6: Stresses for Thermal Load

## 6.5 Transient Convection-Diffusion

The solution of convection-diffusion problems is a difficult task for all numerical methods because of the nature of the governing equation, which includes first-order and second-order partial derivatives in space. According to the value of the Péclet number, the equation becomes parabolic (for diffusion-dominated processes) or hyperbolic (for convection-dominated processes). Traditional finite difference and finite element algorithms are generally accurate for solving the former but not the latter, in which case oscillations and smoothing of the wave front are introduced. This can be interpreted as an “artificial diffusion” intrinsic to these methods [20],[21].

Applications of the Boundary Element Method for steady-state convection-diffusion using the fundamental solution of the complete equation have shown that the BEM seems to be relatively free from these problems [22-24]. This was also the case for some transient applications using formulations with time-dependent fundamental solutions [25-26]. The main restriction of these formulations, however, is the fact that fundamental solutions are only available for equations with constant coefficients, or coefficients with very simple variations in space [27].

Alternative BEM formulations for transient convection-diffusion have been proposed, using the fundamental solution of the diffusion equation and iterating over the convective terms [28] or using the fundamental solution to Laplace’s equation and a dual reciprocity treatment of all the remaining terms (section 5.6). In this section, a different dual reciprocity approach is formulated using the fundamental solution of the corresponding steady-state equation. The resulting domain integral is transformed into equivalent boundary integrals by introducing particular solutions which satisfy an associated non-homogeneous steady-state convection-diffusion equation. Although only problems with constant velocity are analysed herein, the formulation can be extended to deal with variable velocity fields using a similar dual reciprocity approach (see section 5.6 and [29]).

The two-dimensional transient convection-diffusion equation including first-order reaction can be written in the form

$$D\nabla^2\phi - v_x \frac{\partial\phi}{\partial x} - v_y \frac{\partial\phi}{\partial y} - k\phi = \frac{\partial\phi}{\partial t} \quad (6.121)$$

where  $v_x$  and  $v_y$  are the components of the velocity vector  $\mathbf{v}$ ,  $D$  is the diffusivity coefficient (assuming the medium is homogeneous and isotropic) and  $k$  represents the

reaction coefficient. The variable  $\phi$  can be interpreted as temperature for heat transfer problems, concentration for dispersion problems, etc. The mathematical description of the problem is complemented by boundary conditions of the Dirichlet, Neumann or Robin (mixed) types, and by initial conditions at time  $t_0$ .

The above differential equation can be transformed into an equivalent integral equation by applying a weighted residual technique. Starting with the weighted residual statement

$$\int_{\Omega} \left( D\nabla^2\phi - v_x \frac{\partial\phi}{\partial x} - v_y \frac{\partial\phi}{\partial y} - k\phi \right) \phi^* d\Omega = \int_{\Omega} \frac{\partial\phi}{\partial t} \phi^* d\Omega \quad (6.122)$$

in which  $\phi^*$  is the fundamental solution of the corresponding steady-state equation, and integrating by parts twice the Laplacian and once the first-order space derivatives, the following equation is obtained

$$\phi^i - D \int_{\Gamma} \phi^* \frac{\partial\phi}{\partial n} d\Gamma + D \int_{\Gamma} \phi \frac{\partial\phi^*}{\partial n} d\Gamma + \int_{\Gamma} \phi\phi^* v_n d\Gamma = - \int_{\Omega} \frac{\partial\phi}{\partial t} \phi^* d\Omega \quad (6.123)$$

where  $v_n = \mathbf{v} \cdot \mathbf{n}$ ,  $\mathbf{n}$  is the unit outward normal vector and the dot stands for scalar product.

The function  $\phi^*$  above is the solution of

$$D\nabla^2\phi^* + v_x \frac{\partial\phi^*}{\partial x} + v_y \frac{\partial\phi^*}{\partial y} - k\phi^* = -\Delta^i \quad (6.124)$$

in which  $\Delta^i$  represents the Dirac delta function at a source point  $i$ . It can be noticed that the sign of the first-derivative terms is reversed in (6.121) and (6.124), since this operator is not self-adjoint. For two-dimensional problems,  $\phi^*$  is of the form

$$\phi^* = \frac{1}{2\pi D} e^{-\frac{\mathbf{v} \cdot \mathbf{r}}{2D}} K_0(\mu r) \quad (6.125)$$

where

$$\mu = \left[ \left( \frac{|\mathbf{v}|}{2D} \right)^2 + \frac{k}{D} \right]^{\frac{1}{2}} \quad (6.126)$$

and  $r$  is the modulus of  $\mathbf{r}$ , the distance vector between the source and field points. The derivative of the fundamental solution with respect to the outward normal direction is given by

$$\frac{\partial\phi^*}{\partial n} = \frac{1}{2\pi D} e^{-\frac{\mathbf{v} \cdot \mathbf{r}}{2D}} \left[ -\mu K_1(\mu r) \frac{\partial r}{\partial n} - \frac{v_n}{2D} K_0(\mu r) \right] \quad (6.127)$$

In the above,  $K_0$  and  $K_1$  are Bessel functions of second kind, of orders zero and one, respectively.

Equation (6.123) is valid for source points  $i$  on the domain  $\Omega$ . In order to obtain an integral equation for boundary points, the source point is taken to the boundary and a limit analysis carried out due to the jump of  $\partial\phi^*/\partial n$ . The result is the equation

$$c^i \phi^i - D \int_{\Gamma} \phi^* \frac{\partial\phi}{\partial n} d\Gamma + D \int_{\Gamma} \phi \frac{\partial\phi^*}{\partial n} d\Gamma + \int_{\Gamma} \phi\phi^* v_n d\Gamma = - \int_{\Omega} \frac{\partial\phi}{\partial t} \phi^* d\Omega \quad (6.128)$$

in which  $c^i$  is a function of the internal angle the boundary  $\Gamma$  makes at point  $i$  [5]. The expression for  $c^i$  in the present case is the same as for potential problems governed by Laplace's equation, since both fundamental solutions present the same type of singularity.

In order to obtain a boundary integral which is equivalent to the domain integral in equations (6.123) and (6.128), a dual reciprocity approximation is introduced. The basic idea is to expand the time-derivative  $\partial\phi/\partial t$  in the form,

$$\dot{\phi} = \sum_{j=1}^{N+L} f_j(x, y) \alpha_j(t) \quad (6.129)$$

where the dot denotes temporal derivative. The above series involves a set of known functions  $f_j$  which are dependent only on geometry, and a set of unknown coefficients  $\alpha_j$  which are time-dependent only. With this approximation, the domain integral becomes

$$\int_{\Omega} \dot{\phi} \phi^* d\Omega = \sum_{j=1}^{N+L} \alpha_j \int_{\Omega} f_j \phi^* d\Omega \quad (6.130)$$

The next step is to consider that, for each function  $f_j$ , there exists a related function  $\hat{\phi}_j$  which is a particular solution of the equation

$$D\nabla^2 \hat{\phi} - v_x \frac{\partial \hat{\phi}}{\partial x} - v_y \frac{\partial \hat{\phi}}{\partial y} - k \hat{\phi} = f \quad (6.131)$$

Thus, the domain integral can be recast in the form

$$\int_{\Omega} \dot{\phi} \phi^* d\Omega = \sum_{j=1}^{N+L} \alpha_j \int_{\Omega} \left( D\nabla^2 \hat{\phi}_j - v_x \frac{\partial \hat{\phi}_j}{\partial x} - v_y \frac{\partial \hat{\phi}_j}{\partial y} - k \hat{\phi}_j \right) \phi^* d\Omega \quad (6.132)$$

Substituting expansion (6.132) into equation (6.128), and applying integration by parts also to the right side of the resulting equation, one finally arrives at a boundary integral equation of the form

$$c^i \phi^i - D \int_{\Gamma} \phi^* \frac{\partial \phi}{\partial n} d\Gamma + D \int_{\Gamma} \phi \frac{\partial \phi^*}{\partial n} d\Gamma + \int_{\Gamma} \phi \phi^* v_n d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left[ c^i \hat{\phi}_j^i - D \int_{\Gamma} \phi^* \frac{\partial \hat{\phi}_j}{\partial n} d\Gamma + D \int_{\Gamma} \hat{\phi}_j \frac{\partial \phi^*}{\partial n} d\Gamma + \int_{\Gamma} \hat{\phi}_j \phi^* v_n d\Gamma \right] \quad (6.133)$$

For the numerical solution of the problem, equation (6.133) is written in a discretized form in which the integrals over the boundary are approximated by a summation of integrals over individual boundary elements, *i.e.*

$$c^i \phi^i - D \sum_{k=1}^N \int_{\Gamma_k} \phi^* \frac{\partial \phi}{\partial n} d\Gamma + D \sum_{k=1}^N \int_{\Gamma_k} \left( \frac{\partial \phi^*}{\partial n} + \frac{v_n}{D} \phi^* \right) \phi d\Gamma = \sum_{j=1}^{N+L} \alpha_j \left[ c^i \hat{\phi}_j^i - D \sum_{k=1}^N \int_{\Gamma_k} \phi^* \frac{\partial \hat{\phi}_j}{\partial n} d\Gamma + D \sum_{k=1}^N \int_{\Gamma_k} \left( \frac{\partial \phi^*}{\partial n} + \frac{v_n}{D} \phi^* \right) \hat{\phi}_j d\Gamma \right] \quad (6.134)$$

In the above equation, it can be seen that

$$\frac{\partial \phi^*}{\partial n} + \frac{v_n}{D} \phi^* = \frac{1}{2\pi D} e^{-\frac{|{\bf v} \cdot {\bf r}|}{2D}} \left[ -\mu K_1(\mu r) \frac{\partial r}{\partial n} + \frac{v_n}{2D} K_0(\mu r) \right] \quad (6.135)$$

Applying equation (6.134) to all boundary nodes, introducing the approximations for  $\phi$ ,  $q = \partial\phi/\partial n$ ,  $\hat{\phi}$  and  $\hat{q} = \partial\hat{\phi}/\partial n$  within each boundary element, results in the following system of equations

$${\bf H}\phi - {\bf G}q = ({\bf H}\hat{\phi} - {\bf G}\hat{Q})\alpha \quad (6.136)$$

In the above system,  $\mathbf{G}$  and  $\mathbf{H}$  are square matrices whose coefficients are calculated by integrating, over each boundary element, products of  $\phi^*$  and  $\partial\phi^*/\partial n$  by the interpolation functions. Numerical quadrature schemes appropriate for evaluation of these coefficients are discussed in [24].

By evaluating expression (6.129) at all boundary nodes and inverting, one arrives at

$$\alpha = {\bf F}^{-1} \dot{\phi} \quad (6.137)$$

which, substituted into equation (6.136) results in

$${\bf C}\dot{\phi} + {\bf H}\phi = {\bf G}q \quad (6.138)$$

with

$${\bf C} = -({\bf H}\hat{\phi} - {\bf G}\hat{Q}){\bf F}^{-1}$$

System (6.138) can be integrated in time using standard time-stepping procedures. It is noted that the coefficients of matrices  $\mathbf{H}$ ,  $\mathbf{G}$  and  $\mathbf{C}$  all depend on geometry only, thus they can be computed once and stored.

Employing the general two-level time integration scheme discussed in section 5.2, the following discrete form is obtained

$$\left( \frac{1}{\Delta t} {\bf C} + \theta {\bf H} \right) \phi^{m+1} - \theta {\bf G} q^{m+1} = \left[ \frac{1}{\Delta t} {\bf C} - (1-\theta) {\bf H} \right] \phi^m + (1-\theta) {\bf G} q^m \quad (6.139)$$

Upon introducing the boundary conditions at time  $(m+1)\Delta t$ , the left side of the equation can be rearranged and the resulting system solved by using a standard direct procedure like Gauss elimination.

Rather than using the same simple form of the approximating function  $f$  which has been used throughout this book, *i.e.*  $f = 1 + r$ , it was decided in the present case to start with a simple form of particular solution  $\hat{\phi}$  and find the corresponding expression for  $f$  by direct substitution into (6.131). The resulting expressions are:

$$\hat{\phi} = r^3$$

$$\hat{q} = 3r(r_x n_x + r_y n_y)$$

$$f = 9Dr - 3r(r_x v_x + r_y v_y) - kr^3$$

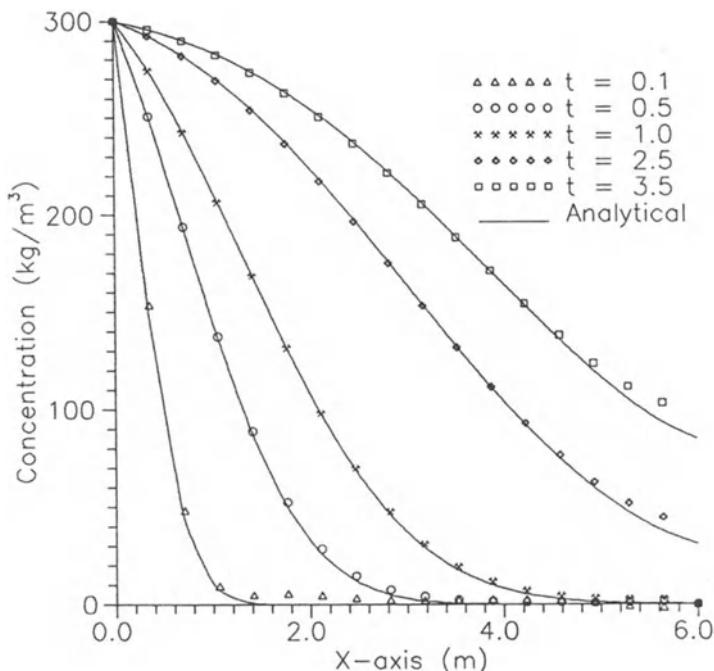
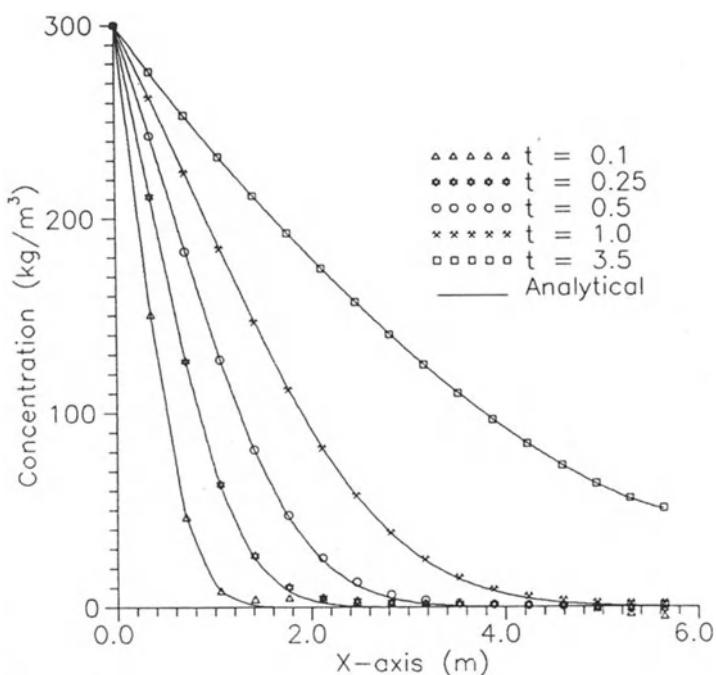
in which  $r_x$  and  $r_y$  are the components of  $\mathbf{r}$  in the direction of the  $x$  and  $y$  axes, respectively.

The above choice was motivated by a previous successful experience with axisymmetric diffusion problems in which a similar approach was used (see section 5.4), and produced a set of functions  $f$  which depend not only on  $r$  but also on the velocity components and the reaction rate. This is a more consistent set of approximating functions for this kind of problem, in line with the behaviour of the fundamental solution (6.125).

The present dual reciprocity boundary element formulation was applied to a one-dimensional convection-diffusion problem with initial condition  $\phi = 0$  and boundary conditions  $\phi = 300 \text{ kg/m}^3$  at  $x = 0$  and  $\partial\phi/\partial n = 0$  at  $x = L$ . All physical parameters were assumed constant, with  $D = 1\text{m}^2/\text{s}$  and different values considered for  $v_x$  and  $k$ , *i.e.*  $v_x = 1$  or  $6 \text{ m/s}$  and  $k = 0, 0.278$  or  $1.389 \text{ s}^{-1}$ . The problem was studied as two-dimensional with cross-section  $6.0 \times 0.7 \text{ m}$ , with the boundary condition  $\partial\phi/\partial n = 0$  specified at the edges parallel to the  $x$ -axis.

For all cases, the discretization consisted of 38 linear boundary elements and one internal pole at the centre of the rectangular region, totalling 39 unknowns. The discontinuity of the normal flux at corners was considered by allowing corner nodes to have 3 degrees of freedom, *i.e.*  $\phi, \partial\phi/\partial n$  before the corner and  $\partial\phi/\partial n$  after the corner, and prescribing 2 of these values. It is noted that the use of double nodes is not permitted with the dual reciprocity scheme because it leads to a singular matrix  $\mathbf{F}$  which cannot be inverted.

Results for the variation of  $\phi$  along  $x$ , at several time levels, are presented in figures 6.12 to 6.17 and compared to analytical solutions given by van Genuchten and Alves [30]. The BEM results were obtained using  $\theta = 1$  and  $\Delta t = 0.05 \text{ s}$ . It can be seen from the graphs that the accuracy of the dual reciprocity boundary element formulation is very good in all cases, with no oscillations and only a minor damping of the wave front.

Figure 6.12: Results for  $v_x = 1\text{m/s}$ ,  $k = 0$ .Figure 6.13: Results for  $v_x = 1\text{m/s}$ ,  $k = 0.278\text{s}^{-1}$

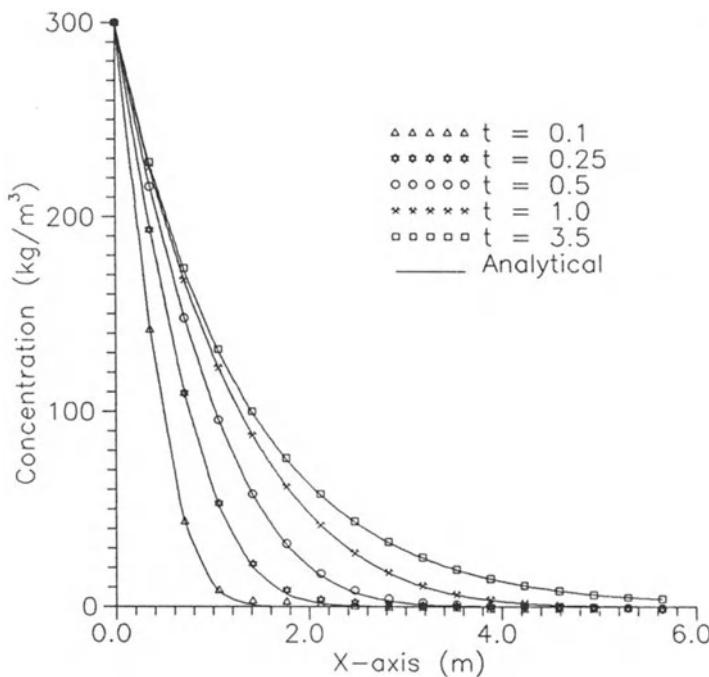


Figure 6.14: Results for  $v_x = 1\text{m/s}$ ,  $k = 1.389\text{s}^{-1}$

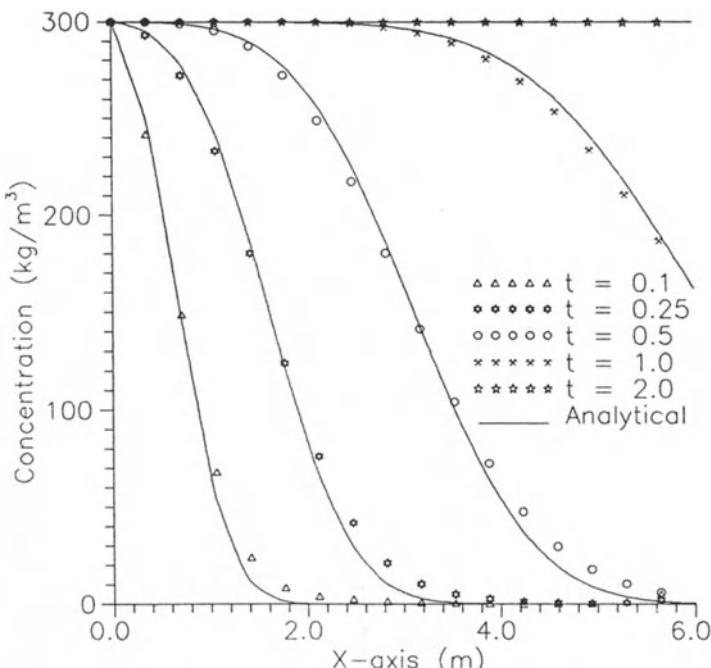
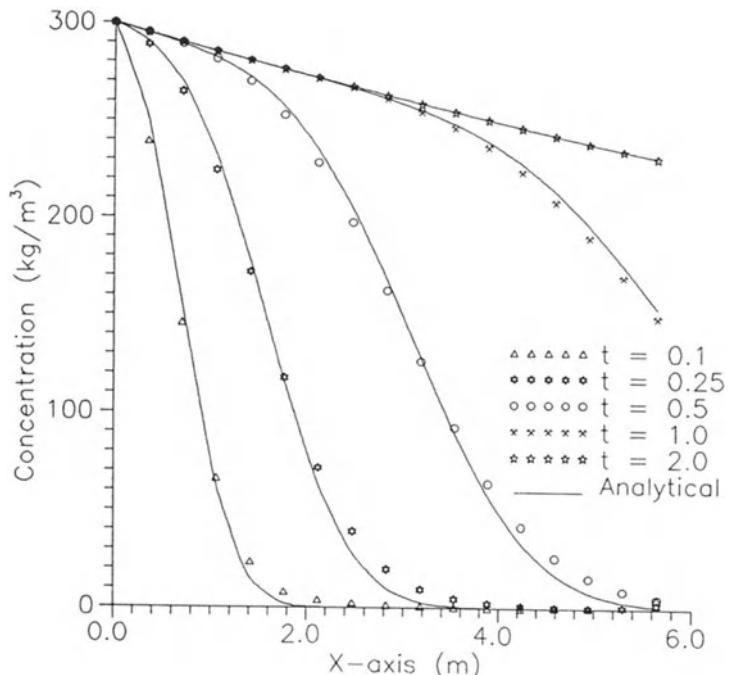
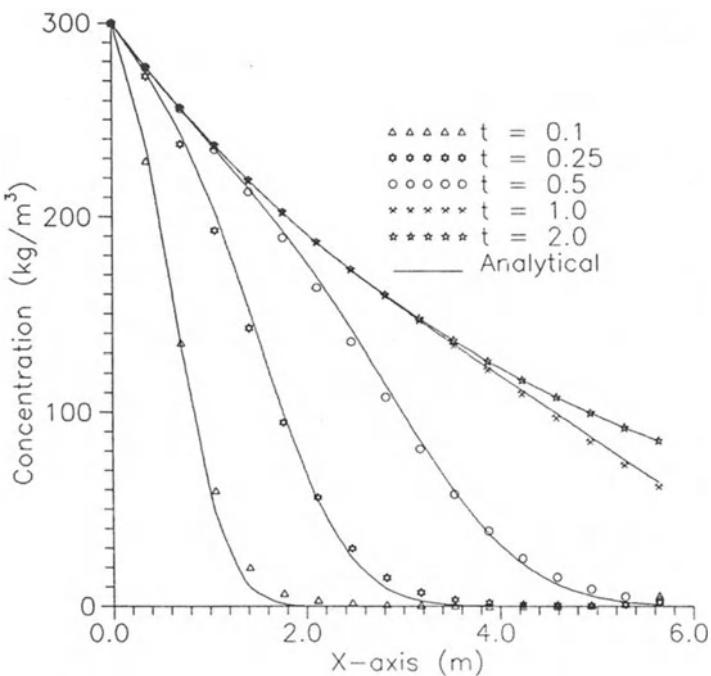


Figure 6.15: Results for  $v_x = 6\text{m/s}$ ,  $k = 0$ .

Figure 6.16: Results for  $v_x = 6\text{m/s}$ ,  $k = 0.278\text{s}^{-1}$ Figure 6.17: Results for  $v_x = 6\text{m/s}$ ,  $k = 1.389\text{s}^{-1}$

## 6.6 References

1. Love, A. E. H.  
A Treatise on the Mathematical Theory of Elasticity, Dover, New York, 1944.
2. Malvern, L. E.  
Introduction to the Mechanics of a Continuous Medium, Prentice-Hall, Englewood Cliffs, N.J., 1969.
3. Timoshenko, S. P. and Goodier, J. N.  
Theory of Elasticity, 3rd edn, Mc-Graw Hill, Tokyo, 1970.
4. Somigliana, C.  
Sopra l'Equilibrio di un Corpo Elastico Isotropo, *Il Nuovo Ciemento*, pp. 17-19, 1886.
5. Brebbia, C. A., Telles, J. C. F. and Wrobel, L. C.  
Boundary Element Techniques, Springer-Verlag, Berlin and New York, 1984.
6. Hartmann, F.  
Computing the C-Matrix in Non-smooth Boundary Points, in *New Developments in Boundary Element Methods*, Computational Mechanics Publications, Southampton, 1980.
7. Brebbia, C. A. and Nardini, D.  
Dynamic Analysis in Solid Mechanics by an Alternative Boundary Element Procedure, *International Journal of Soil Dynamics and Earthquake Engineering*, Vol. 2, No. 4, pp 228-233, 1983.
8. Nardini, D. and Brebbia, C. A.  
Boundary Integral Formulation of Mass Matrices for Dynamic Analysis, in *Topics in Boundary Element Research*, Vol. 2, Springer-Verlag, Berlin and New York, 1985.
9. Wilkinson, J. H.  
The Algebraic Eigenvalue Problem, Clarendon Press, Oxford, 1965.
10. Nardini, D. and Brebbia, C. A.  
Transient Boundary Element Elastodynamics Using the Dual Reciprocity Method and Modal Superposition, in *Boundary Elements VIII*, Vol. 1, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1986.
11. Dong, S. B.  
A Block-Stodola Eigensolution Technique for Large Algebraic Systems with Non-symmetrical Matrices, *International Journal for Numerical Methods in Engineering*, Vol. 11, No. 2, pp 247-269, 1977.
12. Bathe, K. J. and Wilson, E. L.  
Stability and Accuracy Analysis of Direct Integration Methods, *International Journal of Earthquake Engineering and Structural Dynamics*, Vol. 1, pp 283-291, 1973.
13. Kamiya, N. and Sawaki, Y.  
The Plate Bending Analysis by the Dual Reciprocity Boundary Elements, *Engineering Analysis*, Vol. 5, No. 1, pp 36-40, 1988.

14. Silva, N. A. and Venturini, W. S.  
Dual Reciprocity Process Applied to Solve Bending Plates on Elastic Foundations, in *Boundary Elements X*, Vol. 3, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1988.
15. Katsikadelis, J. T. and Armenakas, A. E.  
Plates on Elastic Foundation by BIE Method, *Journal of Engineering Mechanics, ASCE*, Vol. 110, pp. 1086-1104, 1984.
16. Costa Jr., J. A. and Brebbia, C. A.  
Plate Bending Problems Using BEM, in *Boundary Elements VI*, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1984.
17. Sawaki, Y., Takeuchi, K. and Kamiya, N.  
Finite Deflection Analysis of Plates by Dual Reciprocity Boundary Elements, in *Boundary Elements XI*, Vol. 3, Computational Mechanics Publications, Southampton and Springer-Verlag, Berlin and New York, 1989.
18. Berger, H. M.  
A New Approach to the Analysis of Large Deflections of Plates, *Journal of Applied Mechanics*, Vol. 32, pp 465-472, 1955.
19. Sladek, J. and Sladek, V.  
The BIE Analysis of the Berger Equation, *Ingenieur-Archiv*, Vol. 53, pp 385-397, 1983.
20. Roache, P. J.,  
Computational Fluid Dynamics, Hermosa Publishers, Albuquerque, New Mexico, USA, 1972.
21. Hughes, T. J. R. (Ed.),  
Finite Element Methods for Convection Dominated Flows, ASME AMD-Vol. 34, New York, USA, 1979.
22. Ikeuchi, M. and Onishi, K.,  
Boundary Element Solutions to Steady Convective Diffusion Equations, *Applied Mathematical Modelling*, Vol. 7, No. 2, pp 115-118, 1983.
23. Okamoto, N.,  
Boundary Element Method for Chemical Reaction System in Convective Diffusion, in *Numerical Methods in Laminar and Turbulent Flow IV*, Pineridge Press, Swansea, UK, 1985.
24. DeFigueiredo, D. B. and Wrobel, L. C.,  
A Boundary Element Analysis of Convective Heat Diffusion Problems, in *Advanced Computational Methods in Heat Transfer*, Vol. 1, Computational Mechanics Publications, Southampton, and Springer-Verlag, Berlin and New York, 1990.
25. Ikeuchi, M. and Onishi, K.,  
Boundary Elements in Transient Convective Diffusion Problems, in *Boundary Elements V*, Computational Mechanics Publications, Southampton, and Springer-Verlag, Berlin and New York, 1983.

26. Tanaka, Y., Honma, T. and Kaji, I.,  
Transient Solutions of a Three-Dimensional Convective Diffusion Equation Using Mixed Boundary Elements, in *Advanced Boundary Element Methods*, Springer-Verlag, Berlin, 1987.
27. Okubo, A. and Karweit, M. J.,  
Diffusion from a Continuous Source in a Uniform Shear Flow, *Limnology and Oceanography*, Vol. 14, No. 4, pp 514-520, 1969.
28. Brebbia, C. A. and Skerget, L.  
Time Dependent Diffusion Convection Problems Using Boundary Elements, in *Numerical Methods in Laminar and Turbulent Flow III*, Pineridge Press, Swansea, 1983.
29. DeFigueiredo, D. B.,  
Boundary Element Analysis of Convection-Diffusion Problems, Ph.D. Thesis, Wessex Institute of Technology, Southampton, 1990.
30. van Genuchten, M. Th. and Alves, W. J.,  
Analytical Solutions of the One-Dimensional Convective-Dispersive Solute Transport Equation, U.S. Department of Agriculture, Technical Bulletin No. 1661, Washington, D.C., 1982.
31. Brebbia, C. A. and Dominguez, J.  
Boundary Elements: An Introductory Course, Computational Mechanics Publications and McGraw Hill, Southampton and New York, 1989
32. El-Zafrany, A., Cookson, R. A. and Iqbal, M.  
Boundary Element Stress Analysis with Domain Type Loading, in *Advances in the Use of the Boundary Element Method for Stress Analysis*, Mechanical Engineering Publications, London, 1986.

# Chapter 7

## Conclusions

It has been shown throughout this book that the Dual Reciprocity Boundary Element Method possesses the ability to solve a wide range of engineering problems involving different types of governing equations without generating domain integrals. Application of the method to Poisson equations with non-homogeneous terms dependent on position only was considered in chapter 3. The formulation was extended in chapter 4 to include domain integrals dependent on the problem variable as well as position, and in chapter 5 to time-dependent problems. Chapter 6 generalized the use of the method to cases involving fundamental solutions other than that of Laplace's equation. The method has thus been seen to be general, being the only practical way of handling any type of domain integral in boundary element analysis other than cell integration, and with the advantage of not requiring any internal discretization.

The internal nodes used in DRM analysis are usually defined at positions where the solution is required. Their definition is not normally a condition to obtain accurate solutions, except in the case of problems with homogeneous boundary conditions; however, the use of some internal nodes is important in most cases in order to improve the solution accuracy. Results are normally not very sensitive to number and location of the internal nodes and in a specific case this can be determined by a convergence test. Based on our own experience a number of internal nodes  $L = N/2$ , where  $N$  is the number of boundary nodes, provides solutions which are satisfactory for practical engineering problems.

If the exact position of these nodes is not important, they may be automatically distributed by the computer. In any case, the order in which they are defined is not important since only the coordinates are needed as data.

The localized particular solutions  $\hat{u}$  are usually obtained by proposing an approximating function  $f$  and solving the modified equation (*i.e.* Laplace for the case of Poisson-type problems) to find  $\hat{u}$ . There is, in principle, no restriction on the type of function to be used for  $f$ , except that it must produce a non-singular matrix  $\mathbf{F}$ . So far, expansions based on the distance function  $r$  have been shown to be the most accurate and convenient, particularly  $f = 1 + r$  which was used for the majority of examples in this book. It has also been shown that the inclusion of higher-order terms in the  $f$  expansion is not normally necessary. An objection may arise that in using

the DRM it is necessary to invert matrix  $\mathbf{F}$ . The authors do not consider this to be a serious limitation of the method given the state-of-the-art of computer technology, both in terms of memory and speed. It may be noted that no matrix inversion is necessary if the domain terms depend only on position, and that since the coefficients of  $\mathbf{F}$  depend only on geometry, the matrix may be inverted once and stored in a data file for use with all subsequent runs with the same discretization.

In the treatment of general equations considered in chapter 6, it was seen that for elasticity problems it is possible to apply the expansion  $f = 1 + r$  in the same way as for Poisson-type problems. In the case of the transient convection-diffusion equation a more general treatment was given which may be employed when considering cases involving other fundamental solutions not included in this text.

With the material included in this book the DRM may be easily extended to many other problems, and the authors expect a rapid expansion in the use of the method in the near future.

# Appendix 1

## Numerical Integration:

### Regular Integrals

## 1 Introduction

The numerical integration formulae given in this Appendix are used for integration over boundary elements which do not contain the source point. If the element under consideration contains the source point the integrals become singular, and then either analytical integration is used (for simple cases) or the formulae given in Appendix 2 should be employed.

The formulae given here are all based on Gaussian quadrature, which is simple to use and accurate for regular integrals. These formulae may also be used for integration over internal cells.

## 2 One-Dimensional Gaussian Quadrature

This procedure is used to integrate over one-dimensional boundary elements in two-dimensional problems. Each integral is written as

$$I = \int_{-1}^{+1} f(\xi) d\xi = \sum_{i=1}^n f(\xi_i) w_i \quad (1)$$

where  $n$  is the number of integration points,  $\xi_i$  is the local coordinate of the  $i$ th integration point and  $w_i$  is the associated weight factor.

Values of  $\xi_i$  and  $w_i$  are listed in table 1 for  $n$  varying from 2 to 10. The procedure for calculation of these values is given in [1] together with error estimates. Notice that values of  $\xi_i$  are skew-symmetric with respect to  $\xi = 0$  while values of  $w_i$  are symmetric.

## 3 Two- and Three-Dimensional Gauss Quadrature for Rectangles and Rectangular Hexahedra

The equivalent two- and three-dimensional formulae are obtained by simple combinations of (1), *i.e.*

$n$	$\pm \xi_i$	$w_i$
2	0.57735 02691 89626	1.00000 00000 00000
3	0.00000 00000 00000 0.77459 66692 41483	0.88888 88888 88888 0.55555 55555 55555
4	0.33998 10435 84856 0.86113 63115 94053	0.65214 51548 62546 0.34785 48451 37454
5	0.00000 00000 00000 0.53846 93101 05683 0.90617 98459 38664	0.56888 88888 88888 0.47862 86704 99366 0.23692 68850 56189
6	0.23861 91860 83197 0.66120 93864 66265 0.93246 95142 03152	0.46791 39345 72691 0.36076 15730 48139 0.17132 44923 79170
7	0.00000 00000 00000 0.40584 51513 77397 0.74153 11855 99394 0.94910 79123 42759	0.41795 91836 73469 0.38183 00505 05119 0.27970 53914 89277 0.12948 49661 68870
8	0.18343 46424 95650 0.52553 24099 16329 0.79666 64774 13627 0.96028 98564 97536	0.36268 37833 78362 0.31370 66458 77887 0.22238 10344 53374 0.10122 85362 90376
9	0.00000 00000 00000 0.32425 34234 03809 0.61337 14327 00590 0.83603 11073 26636 0.96816 02395 07626	0.33023 93550 01260 0.31234 70770 40003 0.26061 06964 02935 0.18064 81606 94857 0.08127 43883 61574
10	0.14887 43389 81631 0.43339 53941 29247 0.67940 95682 99024 0.86506 33666 88985 0.97390 65285 17172	0.29552 42247 14753 0.26926 67193 09996 0.21908 63625 15982 0.14945 13491 50581 0.06667 13443 08688

Table 1: Regular Gauss Points and Weight Factors

$$I = \int_{-1}^{+1} \int_{-1}^{+1} f(\xi, \eta) d\xi d\eta = \sum_{i=1}^n \sum_{j=1}^n f(\xi_i, \eta_j) w_i w_j \quad (2)$$

and

$$I = \int_{-1}^{+1} \int_{-1}^{+1} \int_{-1}^{+1} f(\xi, \eta, \zeta) d\xi d\eta d\zeta = \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n f(\xi_i, \eta_j, \zeta_k) w_i w_j w_k \quad (3)$$

The same integration points and weight factors as given in table 1 are used in each coordinate direction.

The two-dimensional formula is used for three-dimensional boundary element analysis and for internal cells in two-dimensional analysis. Equation (3) may be employed for internal cells in three-dimensional analysis.

## 4 Two-Dimensional Quadrature for Triangular Domains

Numerical integration over a triangle can be carried out using triangular coordinates as shown in figure 1, through the equation

$$I = \int_0^1 \left[ \int_0^{1-\xi_2} f(\xi_1, \xi_2, \xi_3) d\xi_1 \right] d\xi_2 = \sum_{i=1}^n f(\xi_1^i, \xi_2^i, \xi_3^i) w_i \quad (4)$$

where  $n$  is the number of integration points,  $\xi_1^i$ ,  $\xi_2^i$  and  $\xi_3^i$  are the triangular coordinates of the integration point  $i$  and  $w_i$  is the associated weight factor. Values of  $\xi_1^i$ ,  $\xi_2^i$ ,  $\xi_3^i$  and  $w_i$  from reference [2] are given in table 2.

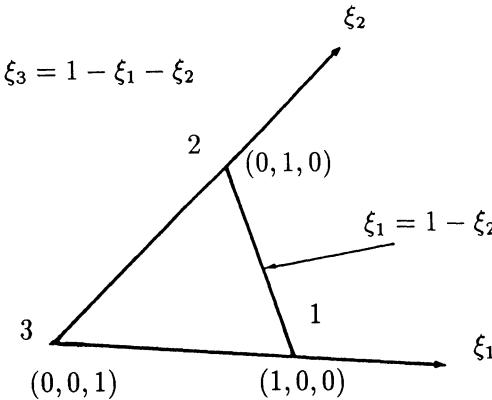


Figure 1: Triangular Coordinates

Equation (4) is used for triangular boundary elements in three-dimensional analysis and for triangular internal cells in two-dimensional analysis. Similar formulae may be constructed for tetrahedra and pentahedra as used for three-dimensional internal cells.

$n$	$i$	$\xi_1^i$	$\xi_2^i$	$\xi_3^i$	$w_i$
1 (linear)	1	1/3	1/3	1/3	1
2 (quadratic)	1	1/2	1/2	0	1/3
	2	0	1/2	1/2	1/3
	3	1/2	0	1/2	1/3
4 (cubic)	1	1/3	1/3	1/3	-9/16
	2	3/5	1/5	1/5	25/48
	3	1/5	3/5	1/5	25/48
	4	1/5	1/5	3/5	25/48
7 (quintic)	1	0.33333333	0.33333333	0.33333333	0.22500000
	2	0.79742699	0.10128651	0.10128651	0.12593918
	3	0.10128651	0.79742699	0.10128651	0.12593918
	4	0.10128651	0.10128651	0.79742699	0.12593918
	5	0.05971587	0.47014206	0.47014206	0.13239415
	6	0.47014206	0.05971587	0.47014206	0.13239415
	7	0.47014206	0.47014206	0.05971587	0.13239415

Table 2: Modified Gauss Points and Weight Factors for Triangles

## References

1. Stroud, A. H. and Secrest, D.  
Gaussian Quadrature Formulas, Prentice-Hall, New York, 1966.
2. Hammer, P. C., Marlowe, O. J. and Stroud, A. H.  
Numerical Integration over Simplexes and Cones, *Mathematics of Computation*, Vol. 10, pp 130-139, 1956.

# Appendix 2

## Numerical Integration: Singular Integrals

### 1 Introduction

As the fundamental solutions used in BEM analysis contain a singularity, the simple Gauss quadrature formulae given in Appendix 1 cannot be used when integration is carried out over an element which contains the source point. In simple cases, for example one-dimensional constant and linear elements, the relevant values may be obtained by analytical integration, see equation (2.39) and section 2.4.3, respectively. Analytical integration may also be used for constant triangular boundary elements in three-dimensional analysis [1], and with some difficulty for higher-order elements as long as curved geometry is not employed.

It is usually more convenient, however, to use numerical integration techniques, and this Appendix describes quadrature schemes appropriate to singular integrals.

### 2 Logarithmic Singularities

A modified Gauss quadrature appropriate to integrals with logarithmic singularity is of the form

$$I = \int_0^1 \ln\left(\frac{1}{\xi}\right) f(\xi) d\xi = \sum_{i=1}^n f(\xi_i) w_i \quad (1)$$

Integration points and weight factors taken from [2] are given in table 1, for  $n$  varying from 2 to 7.

The use of this procedure is limited to two-dimensional analysis and implies carrying out a specific coordinate transformation for each term to be integrated to reduce it to the form given in equation (1).

### 3 The Telles' Transformation

As a simple alternative, valid for both two- and three-dimensional analysis, and which uses standard Gauss points and weight factors, Telles [3] has proposed a transformation, the objective of which is to cancel the singularity by forcing its Jacobian to be zero at the singular point. Assume the integral to be evaluated is

$n$	$\xi_i$	$w_i$
2	0.11200880	0.71853931
	0.60227691	0.28146068
3	0.06389079	0.51340455
	0.36899706	0.39198004
4	0.76688030	0.09461540
	0.04144848	0.38346406
5	0.24527491	0.38687532
	0.55616545	0.19043513
6	0.84898239	0.03922548
	0.02913447	0.29789346
7	0.17397721	0.34977622
	0.41170251	0.23448829
8	0.67731417	0.09893046
	0.89477136	0.01891155
9	0.02163400	0.23876366
	0.12958339	0.30828657
10	0.31402045	0.24531742
	0.53865721	0.14200875
11	0.75691533	0.05545462
	0.92266884	0.01016895
12	0.01671935	0.19616938
	0.10018568	0.27030264
13	0.24629424	0.23968187
	0.43346349	0.16577577
14	0.63235098	0.08894322
	0.81111862	0.03319430
15	0.94084816	0.00593278

Table 1: Integration Table for Logarithmic Singularity

$$I = \int_{-1}^{+1} f(\xi) d\xi \quad (2)$$

in which  $f(\xi)$  is singular at point  $\bar{\xi}$ ,  $-1 \leq \bar{\xi} \leq 1$ . If a second-order transformation

$$\xi(\gamma) = a\gamma^2 + b\gamma + c \quad (3)$$

is employed such that

$$\left. \frac{d\xi}{d\gamma} \right|_{\bar{\xi}} = 0 \quad (4)$$

$$\xi(+1) = +1$$

$$\xi(-1) = -1$$

then the constants in (3) can be calculated in the form

$$a = -c \quad (5)$$

$$b = 1$$

$$c = \frac{\bar{\xi} \pm \sqrt{\bar{\xi}^2 - 1}}{2}$$

The last two conditions in (4) are imposed in order to maintain the same integration limits as in standard Gauss quadrature. The first condition imposes a zero Jacobian at  $\bar{\xi}$ . In this case, however, it is necessary that  $|\bar{\xi}| \geq 1$  in order to avoid complex roots in the evaluation of  $c$ . Thus, the use of the quadratic expansion (3) is limited to cases where the singularity is at the extreme points, as for linear elements.

A general transformation, valid for any position of the singularity, can be obtained by assuming a third-order expansion for  $\xi$ , i.e.

$$\xi = a\gamma^3 + b\gamma^2 + c\gamma + d \quad (6)$$

In this case, in addition to the conditions (4) a further requirement is necessary:

$$\left. \frac{d^2\xi}{d\gamma^2} \right|_{\bar{\xi}} = 0 \quad (7)$$

which implies that the Jacobian of the transformation has a minimum at  $\bar{\xi}$ .

The constants are now given by

$$a = \frac{1}{Q} \quad (8)$$

$$b = -\frac{3\bar{\gamma}}{Q}$$

$$c = \frac{3\bar{\gamma}^2}{Q}$$

$$d = -b$$

where  $Q = 1 + 3\bar{\gamma}^2$  and  $\bar{\gamma}$  is simply the value of  $\gamma$  which satisfies  $\xi(\bar{\gamma}) = \bar{\xi}$ , i.e.

$$\bar{\gamma} = \sqrt[3]{\bar{\xi}\bar{\xi}^* + |\xi^*|} + \sqrt[3]{\bar{\xi}\bar{\xi}^* - |\xi^*|} + \bar{\xi} \quad (9)$$

where  $\xi^* = \bar{\xi}^2 - 1$ .

After the transformation, standard Gauss integration points and weight factors (Appendix 1) are used. An interesting feature of the above transformations is that a large concentration of points is automatically obtained near the singularity.

## References

1. Brebbia, C. A., Telles, J. F. C. and Wrobel, L. C.  
Boundary Element Techniques, Springer-Verlag, Berlin and New York, 1984.
2. Stroud, A. H. and Secrest, D.  
Gaussian Quadrature Formulas, Prentice-Hall, New York, 1966.
3. Telles, J. C. F.  
A Self-Adaptive Coordinate Transformation for Efficient Numerical Evaluation of General Boundary Element Integrals, *International Journal for Numerical Methods in Engineering*, Vol. 24, pp 959-973, 1989.

# Index

- Accelerations 201, 205, 235, 250  
Acoustics 119, 130  
Activation energy 135  
Analytical integration 4, 22, 70  
Angle of twist 92  
Approximating functions 71, 75, 77, 92,  
    112, 142, 230  
Arbitrary  
    constant 122, 123  
    geometry 28  
Artificial  
    boundary 204, 205  
    damping 202  
Assembled equations 26  
Axisymmetric diffusion 198  
  
Backward differences 202  
Benchmark 98  
Berger's equation 223, 244, 245  
Bessel functions 256  
Betti's reciprocal theorem 13  
Bi-harmonic equation 44, 238, 245  
Biot number 214  
Body forces 35, 223, 224, 226, 228, 230,  
    249  
Boundary  
    conditions 12, 13, 15, 17, 18, 19, 21,  
        26, 40, 41, 43, 46, 48, 51, 54, 55,  
        74, 78, 93, 100, 101, 106, 107, 111,  
        112, 122, 126, 134, 139, 226, 228  
    of the third kind 139, 141, 143, 145  
fluxes 16, 131, 207  
integral 17, 21, 24, 35, 43, 44, 45, 70,  
    72, 141  
    equations 12, 13, 15, 16, 18, 45, 69  
nodes 26, 36, 46, 51, 55, 71, 73, 74, 78,  
    79, 80, 89, 93, 102, 105, 125, 126  
Branches 136  
Burger's equation 5, 34, 132, 134  
  
Cartesian  
    system 29  
    tensor notation 224  
Cauchy principal value 18  
Cell integration 12, 35, 93, 94  
Centrifugal load 252  
Central differences 202  
Coefficient  $c$  27  
Collocation 11, 73, 74, 77, 81, 82, 107  
Collision number 135  
Compatibility conditions 105  
Completeness 77  
Computer program  
    1 47  
    2 83  
    3 146  
    4 177  
Concentrated  
    load 242  
    source 15, 37  
Concentration 206  
Condensation 130  
Conductivity 139, 140, 212  
Connectivity 166  
Constant elements 19, 20, 22, 167, 196,  
    199, 242  
Convection-diffusion 5, 132, 175, 206, 207,  
    255, 259  
Convective problems 76, 112, 113, 143, 169  
Convergence 46, 47, 133, 134, 136, 171,  
    197, 215, 216  
Corner points 23, 25, 26, 27, 40, 99, 195,  
    207, 259  
Coupled problem 111  
Crank-Nicolson scheme 197  
Critical value 135, 215, 216  
Cubic elements 12, 31, 33  
Curved elements 19, 22, 28, 32, 83, 102,  
    138

Cutouts 196

Damping 202, 204

Density 135, 205

Derivatives 112, 113, 141  
second 141, 142, 146

Diagonal  
matrix 133, 143  
terms 27, 28, 55

Diffusion equation 175, 176

Dirac delta function 15, 37, 226, 240

Direct formulation 2, 11

Direction cosines 76, 93

Discontinuity 18, 25, 55, 79

Discontinuous elements 27, 40, 195

Dispersion 206  
coefficient 206

Displacements 121, 226

Distance function 75, 81

Distributed sources 98, 106, 107

Domain 12, 13, 15, 16, 27, 36, 37, 39, 57,  
71, 72, 104, 106  
discretization 186  
integral 12, 35, 36, 41, 43, 44, 45, 46,  
47, 57, 69, 72, 102, 112, 116, 139,  
230, 241, 257  
sources 12, 116  
terms 34, 35, 37, 40

Duhamel's integral 236

Dynamic analysis 232

Eccentric load 244

Eigenvalue/Eigenvector 118, 120, 121, 122,  
130, 234, 235  
complex 120, 130

Einstein's convention 15

Elastic foundation 223

Equilibrium  
conditions 105, 224  
equation 224

Elasticity 223, 224, 246

Elastodynamics 223

Errors 14, 39

Essential boundary conditions 13, 43, 115,  
167

Field point 73, 82

Finite differences 1, 2, 12, 113, 202

Finite elements 1, 2, 12, 21, 30, 69, 71,  
113, 130, 138 176, 185, 187, 199,  
202, 212, 236

Forced vibrations 235

Fourier expansion 4, 70

Frank-Kamenetskii 135

Free  
surface 204  
term 18  
vibrations 234, 236

Functional 196, 197

Fundamental solution 12, 15, 16, 17, 20,  
21, 22, 35, 37, 41, 44, 45, 47, 69,  
72, 165, 185, 198, 207 223, 227,  
240, 245, 246, 256, 259

Galerkin  
scheme 196  
vector 4, 12, 43, 45, 70, 232

Gauss  
elimination 58, 77, 79, 88, 152, 216,  
229, 248, 258  
quadrature 22, 35, 47

Gravitational load 250

Green's  
theorem 12  
third identity 13, 44, 45

Harbour resonance 117

Harmonic functions 43

Heat  
capacity matrix 176, 202, 215  
conduction/diffusion 135, 169  
of decomposition 135  
sources 139

Heaviside load 202, 203

Helmholtz equation 5, 45, 74, 118, 132, 233

Hemisphere 17

Homogeneous  
boundary conditions 34, 37, 43, 74, 99,  
113, 122, 123  
coordinates 24, 29  
equations 41, 43, 70

Hooke's law 225, 249

Houbolt scheme 202, 235

Householder algorithm 234

Identity matrix 80, 111

- Ignition
  - point 218
  - temperature 135, 136, 216
- Impedance tube 130
- Indicial notation 224
- Indirect formulation 2, 11
- Infinite regions 27, 199, 204
- Influence
  - coefficients 11, 12, 20
  - matrices 46
- Initial
  - conditions 176, 177, 186
  - displacements 201
  - states 35
  - strain 253
  - velocities 201
- Integral equations 11, 12, 13, 15, 16, 18, 24, 35, 46, 69, 72
- Integration
  - by parts 12, 14, 15, 16, 35, 41, 44, 72, 226, 245
  - two-level 177, 196, 197, 215
- Interface 105, 106
- Internal
  - angle 24, 257
  - boundaries 106, 200
  - cells 12, 13, 35, 36, 37, 43, 47, 48, 51, 57, 69, 70, 74, 83, 100, 113, 185
  - degrees of freedom 186
  - derivatives 116
  - fluxes 21
  - nodes/points 21, 34, 36, 37, 48, 51, 59, 70, 71, 73, 75, 79, 80, 81, 93, 94, 102, 105, 112, 116, 123, 125, 127, 138, 139, 145, 168, 186, 199, 216, 242, 249
  - points 195
  - regions 106
  - values 21
- Interpolation functions 24, 29, 31, 32, 71, 73, 165, 228
- Isoparametric element 30
- Isotropic medium/material 15, 40, 135, 216
- Iterations 34, 110, 133, 136, 137, 142, 144, 209, 216
- Jacobian 12, 30, 32, 211
- Jump 18
- Kelvin's fundamental solution 5, 223, 227
- Kirchhoff
  - transformation 131, 139, 140, 144, 209, 213
  - plate theory 5, 223
- Kronecker's delta 20, 77, 224
- Laplace
  - equation 4, 5, 12, 13, 16, 18, 33, 34, 35, 43, 45, 47, 51, 60, 70, 72, 106, 133, 140, 165, 175, 198, 257
  - operator 14, 16, 47, 72, 110, 223, 245
- Linear elements 12, 13, 19, 23, 24, 25, 27, 30, 34, 40, 47, 93, 98, 102, 112, 146, 177, 185, 216
- Localized particular solutions 43
- Mass
  - density 232
  - matrix 120, 202, 233
- Mixed
  - boundary conditions 195
  - formulation 21
- Modal superposition 235
- Monte Carlo method 12, 37, 38, 40, 41, 214
- Multiple Reciprocity method 4, 12, 45, 70
- Natural
  - boundary conditions 13
  - frequency 118, 121, 122, 234
  - modes 234, 236
- Navier equation 227
- Navier Stokes equations 132
- Newmark scheme 202
- Newton-Raphson scheme 210, 214
- Non-homogeneous 141, 215
- Non-inversion DRM 130
- Non-linear
  - boundary conditions 131, 175, 209, 212
  - material 131, 139, 175, 209
  - problems/cases 70, 131, 207
  - sources 175, 209
  - terms 35, 135, 136, 170
  - transient problems 175
- Normal derivative 14, 17, 21, 46, 75

Numerical integration 12, 22, 30, 35, 47, 165, 166  
Oscillations 113, 187, 206  
Outward normal 13, 55, 72, 76, 79, 89, 93, 240  
  
Particular  
  integrals 12  
  solution 5, 12, 41, 42, 43, 70, 75, 78, 112, 115, 134, 167, 170, 230, 248  
Pascal's triangle 75  
Péclet number 255  
Phase change 132, 209  
Plane stress/strain 227  
Plate bending 223, 238  
Poisson  
  equation 4, 5, 12, 15, 35, 37, 38, 39, 40, 41, 43, 45, 47, 61, 63, 70, 74, 75, 76, 77, 81, 83, 92, 98, 109, 223  
  ratio 225  
Projections 81  
  
Q-R algorithm 234  
Quadratic elements 12, 19, 28, 29, 30, 102, 103, 104, 138, 185, 250  
  
Radiation 209, 213  
Radiative interchange factor 213  
Random points 12, 38, 39, 40  
Reaction coefficient 256  
Reactive solid 135  
Rectangular matrix 26  
Relaxation 171  
Rigid body 249  
  
Saint Venant 92  
Schematized equations 78, 81, 110  
Semicircle 17  
Shear modulus 92, 225  
Singular matrix 77, 99  
Singularity 18, 22, 30, 46, 134, 171  
Smooth surfaces 17, 20, 24, 26, 27  
Somigliana's identity 227  
Sommerfeld radiation condition 204  
Source node/point 72, 82, 106, 107, 114  
Space derivatives 112, 113  
Specific heat 135  
  
Sphere 16  
Spheroid  
  prolate 199  
Spherical coordinates 16  
Spontaneous ignition 5, 134, 135, 136, 138, 215  
Steady-state 132, 134, 209, 215  
Stefan-Boltzman constant 213  
Stiffness matrix 120  
Strain  
  -displacement relationship 225  
  tensor 225, 231  
Stress  
  function 92  
  tensor 224, 231  
Subregions 102, 104, 106, 214  
Subspace iteration method 234  
Superelement 71  
Surface forces 224  
Symmetric matrix 79  
Symmetry 16, 22, 40, 93, 115, 168, 170, 171, 185, 196, 199, 212  
  
Tangent matrix 211  
Tangential fluxes 33  
Temperature 40, 98, 135, 139, 140, 195, 196, 199, 200, 201, 212 216, 254  
Thermal  
  conductivity 40, 135, 209  
  diffusivity 135, 196  
  load 253  
  shock 185, 187, 195  
Three-dimensional analysis 5, 165, 246  
Time  
  dependent 35, 71, 74, 83, 110, 111, 136, 139, 175  
  derivative 176  
  integration 176, 202  
  step 177, 186, 187, 197, 207, 212, 215, 258  
Torsion 92, 93, 94  
Tractions 224, 226, 231, 248  
Transformations 12, 22, 28, 44  
Transport 206  
Transpose 82  
Trigonometric series 75  
Trivial solution 112, 122

Unique values 25, 26, 27

Unit source 15, 27

Universal gas constant 135

Upwind 113

Variance 39

Velocity 206, 207, 255, 259

Vibrating beam 117, 119

Virtual work 13

Wave

celerity 201

equation 201, 204

propagation 5, 175, 201

Weighted residuals 12, 13, 35, 72, 226

Weighting function 14, 15

Winkler's springs 238

Young's modulus 225

## THE AUTHORS

**PAUL WILLIAM PARTRIDGE** was born in West Bromwich, U.K. and obtained his BSc in Civil Engineering from the University of Southampton in 1972. He carried out research in finite element analysis of shallow water problems under the supervision of Dr. C.A. Brebbia, leading to a PhD in 1976 at the same university. He worked as a Lecturer for the Post-Graduate Course in Civil Engineering at the Federal University of Rio Grande do Sul, Brazil, where he supervised the university's first MSc thesis on boundary elements in 1980. He transferred to the University of Brasilia in 1985, where he was head of the Civil Engineering Department from 1986 to 1988. The work for this book on the Dual Reciprocity Method was started during his Post-Doctoral research at the Wessex Institute of Technology during the period 1989 to 1990. He is currently Reader in Advanced Computing at the same institute.

Dr. P.W. Partridge is the author and co-author of over forty scientific papers on the use of numerical methods in engineering, including several book chapters, however this is his first venture into writing a complete book.

**CARLOS ALBERTO BREBBIA** was born in Argentina of Italian origin and studied there at the University of Litoral, where he obtained a degree in Civil Engineering. He carried out research work at the University of Southampton and the Massachusetts Institute of Technology, obtaining a PhD degree from the former in 1968. His academic career started as a Lecturer at Southampton University, where he later on became a Reader in Computational Engineering and was appointed Professor of Civil Engineering at the University of Irvine, California. In 1981 he resigned his Chair in California to set up the Wessex Institute of Technology in the U.K., of which he is now the Director.

Dr C.A. Brebbia has published numerous papers and is editor or co-editor of numerous books, many of them dealing with advanced engineering topics and in particular developments in boundary elements. He is also the author and co-author of 11 books including some well known texts on boundary elements. Dr Brebbia is editor of the *International Journal of Engineering Analysis with Boundary Elements* and amongst other activities has contributed to organising many conferences and seminars in this field.

**LUIZ CARLOS WROBEL** was born in Brazil and obtained both his BSc and MSc in Civil Engineering there, respectively from the Fluminense Federal University in 1974 and COPPE/Federal University of Rio de Janeiro in 1977. He was one of the first researchers working with boundary elements at the University of Southampton, obtaining a PhD degree there in 1981. He worked as a lecturer for the Post-Graduate Centre in Engineering at the Federal University of Rio de Janeiro (COPPE/UFRJ)

during the period 1981 to 1989, and is at present Reader in Computational Mechanics at the Wessex Institute of Technology.

Dr. L.C. Wrobel has published over 120 papers in scientific journals and conference proceedings, on the field of numerical methods in engineering. He is co-author of the book *Boundary Element Techniques*, published by Springer-Verlag in 1984, and co-editor of three others. Dr. L.C. Wrobel is a member of the editorial board of the journals *Engineering Analysis with Boundary Elements*, and *International Journal for Numerical Methods in Heat and Fluid Flow*; he also acts as referee for several other scientific journals.