The Laplace transform boundary element method for diffusion-type problems

Diane Crann

A thesis submitted in partial fulfilment of the requirements of the University of Hertfordshire for the degree of

Doctor of Philosophy

The programme of research was carried out in the
Faculty of Engineering and Information Sciences
University of Hertfordshire

May 2005

When this you see remember me
And bear me in your mind;
And be not like the weathercock
That turn att evry wind.
When I am dead and laid in grav
And all my bones are rotten,
By this may I remembered be
When I should be forgotten.

Anon.
(Cross stitch sampler 1736)

Acknowledgements

This thesis has been a long time coming and I'm sure many people thought it would never arrive. However I was determined; this is my hobby and, for me, mathematics is fun and enjoyable to do.

When I first started doing research at the University, I'd already been to a number of international boundary element conferences, organised by Professor Carlos Brebbia, as a 'partner' and the words base node, target element, singular integral, inherent parallelism, were part of my everyday mathematical language. I typed BEM papers and a thesis and the language became very familiar. I wanted to be part of this community, understand more and be accepted as a mathematician, not just a mathematician's partner.

My friend and fellow student in BEM research has been Linda Radford and we have supported each other through the ups and downs of our day-to-day lives while 'doing our homework'. We've shared notes, compared results and she has kept me going when things haven't been straight-forward. She's been so supportive and I hope I can now help her towards finishing her own research.

Many people in the University's research community have also been very supportive. Professor Bruce Christianson, my first supervisor, has been very encouraging and given me the appropriate confidence when necessary. I hope I've 'blown my own trumpet' as much as he wanted me to and I owe him many thanks for his continued support. I'd also like to thank Dr Mike Bartholomew-Biggs for his support and advice through these last final months making me realise I really can 'speak mathematically'.

Early on I had enormous help, teaching and advice from Dr Steve Brown, who was then in the Computer Science Department, and he always said at some time I would know my '15 minutes of fame' and I think my work on

AD is this for me.

Dr Jawaid Mushtaq was instrumental in the parallel computation work and I thank him for all his help with the different architectures we had available. I'll always remember the difference between a mathematician and an engineer with numerical computation; the first thing the engineer does is to take the back off the computer.

Dr Wattana Toutip and Dr Mick Honnor came to the Department and continued with parallel BEM and dual reciprocity work. I've been able to follow on from Wattana's work and I thank them both for their helpful comments.

Many other people, family, friends and colleagues, have contributed with help, advice and encouragement and I thank them for their continued kindness.

My oral examination was surprisingly enjoyable due to the thoroughly professional yet friendly approach of my examiners Professor Ferri Aliabadi and Dr Steve Kane. Their comments were extremely helpful and have confirmed to me that my ideas for future work are definitely worth continuing. I really don't want to stop now.

Finally, and most importantly, I have to thank Professor Alan Davies for seeing me through this research as I know I have been a trial to his amazing patience on many occasions. I hope we have many years together continuing to develop mathematical ideas and being able to see the world while attending mathematical conferences and renewing other friendships.

Abstract

Diffusion-type problems are described by parabolic partial differential equations; they are defined on a domain involving both time and space. The usual method of solution is to use a finite difference time-stepping process which leads to an elliptic equation in the space variable. The major drawback with the finite difference method in time is the possibility of severe stability restrictions.

An alternative process is to use the Laplace transform. The transformed problem can be solved using a suitable partial differential equation solver and the solution is transformed back into the time domain using a suitable inversion process. In all practical situations a numerical inversion is required. For problems with discontinuous or periodic boundary conditions, the numerical inversion is not straightforward and we show how to overcome these difficulties.

The boundary element method is a well-established technique for solving elliptic problems. One of the procedures required is the evaluation of singular integrals which arise in the solution process and a new formulation is developed to handle these integrals.

For the solution of non-homogeneous equations an additional technique is required and the dual reciprocity method used in conjunction with the boundary element method provides a way forward.

The Laplace transform is a linear operator and as such cannot handle non-linear terms. We address this problem by a linearisation process together with a suitable iterative scheme. We apply such a procedure to a non-linear coupled electromagnetic heating problem with electrical and thermal properties exhibiting temperature dependencies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In this chapter we give an overview of the programme of research associated with the Laplace transform boundary element method (LTBEM). We provide a background to the work and explain how the thesis is set out. Firstly, however, we state the objectives which prompted this particular work and followed on from research already undertaken.

Our objectives at the beginning of this research work were:

1. To investigate the LTBEM for accuracy when considering numerical inversion methods,

2. To investigate the LTBEM for accuracy when considering non-monotonic boundary conditions,

3. To investigate the LTBEM on a distributed memory architecture for efficiency of computation.

1.2 Background of the research

Eight years ago when this work began the ideas of the research team were centred upon investigating the boundary element method and the solution to problems using a distributed memory architecture. Four transputers were available, configured in parallel, then the work was transferred to a network of SUN workstations using the PVM message passing protocol and finally the university acquired an *nCube* parallel machine. The Laplace transform method was considered for reducing a parabolic problem to either Laplace's equation or the modified Helmholtz problem and a variety of different elliptic solvers were used before inverting back into the time space, the ideas which form the basis of this thesis.

However, with the university losing the *nCube* and pc's themselves having a much larger memory than before, parallelisation wasn't such a priority and the work took a different direction to investigate the evaluation of singular integrals within the boundary element method. Working with members of the Computer Science Department, Automatic Differentiation (AD) was considered and a program was developed using Taylor polynomial coefficients to evaluate the singular integrals involved with quadratic elements along similar lines to AD. Although the method worked well and accuracy on test problems was very encouraging, the efficiency of the method was not as favourable as other methods in use and it was decided to concentrate on linear elements in the boundary element method and use code for implementation which was already available.

Inversion techniques for the Laplace transform were investigated and a real-variable inversion method was chosen which worked well, gave accurate results and was easy to implement There were two problems that were acknowledged with the method, namely inversions of transforms associated with discontinuous and periodic functions. Numerical techniques were used to recover the solutions and very good results were obtained. The method

was very satisfactory, it was robust and accurate, and in order to move on a further refinement was needed to handle the non-homogeneous problems so the dual reciprocity method was included. Following testing on a number of examples we found that this refined method gave accurate results leading us to consider non-linear initial boundary-value problems.

In the following chapters, this story becomes clear as we move forward through the thesis.

A number of papers have been published throughout the period of this research programme highlighting the contribution to knowledge within this area of work. We refer to them where appropriate in the thesis.

A significant number of numerical computations have been developed but only certain selected results have been included in the thesis. A complete set of results can be found in the technical report by Crann (2005).

1.3 Development of the thesis

In Chapter 2 we give a general classification of partial differential equations and explain the significance of given boundary and/or initial conditions. We discuss various methods for finding the solution of such equations and comment on the advantages and disadvantages of using each of the methods.

In Chapter 3 we describe in further detail the background and numerical implementation of the boundary element method (BEM) and we consider in Chapter 4 the problems associated with the evaluation of the integrals which occur in the BEM. We formulate a new method for dealing with these integrals and show that in terms of accuracy it compares well with alternative methods.

The Laplace transform method is shown to be very convenient when used in conjunction with other solution processes for solving parabolic problems. The difficulty associated with using the Laplace transform manifests itself in the inversion which is required after the transformed equation has been

solved in the Laplace space. In Chapter 5 we consider two real-variable methods of inverting the Laplace transform which we test on a variety of transforms. In Chapter 6 we then use the Laplace transform method with our preferred inversion process to solve parabolic problems. We use a variety of methods both sequentially and in parallel to demonstrate the versatility of the Laplace transform approach. We concentrate on the Laplace transform boundary element method in the remainder of this thesis.

We extend the LTBEM in Chapter 7 to accommodate non-homogeneous problems using the dual reciprocity method and demonstrate the combined method with a number of linear problems.

The standard form of the LTBEM is not suitable for problems with non-monotonic time-dependent boundary conditions due to the inversion processes which smooth out the discontinuities or oscillations. In Chapter 8, we show that using the Laplace transform method in a piecewise manner we can find the solution with good accuracy within the neighbourhood of a discontinuity or predict the oscillatory nature of the solution.

For our final numerical work, in Chapter 9, we demonstrate that non-linear problems can be solved using the LTBEM with dual reciprocity using linearisation and iterative schemes to handle the non-linearities. We solve a variety of non-linear problems and consider a coupled non-linear problem which we solve by our method and report very good results.

In our final chapter we summarise the contribution made in this thesis and bring together our ideas on the significance of the work and the areas for future research which it has opened. We also list the published work which has arisen from this research and a brief explanation of the topic and where in the thesis it is presented.

Chapter 2

Initial boundary-value problems

2.1 Introduction

Many problems in physical science and engineering are modelled mathematically by differential equations. Examples can be found in the classical texts in areas such as fluid mechanics (Lamb 1932, Dryden *et al.* 1956), heat transfer (Jakob 1949, Carslaw and Jaeger 1959), elasticity (Love 1927, Sokolnikoff 1956), diffusion (Crank 1975) and electromagnetic field problems (Stratton 1941). Most practical problems involve more than one independent variable and so are modelled by partial differential equations. More recently such equations have been developed to model situations in biological science (Edelstein-Keshet 1988) and in finance (Wilmott *et al.* 1995).

For the mathematical models of these physical problems to have a unique solution, boundary conditions and initial conditions are necessary. If the number of conditions is sufficient to determine a unique solution that depends continuously on the data, then the problem is said to be *well-posed* or *properly-posed* (Renardy and Rogers 1993). Continuity of the solution may also be interpreted as small changes in data yield small changes in the

solution.

2.1.1 Classification of partial differential equations

We can classify partial differential equations in three ways as follows (Williams 1980):

1. *Elliptic equations* are associated with steady-state problems and require conditions posed on a closed boundary. Changes in the boundary data are felt throughout the domain instantaneously, *i.e.* these equations are not associated with propagation problems.

   Typical examples of elliptic equations are Laplace's equation

   $$\nabla^2 u = 0$$

   and Poisson's equation

   $$\nabla^2 u = f \tag{2.1}$$

   where f is a known function of position (x, y).

2. *Hyperbolic equations* are often associated with time-dependent problems and the solution is obtained starting from some given initial condition, propagating through waves of finite speed. The solution at any point in the domain depends only on a finite subset of the initial data, the so-called domain of dependence.

   A typical equation is the wave equation

   $$\frac{\partial^2 u}{\partial x^2} = \frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} \tag{2.2}$$

3. *Parabolic equations* are also associated with time-dependent problems starting from an initial condition. However, the solution at any point depends on the complete set of initial data. They are similar to elliptic equations in that changes in the boundary data are propagated at

infinite speed. A typical example is the diffusion or heat condution equation

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{\alpha}\frac{\partial u}{\partial t} \tag{2.3}$$

An equation is *linear* when the dependent variable and all its partial derivatives occur as single entities *e.g.*

$$a(x,y)\frac{\partial^2 u}{\partial y^2} + b(x,y)\frac{\partial u}{\partial x} + c(x,y)u = g(x,y)$$

otherwise the equation is *non-linear e.g.*

$$a(x,y,u)\frac{\partial^2 u}{\partial y^2} + b(x,y,u)\frac{\partial u}{\partial x} + c(x,y,u)u = g(x,y,u) \tag{2.4}$$

where at least one of a, b, c or g is an explicit function of u.

This is particularly important in Chapter 5 where we introduce the Laplace transform since the transform is applicable only in the case of linear equations. For non-linear problems, in Chapter 9, we shall seek a suitable linearisation procedure.

If $g(x,y,u) \equiv 0$ in equation (2.4), then the equation is said to be *homogeneous.*

2.1.2 Boundary and initial conditions

Initial boundary value problems comprise a partial differential equation defined in some region D together with specified conditions on the boundary C and given values in D at some starting time.

The three most commonly occuring types of *boundary condition* associated with partial differential equations are:

1. *Dirichlet condition*, where the value of the dependent variable on the boundary is given,

2. *Neumann condition*, where the first-order space derivative of the dependent variable on the boundary in a direction normal to the boundary is given, and

3. *Robin, or mixed condition*, a linear combination of the Dirichlet and Neumann conditions.

The *initial conditions* are the prescribed values of the function and/or its time derivative throughout D at time zero.

Problems which comprise a differential equation together with boundary conditions only are called *boundary-value problems.* Problems which comprise a differential equation together with initial conditions only are called *initial-value* problems. Elliptic partial differential equations are associated with boundary value problems. Hyperbolic and parabolic partial differential equations require both boundary values and initial values and are associated with *initial boundary-value problems.*

We shall call the equation

$$\nabla^2 u = f(x, y, u, u_x, u_y,)$$

where we use the usual notation $u_x = \partial u/\partial x$ *etc.*, with boundary and/or initial conditions a *Poisson-type problem.*

For Poisson-type problems to be well-posed we require that either u or its normal derivative, $\partial u/\partial n$, must be specified at each point on the boundary. In particular the example due to Hadamard (1923) shows that we cannot specify both u and its derivative independently at any point on the boundary. Throughout this thesis whenever we deal with time dependence it will be in the context of well-posed parabolic problems so that we need just one initial condition, *i.e.* we shall specify the initial value, u_0, of u.

In this thesis we shall be looking at a generalisation of the diffusion equation in the form

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} + h\left(x, y, t, u, u_x, u_y\right)$$

We shall call this equation a *diffusion-type equation*; some authors call it the diffusion-reaction equation (Logan 1994).

2.2 Numerical solutions of partial differential equations

Williams (1980) gives an account of some analytical methods of solving linear partial differential equations. The methods either find the solutions from an infinite series of products of functions of the separate independent variables or use integral representations by means of integral transforms, the most common being Laplace or Fourier transforms. The first method can be used only for those relatively simple problems where the independent variables can be separated. Methods using an integral transform require the recovery of the solution using an inversion process which is usually done using standard tables. Again only relatively simple problems are currently amenable to these methods.

The most widely used numerical methods for solving partial differential equations are the Finite Difference Method (FDM) (Smith 1978), the Finite Element Method (FEM) (Davies 1985) and the Boundary Element Method (BEM) (Brebbia and Dominguez 1989). In a recent search on an online bibliographic database Cheng and Cheng (2005) obtained 66,000 entries for the FEM, followed by the FDM with 19,000, BEM with 10,000 and other methods trailed far behind with under 3,500, showing that the FEM has been by far the most popular method for published articles. An indication of the number of annual publications for the BEM seems to be reaching a steady state at about 700-800 papers per year, compared with 5,000 for the FEM and 1,400 for the FDM. The BEM has reached a level of maturity and is well-established as a suitable approach to the solution of partial differential equations.

However, they each have advantages and disadvantages in practical use and a particular method can be chosen to highlight the different aspects of the type of problem in question. The FDM is easy to implement with a good

history of successful applications although for irregular geometry problems can occur with implementation. The FEM is also well-established and is able to give a good representation of all geometries, however unbounded problems require a finite approximation of the boundary at infinity. The BEM has a smaller system matrix due to the reduction in one dimension of the problem compared with the other methods. However solvers used in the FEM are not appropriate. Exterior problems can be handled easily. The method is restricted to those problems for which a fundamental solution is known.

2.2.1 The Finite Difference Method (FDM)

This is the most straightforward method and can be used to solve each type of partial differential equation.

The region is discretised with a grid system, usually rectangular, and the derivatives of the partial differential equations are replaced at each grid point with their corresponding finite-difference representation. Forward, backward or central differences can be used, and the boundary and initial conditions are taken into account during the geometrical set-up.

For Poisson's equation, equation (2.1) which we shall consider in Chapter 7, we use a central difference approximation leading to:

$$\frac{1}{h^2}\left(U_{i-1j} - 2U_{ij} + U_{i+1j}\right) + \frac{1}{k^2}\left(U_{ij-1} - 2U_{ij} + U_{ij+1}\right) = f_{ij}$$

and in the case $h = k$ we have the usual five-point stencil:

$$\frac{1}{h^2}(U_{i-1j} + U_{ij+1} + U_{i+1j} + U_{ij-1} - 4U_{ij}) = f_{ij}$$

Figure 2.1 shows a typical grid mesh for the FDM in which we define h and k. U_{ij} is the approximate value of $u(x, y)$ at the grid point i, j.

For the diffusion equation, equation (2.3) we can use a central difference approximation in space and forward difference in time to obtain the explicit

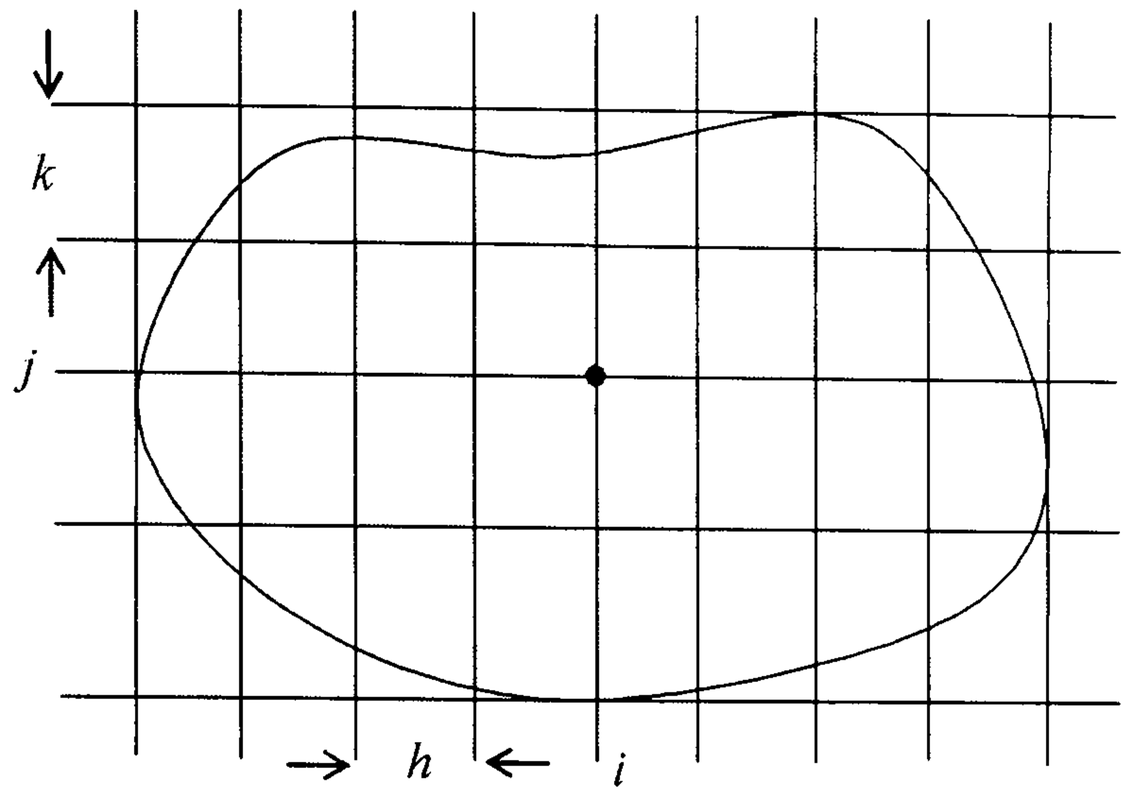

Figure 2.1: A typical grid mesh for the FDM

form

$$U_{ij+1} = U_{ij} + \frac{\alpha k}{h^2}\left(U_{i-1j} - 2U_{ij} + U_{i+1j}\right)$$

In later chapters we shall use the FDM approximation as a comparison for our results.

The finite difference solution is always found at every point on the grid, for every time value, even if only a part of the region's solution is required.

The FDM method is simple and straightforward to use. The rectangular geometry is good for regular boundaries but more complicated geometry is difficult, as is mesh refinement. In principle, accuracy can be improved by reducing the mesh-size, thereby making the grid fit the region better. However, a significant problem associated with FDM is the possibility of numerical instability and care is required to avoid unstable schemes for time-dependent problems.

2.2.2 The Finite Element Method (FEM)

This method is used widely for elliptic problems. Again a grid system is defined over the entire region, however it does not need to be regular. In fact it is often the case that a graded mesh is used to improve accuracy in

specific regions. A typical triangular mesh is shown in Figure 2.2.

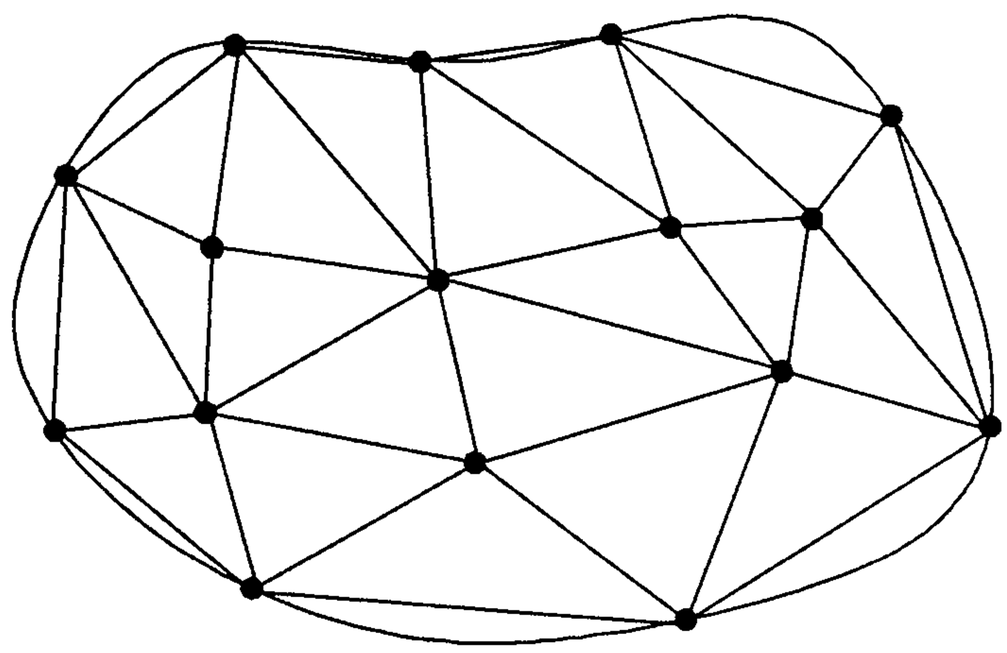

Figure 2.2: A typical grid mesh for the FEM

The triangular mesh fits the boundary of the region geometrically more accurately than a rectangular mesh similar to that of the FDM. Mesh refinement is easily possible. The equation at each node is again described using information from its neighbouring points, using the boundary conditions as necessary. The elements of the system matrix require integrals over each element region and these are performed numerically, usually using Gaussian quadrature. The system matrix is sparse, symmetric and positive definite, allowing very efficient equation solvers to be used. The system matrix may also be banded if the node numbering is appropriate.

The whole grid system is solved and the solution at each point of the mesh is found whether or not it is needed.

There was much innovative work in the early years to improve the efficiency of the solution process *e.g.* isoparametric elements allow even better geometrical approximations by using curved arcs rather than straight lines on the boundary (Irons 1966), the frontal method for finding each solution as the solver works through a banded solution matrix (Irons 1970).

The finite element method has now reached a stage of well-developed maturity. Most practical engineering problems related to solids, structures, fluids, electromagnetism *etc.* are currently solved using a large number of

well-developed FEM packages that are commercially available. Comprehensive details of recent developments can be found in Zienkiewicz and Taylor (2000).

2.2.3 The Boundary Element Method (BEM)

The boundary element method has become the third well-accepted method of solving elliptic equations with a known fundamental solution (Kythe 1996).

The partial differential equation is recast as a boundary integral equation, using the known fundamental solution and relationships such as Green's second theorem, and is solved over the boundary only of the region. In the case of linear elements we have N elements and N nodes see Figure 2.3.

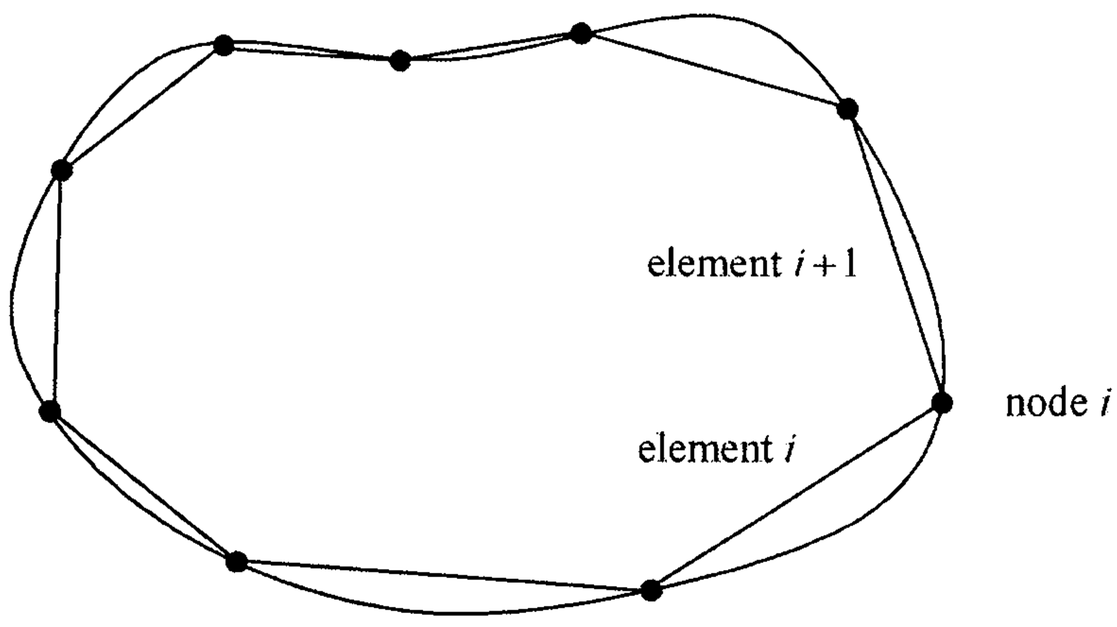

Figure 2.3: A typical grid mesh for the BEM

Interpolation functions are used to describe the geometry over each element, the simplest being constant functions, but more complicated linear, quadratic or high order functions can be used. Again integrals are required over the elements and in general, analytical integration is neither possible nor practical. However it is often the case that the singular integrals, which occur due to the singularities in the fundamental solution, may be evaluated analytically. The non-singular integrals are usually evaluated using Gauss quadrature.

The system matrix is formed by repeating the integration process over each element. The boundary values are applied at every node and values of the function and derivative at all points on the boundary are found by solving the system equations. Values at the internal points may then be found using the solution on the boundary.

The advantages of the BEM are that fewer nodes are used than in the FDM or FEM, as only the boundary is discretised, rather than the whole region, and therefore fewer equations need to be solved. Values at the required internal points only have to be obtained, rather than the solution over the whole interior region.

In order to be able to set up the BEM equations we need to know a fundamental solution to the equation and this is not always the case. Also, the BEM solution matrix is dense, not necessarily symmetric nor positive definite. It is not diagonally dominant. However, it is non-singular. The equations are not appropriate for the efficient solvers used in the FEM, although the search for such schemes is the subject of a good deal of current reseach, such as conjugate gradients (Broyden and Vespucci 2004), multipole acceleration (Mammoli and Ingber 1999, Popov and Power 2001), fast wavelet transforms (Bucher and Wrobel 2001).

2.2.4 Mesh-free methods

The three methods FDM, FEM and BEM are the most commonly used processes. However, recent interest has been growing in so-called 'mesh-free' methods. Researchers have seen mesh-free methods as being very efficient and accurate under suitable circumstances (Liu 2003). There is no need to define any sort of mesh; the solution is developed in terms of a set of basis functions which are defined over the whole domain. The methods are, in principle, easy to understand and are, in practice, easier to implement than FDM, FEM or BEM. We describe briefly two of these methods. Further

information and references can be found in the report by Davies and Crann (2000).

Kansa's Multiquadratic Method (MQM)

This method is a relatively new idea which has been investigated for elliptic partial differential equations. It has the advantage that a fundamental solution is not required. The approach is to approximate the solution surface using a scattered data approximation.

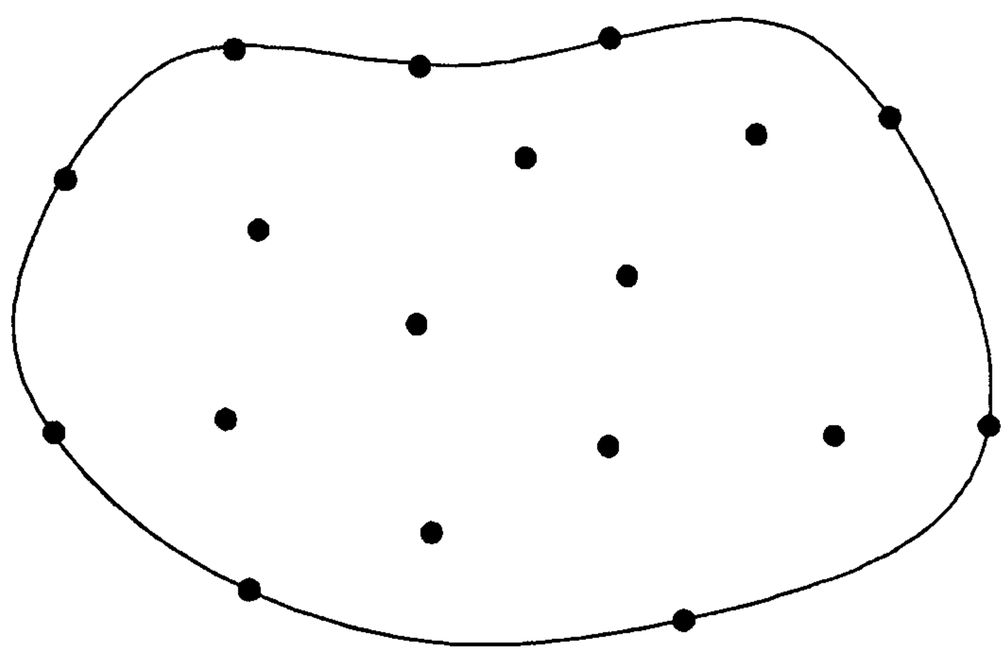

Figure 2.4: The region for the MQM

In this case a combination of radial basis functions is set up to interpolate the solution at every point, internally and on the boundary, using information from every node, see Figure 2.4.

A shape parameter is sought and different values are being investigated to aid stability. This method is remarkably simple and offers good results under certain conditions (Franke 1982). However, ill-conditioning is a significant problem and much work is currently being done to develop procedures that are not so susceptible to ill-conditioning.

The Method of Fundamental Solutions (MFS)

The method of fundamental solutions requires knowledge of the fundamental solution and so it is limited to those equations with a known fundamental

solution.

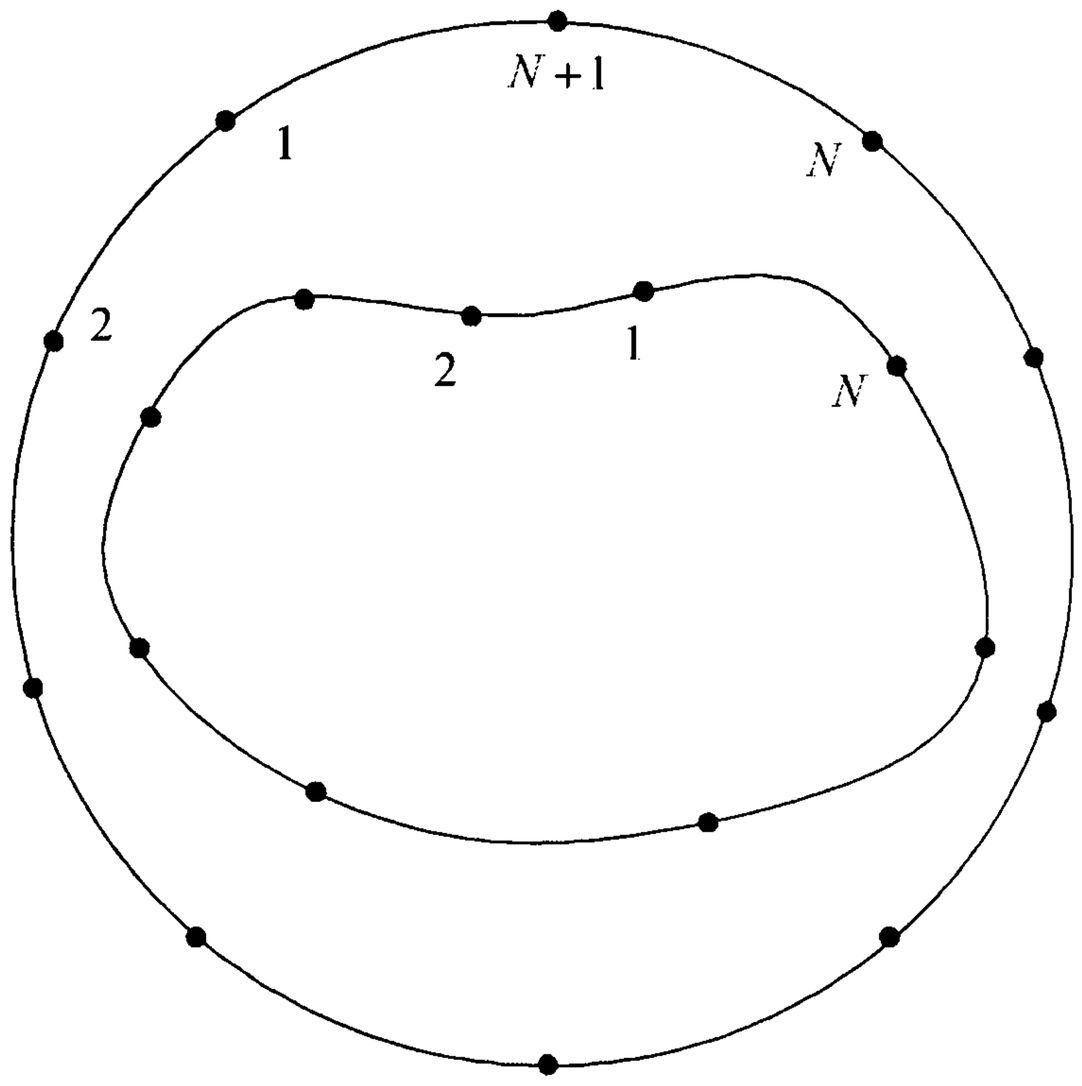

Figure 2.5: The discretised region for the MFS

The boundary is again discretised using N nodes. The whole region is surrounded by a known curve, usually a circle, discretised into $N+1$ nodes, see Figure 2.5. The solution is sought as a linear combination of fundamental solution values and a system of equations is developed using the boundary conditions. The set of equations is solved and values for internal points are found using these solutions.

The setting-up of the equations is straightforward and good results have been found for certain types of problem (Goldberg and Chen 1999). However the method also suffers from ill-conditioning problems similar to those in the MQM.

Chantasiriwan (2004) extends both MFS and MQM with additional terms in the setting up of the approximations. He reports good results for Poisson, Helmholtz and diffusion-convection problems.

2.3 Summary of Chapter 2

In this chapter we have set the scene for the solution of partial differential equations with boundary and initial conditions. Very few of these equations have analytical solutions. Numerical methods to solve these problems are almost always FDM (for elliptic, hyperbolic and parabolic equations), FEM (for elliptic equations) and BEM (for elliptic equations with a known fundamental solution). Researchers are investigating other methods of solution but such techniques are a long way from competing with the main three methods.

In the next chapter we describe the BEM in some detail.

Chapter 3

The Boundary Element Method

3.1 Introduction

Integral equation techniques in boundary-value problems have been used since the late nineteenth century. Green's second theorem in 1828 (Green 1828) and Somigliana's identity in 1886 (cited by Becker 1992) formed the basis of the direct approach in potential-type and elasticity problems respectively. Fredholm (1903) first published a basis of the 'indirect' boundary integral approach, using fictitious density functions or sources that have no physical meaning but can be used to calculate physical quantities such as displacements and stresses.

Integral formulations in potential and elasticity theory continued from Kellog (1929), Muskhelishvili (1953), Mikhlin (1957) and Kupradze (1965) but were solved analytically and were therefore limited to simple problems.

In the early sixties, the use of computers and numerical techniques started attracting much more interest in practical problems. Jaswon (1963) and Symm (1963) published the first modern 'semi-direct' formulation, where the functions used to formulate the problem can be differentiated or inte-

grated to calculate physical quantities. They used constant elements and employed Simpson's rule to evaluate the non-singular integrals, the singular integrals being integrated analytically. Similar integral equation approaches were adopted by Jaswon and Ponter (1963) for torsion problems and Hess and Smith (1964) for potential flow problems around arbitrary shapes. Harrington *et al.* (1969) continued similarly for two-dimensional electrical engineering problems.

Rizzo (1967) was the first to use the 'direct' approach of using physical quantities in an integral equation applicable over the boundary. It is interesting to note that Rizzo extended the ideas from potential problems to develop the BEM for elasticity in contrast to Zienkiewicz and Cheung (1965) who extended the FEM by applying ideas from elasticity to potential problems (Becker 2003). Cruse (1969) used a similar formulation to Rizzo to solve a three-dimensional problem using flat triangular elements on the surface. Other early work provided a firm foundation for boundary element development and demonstrated that the approach could be reliable and accurate. The name 'boundary element method' was first used by Brebbia and Dominguez (1977) who realised the analogy between the discretisation process for the boundary integral equation method and that for the already established finite element method.

Higher order elements, quadratic shape functions, were described by Lachat and Watson (1976). Together with further publications by Jaswon and Symm (1977), Brebbia (1978) and many others, the boundary element method was accepted as a serious alternative to the finite element method with clear advantages from the modelling point of view.

During the eighties the development of parallel computing received considerable attention since it offered the possibility of significantly improved computation times. Ortega and Voigt (1985) considered such approaches for finite differences and Lai and Liddell (1987) did the same for finite elements.

Symm (1984) described the first parallel implementation for the boundary element method and this work was continued by Davies (1988a, b, c) and subsequently by many others (Ingber and Davies 1997).

Cheng and Cheng (2005) give an excellent historical account of the development of the BEM with short biographies of the major contributors.

3.2 The Boundary Integral Equation

3.2.1 Laplace's equation

The basis of the BEM is that boundary-value problems involving partial differential equations can be transformed to boundary integral equations. We illustrate using the two-dimensional potential problem defined on a region D, bounded by the closed curve $C = C_1 + C_2$, see Figure 3.1.

Suppose that u satisfies Laplace's equation

$$\nabla^2 u = 0 \quad \text{in } D$$

subject to the Dirchlet condition

$$u = u_1(s) \quad \text{on } C_1$$

and the Neumann condition

$$\frac{\partial u}{\partial n} \equiv q = q_2(s) \quad \text{on } C_2$$

where $\mathbf{n}$ is the outward normal vector to C and s is the distance around C.

We would like to know u at any point inside, on or outside C. We consider only Dirichlet and Neumann conditions but the approach can easily be modified to incorporate a Robin boundary condition.

Suppose that $\mathbf{R}$ is the position vector of a point Q, relative to a point P. Surround P by a small disc, D_ϵ, centre P radius ϵ. The points P and Q are often called the source and field points respectively, see Figure 3.1.

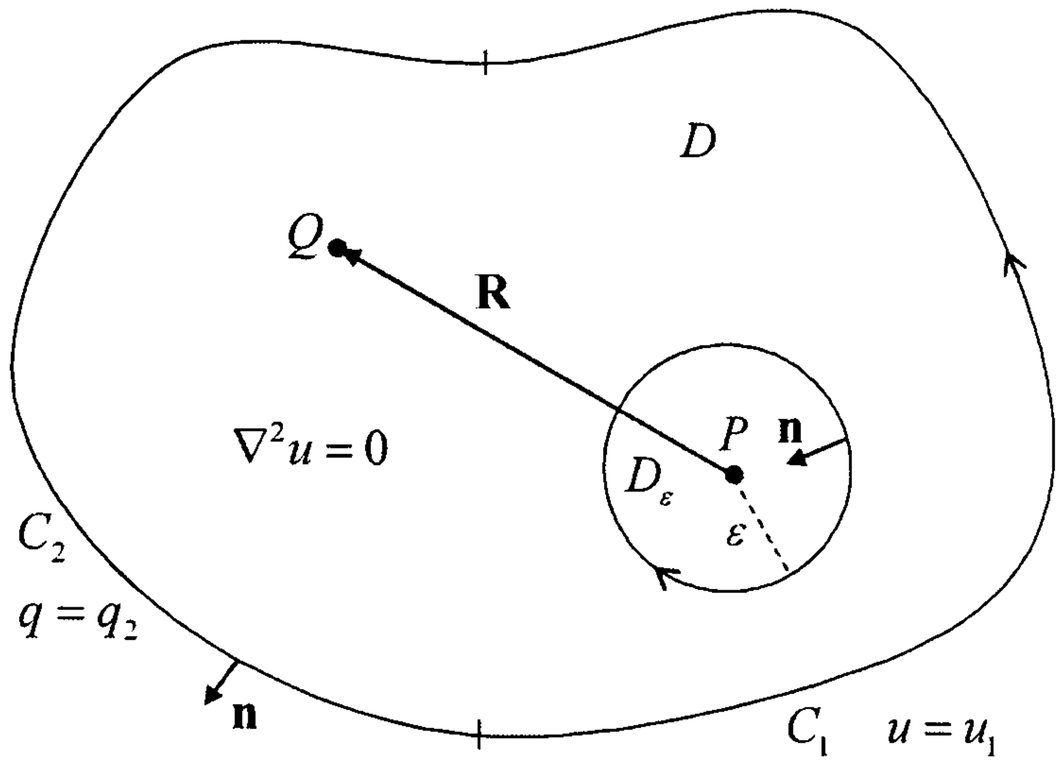

Figure 3.1: Potential problem in the region of D

A good description of the fundamental solution is given by Kythe (1996). It can be explained as the solution to the original partial differential equation over an unbounded region, subject to a point source of unit strength. In our case the fundamental solution satisfies Laplace's equation at all points except the point of application of the source. The fundamental solution, u^*, satisfies $\nabla^2 u^* = 0$ everywhere except at P where it has a logarithmic singularity. In particular $\nabla^2 u^* = 0$ in that part of D which excludes the disc D_ε.

We apply the second form of Green's theorem to the region $D - D_\varepsilon$

$$\int_{D-D_\varepsilon} (u\nabla^2 u^* - u^*\nabla^2 u)\, dA = \oint_{C+C_\varepsilon} \left(u\frac{\partial u^*}{\partial n} - u^*\frac{\partial u}{\partial n} \right) ds \qquad (3.1)$$

and consider what happens as $\varepsilon \to 0$ for P inside, on and outside the boundary C.

A fundamental solution of Laplace's equation in two dimensions is

$$u^* = -\frac{1}{2\pi}\ln R$$

For the interior solution for u suppose that P and Q are inside C. In the limit as $\varepsilon \to 0$, equation (3.1) becomes

$$u_P = \frac{1}{2\pi}\oint_C (u\frac{\partial}{\partial n}(\ln R) - q\ln R)\, ds \qquad (3.2)$$

Suppose that P itself is a point on the boundary at which there is a kink with angle α_P, see Figure 3.2, then in a similar manner to the derivation of equation (3.2), equation (3.1) becomes for points P on the boundary,

$$\frac{\alpha_P}{2\pi} u_P = \frac{1}{2\pi} \oint_C (u \frac{\partial}{\partial n}(\ln R) - q \ln R)\, ds \tag{3.3}$$

If the boundary is smooth at P then $\alpha = \pi$.

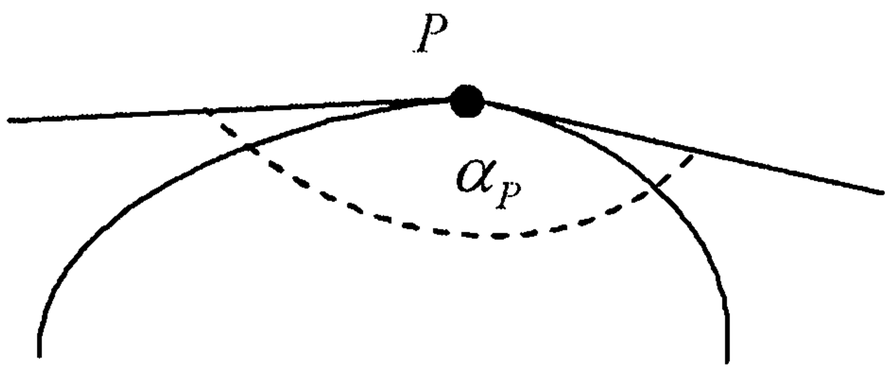

Figure 3.2: Point P on the boundary

If P is outside the boundary then

$$0 = \frac{1}{2\pi} \oint_C (u \frac{\partial}{\partial n}(\ln R) - q \ln R)\, ds \tag{3.4}$$

It is convenient to write these equations in the form

$$c_P u_P = \frac{1}{2\pi} \oint_C (u \frac{\partial}{\partial n}(\ln R) - q \ln R)\, ds$$

where

$$c_P = \begin{cases} 1 & \text{for } P \text{ inside the boundary} \\ \alpha_P/2\pi & \text{for } P \text{ on the boundary} \\ 0 & \text{for } P \text{ outside the boundary} \end{cases}$$

These equations, (3.2), (3.3) and (3.4) enable us to obtain values of u at any point, P, if we know the values of u and q everywhere on the boundary. Unfortunately this is not the case. For properly-posed problems we know only one of u or q at each boundary point, so before we can use equation (3.2) we must obtain both u and q everywhere on the boundary.

3.2.2 General second order linear partial differential equations

Laplace's equation is a special case of the second order partial differential equation

$$a_1\frac{\partial^2 u}{\partial x^2} + a_2\frac{\partial^2 u}{\partial y^2} + a_3\frac{\partial^2 u}{\partial x \partial y} + a_4\frac{\partial u}{\partial x} + a_5\frac{\partial u}{\partial y} + a_6 u = b(x,y)$$

i.e. in operator form $\mathcal{F}[u] = b$.

Suppose that $\mathcal{F}$ has a fundamental solution u^* with associated normal derivative q^*, then in a similar manner to the derivation of equations (3.2), (3.3) and (3.4) we can obtain the following integral formulation of the partial differential equation

$$c_P u_P = \oint_C (qu^* - uq^*)ds + \int_D u^* b\, dA \tag{3.5}$$

where

$$c_P = \begin{cases} 1 & P \in D \\ \alpha_P/2\pi & P \in C \\ 0 & P \notin D \cup C \end{cases}$$

We notice that if the equation is non-homogeneous then we have the domain integral $\int_D u^* b\, dA$ which needs special treatment and we shall consider this in Chapter 7. The homogeneous equation leads to a boundary only integral.

3.3 The Boundary Element Method

The integral equation in Section 3.2 has been known since the early nineteenth century but it has only been since the introduction of the modern digital computer in the nineteen sixties that the equation has been exploited as an important technique for the solution of the potential problem.

The boundary element method provides an approximate solution to the boundary integral equation. First we must approximate the boundary, C, by a simpler curve. We shall assume that C is approximated by a polygon,

C_N, the N edges of which are called the boundary elements. We choose a set of N points, called the nodes, at which we shall seek approximations U_i and Q_i $(i = 1, 2, \ldots, N)$ to the exact values u_i and q_i respectively. We shall adopt the numbering notation i to represent node number i and $[j]$ to represent element number j, see Figure 3.3.

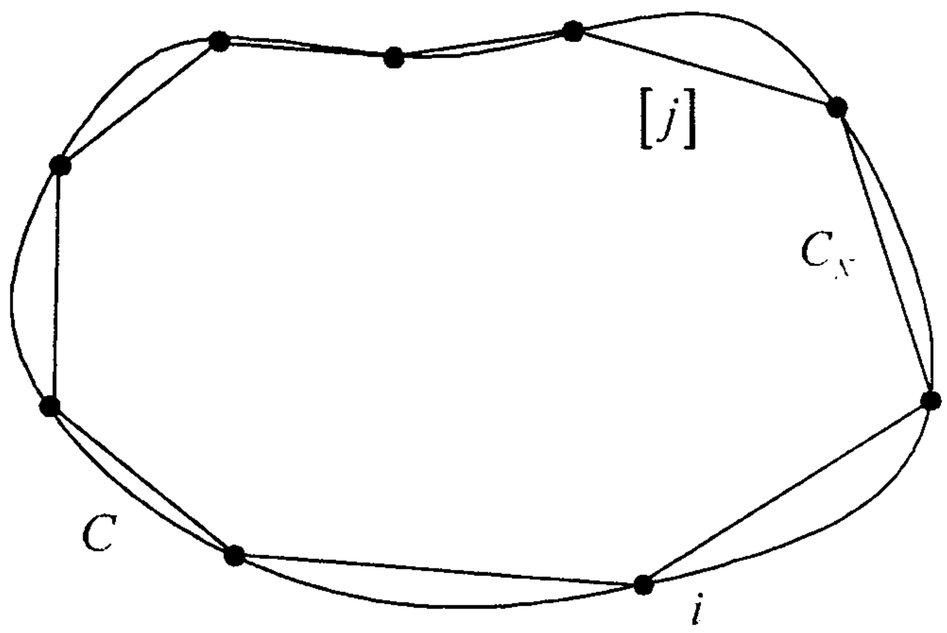

Figure 3.3: Boundary element approximation to the curve C

Suppose that $\{w_j(s) : j = 1, 2, \ldots, N\}$ is a set of linearly independent functions of arc length, s, around C_N, where, if node j is at the point s_j, then $w_i(s_j) = \delta_{ij}$ with the Kronecker delta given by

$$\delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases}$$

The boundary element approximations to the geometry may be of any order. We illustrate constant, linear and quadratic elements, see Figure 3.4.

Similarly we may approximate u and q using the same interpolation functions

$$\tilde{u} = \sum_{j=1}^{N} w_j(s) U_j \quad \text{and} \quad \tilde{q} = \sum_{j=1}^{N} w_j(s) Q_j \tag{3.6}$$

When the same interpolation is used to approximate the geometry and the unknowns we have the so-called *isoparametric elements.*

We shall use the point collocation method to find an approximate solution to equation (3.3) by substituting the approximations (3.6) into the

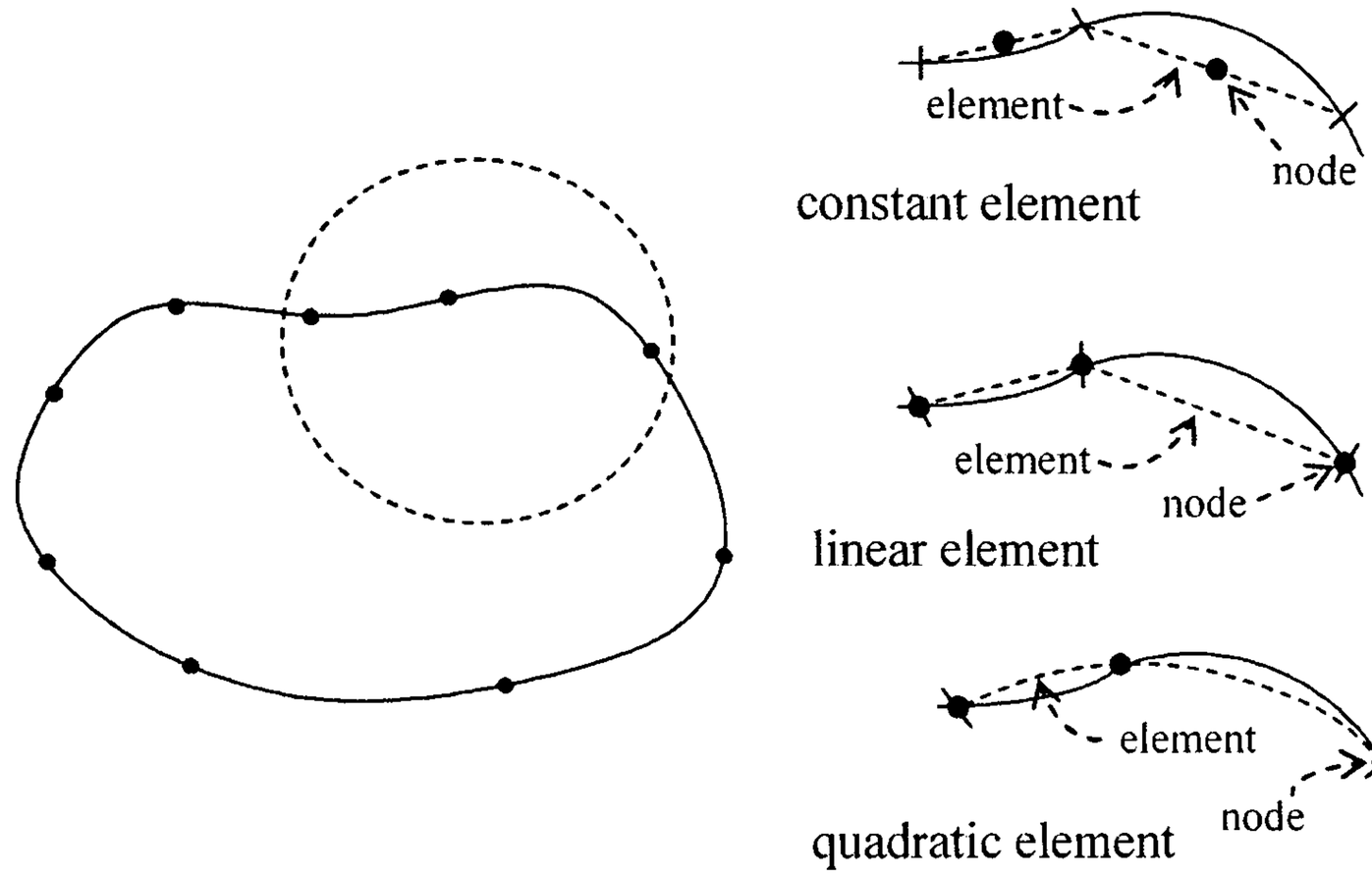

Figure 3.4: Constant, linear and quadratic boundary element approximations to the curve C

boundary integral equation (3.3) with the curve C replaced by C_N and choosing the boundary point P to be, successively, the nodes $1, 2, \ldots, N$. Hence we obtain, writing $c_i = \alpha_i/2\pi$,

$$c_i U_i = \frac{1}{2\pi} \oint_{C_N} \left[\left(\sum_{j=1}^{N} w_j(s) U_j \right) \frac{\partial}{\partial n}(\ln R_i) - \left(\sum_{j=1}^{N} w_j(s) Q_j \right) \ln R_i \right] ds$$

$$i = 1, 2, \ldots, N$$

which we may write as

$$c_i U_i = \sum_{j=1}^{N} \left(\frac{1}{2\pi} \oint_{C_N} w_j(s) \frac{\partial}{\partial n}(\ln R_i) ds) \right) U_j - \sum_{j=1}^{N} \left(\frac{1}{2\pi} \oint_{C_N} w_j(s) \ln R_i ds \right) Q_j$$

$$i = 1, 2, \ldots, N$$

where $R_i = |\mathbf{R}_i|$ and $\mathbf{R}_i(s)$ is the position vector of a boundary point, s, relative to node i.

We can rewrite this equation as

$$\sum_{j=1}^{N} H_{ij} U_j + \sum_{j=1}^{N} G_{ij} Q_j = 0$$

where

$$H_{ij} = \frac{1}{2\pi}\int_{[j]} w_j(s)\frac{\partial}{\partial n}(\ln R_{ij})\,ds - c_i\delta_{ij} \quad \text{and} \quad G_{ij} = -\frac{1}{2\pi}\int_{[j]} w_j(s)\ln R_{ij}\,ds$$

$R_{ij} = |\mathbf{R}_{ij}|$ and $\mathbf{R}_{ij}$ is the position vector of a point in the target element $[j]$ relative to the base node i, see Figure 3.5

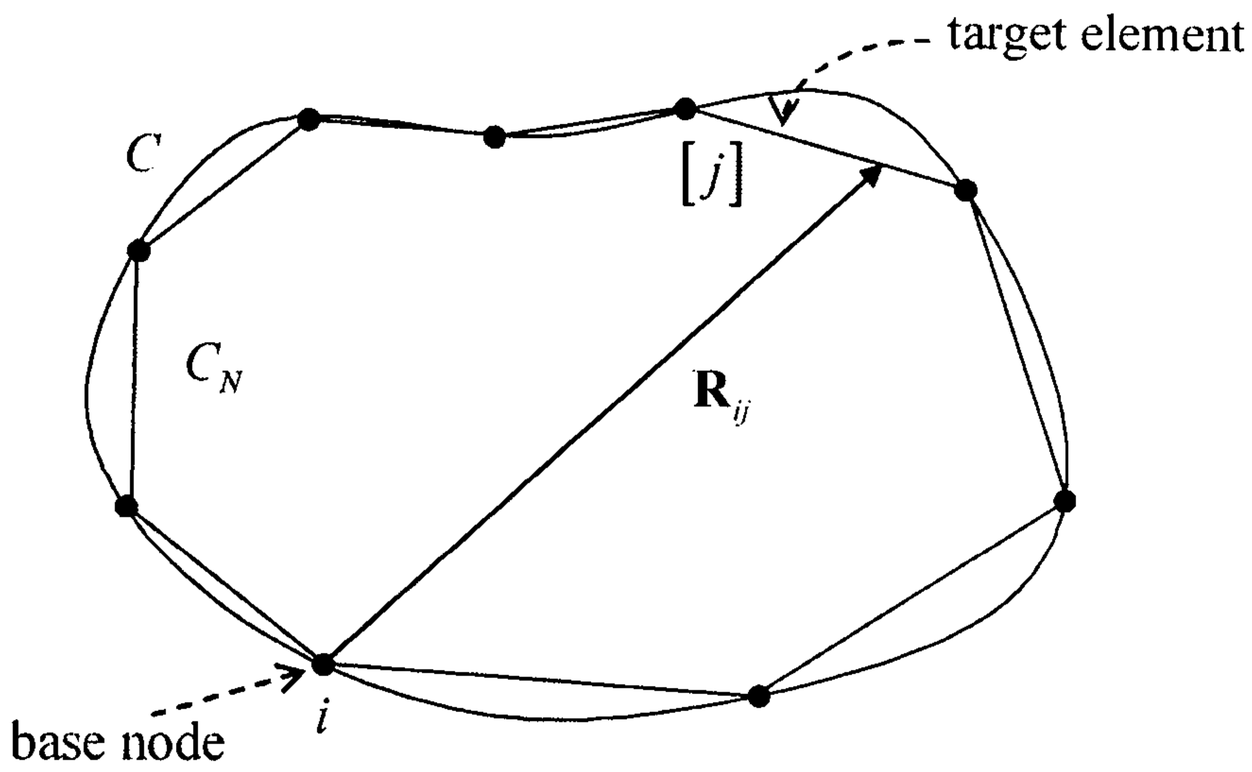

Figure 3.5: Target element relative to the base node

This enables us to approximate the unknown values on the boundary and subsequently obtain the solution at the required points around D. Full details of the method can be obtained from Brebbia and Dominguez (1989).

The approximation to the boundary integral equation can be written in matrix form

$$\mathbf{HU} + \mathbf{GQ} = \mathbf{0} \tag{3.7}$$

where $\mathbf{U}$ and $\mathbf{Q}$ are vectors of the boundary potentials and fluxes respectively.

However, for properly-posed problems we know only one of either u_i or q_i at any point and we partition the matrices to show $\mathbf{U}_1$ and $\mathbf{Q}_2$ the known values and $\mathbf{U}_2$ and $\mathbf{Q}_1$ the unknown values in the form

$$\begin{bmatrix} \mathbf{H}^1 & \mathbf{H}^2 \end{bmatrix}\begin{bmatrix} \mathbf{U}_1 \\ \mathbf{U}_2 \end{bmatrix} + \begin{bmatrix} \mathbf{G}^1 & \mathbf{G}^2 \end{bmatrix}\begin{bmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \end{bmatrix} = \mathbf{0}$$

The equations are rearranged in the form

$$\mathbf{Ax} = \mathbf{b}$$

with the system matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{H}^2 & \mathbf{G}^1 \end{bmatrix}$$

and

$$\mathbf{b} = -\begin{bmatrix} \mathbf{H}^1 & \mathbf{G}^2 \end{bmatrix} \begin{bmatrix} \mathbf{U}_1 \\ \mathbf{Q}_2 \end{bmatrix}$$

and the unknown vector

$$\mathbf{x} = \begin{bmatrix} \mathbf{U}_2 \\ \mathbf{Q}_1 \end{bmatrix}$$

and solved by a suitable linear equation solution routine.

In all our problems we have used Gaussian elimination with partial pivoting, a process which is $\mathcal{O}(N^3)$ for an $N \times N$ system. Recall from Section 2.2.3 that the BEM equations are densely populated, non-symmetric and non-positive definite, so that more efficient solvers such as conjugate gradient methods (Broyden and Vespucci 2004) cannot be used. We notice here that in the calculation of the coefficients in the matrices $\mathbf{H}$ and $\mathbf{G}$ the same computational effort is used no matter how far the base node is from the target element. However, as we have already mentioned, recent research has been directed at methods such as multipole expansions and wavelet transforms which exploit this fact to reduce the computational effort.

Once the boundary equations have been solved internal values are calculated at L points using the discretised form of

$$U_k = \frac{1}{2\pi} \oint_{C_N} \left[\left(\sum_{j=1}^{N} w_j(s) U_j \right) \frac{\partial}{\partial n} (\ln R_k) - \left(\sum_{j=1}^{N} w_j(s) Q_j \right) \ln R_k \right] ds$$

$$k = 1, 2, \ldots, L$$

or in matrix form

$$\mathbf{U}_{int} = \check{\mathbf{H}}\mathbf{U} + \check{\mathbf{G}}\mathbf{Q}$$

where

$$\check{H}_{jk} = \frac{1}{2\pi}\int_{[j]} w_j(s)\frac{\partial}{\partial n}\left(\ln R_{jk}\right) ds \quad \text{and} \quad \check{G}_{jk} = -\frac{1}{2\pi}\int_{[j]} w_j(s)\ln R_{jk} ds$$

Of the three methods FDM, FEM and BEM, the BEM is conceptually more difficult to understand and implement. The BEM comprises three distinct stages and it is important to be able to see how the method progresses from one stage to the next.

The spreadsheet offers an environment which is easy to use and ideal for small problems and for the investigation of the properties of the solutions such as convergence and for changing the geometry or boundary conditions. It is not necessary to rearrange equation (3.7). The facility 'Solver' in the *Excel*® spreadsheet package allows us to solve the equations directly and then find the internal solutions. Davies and Crann (1998) describe a constant element implementation on a spreadsheet.

3.4 Summary of Chapter 3

The boundary element method is now a well-accepted method and a powerful technique for solving elliptic problems when there is a known fundamental solution. The BEM is established as an effective alternative to the FDM and FEM.

In this chapter we have given a general introduction to boundary element history and theory, as far as we shall require it, and described the numerical implementation of the method for potential problems.

Chapter 4

Singular Integrals

4.1 Introduction

One of the problems encountered in boundary element computations is the evaluation of the integrals which occur when the base node is in the target element; if the kernel of the integral equation becomes infinite when the integration variable and collocation point coincide, then the integral becomes *singular*.

When the base node is not in the target element then the integrals are *regular*. Such integrals are commonly evaluated using Gauss quadrature. Equation (4.1) shows the numerical method for a function with a single independent variable:

$$\int_{-1}^{+1} f(\xi) d\xi \approx \sum_{g=1}^{G} w_g f(\xi_g) \tag{4.1}$$

where G is the total number of Gauss quadrature points, ξ_g is the Gauss coordinate, the abscissa, and w_g is the associated weight. The coordinates, which are roots of Legendre Polynomials, and the weights may be found in Stroud and Secrest (1966).

For potential problems with constant or linear elements, when the base node is in the target element, the singular integrals may be performed analyt-

ically (Jaswon and Symm 1977). For quadratic elements with straight edges analytic values have been given by Davies (1989). However, for isoparametric quadratic elements no such analytical values are available and an approximate method is required.

For other elliptic problems the resulting singular integrals cannot be integrated analytically and require a numerical evaluation *e.g.* in Chapter 5 we consider the modified Helmholtz equation with fundamental solution $\frac{1}{2\pi}K_0(pR)$, where K_0 is the modified Bessel function of the second kind and order zero and p is the Helmholtz parameter.

Gray (1993) uses the computer algebra package *Maple*® (Abell and Braselton 1994) to deal with singular integrals in an isoparametric Galerkin formulation, in a semi-analytic fashion. In a similar manner Ademoyero (2003) had partial success with the integrals involving Modified Bessel functions for the Modified Helmholtz equation. However, in general we must use a fully numerical approach and there are three commonly used ways of dealing with singular integrals. We shall describe these together with some others which have been investigated.

We note that when the base node is in the target element the integral has both non-singular and singular contributions.

4.2 Logarithmic Gauss quadrature

When the integrand contains a logarithmic function, $\ln(\xi)$, it is possible to use a logarithmic quadrature based on Gauss quadrature for regular integrals. The formula is shown in equation (4.2)

$$\int_0^1 f(\xi)\ln(\xi)d\xi \approx -\sum_{g=1}^{G} w_g f(\xi_g) \tag{4.2}$$

where the coordinates, ξ_g, and weights, w_g, are given by Stroud and Secrest (1966). Note that the integrals are effected over the interval $[0,1]$ com-

pared with the interval for regular integrals of $[-1, 1]$ and consequently an appropriate transformation must be made.

A logarithmic quadrature rule is described by Crow (1993) where a weighting function is used for the non-singular and singular part of the integral. This rule is used in a boundary element context by Smith (1996).

4.3 Telles self-adaptive scheme

A second numerical approach uses a transformation in such a way that the Jacobian is zero at the singular point, thus removing the singularity (Telles 1987). Conventional Gauss quadrature may then be used. The effect of the transform is to bunch the Gauss points towards the singularity.

The singular integrals are written in the form

$$I = \int_{-1}^{1} f(\xi) d\xi \tag{4.3}$$

and we seek a transformation $\xi \rightarrow \eta$ which maps $[-1, 1] \rightarrow [-1, 1]$ via a cubic polynomial

$$\xi = a\eta^3 + b\eta^2 + c\eta + d \tag{4.4}$$

Suppose that the integral has a singularity at $\bar{\xi}$ and that $\bar{\eta}$ is the corresponding value of η, then we choose a, b, c and d so that

$$\left(\frac{d^2\xi}{d\eta^2}\right)_{\bar{\eta}} = 0$$

$$\left(\frac{d\xi}{d\eta}\right)_{\bar{\eta}} = 0$$

$$\xi(1) = 1$$

$$\xi(-1) = -1$$

The values of a, b, c and d, given by Telles, are

$$a = \frac{1}{Q}, \quad b = -\frac{3\bar{\eta}}{Q}, \quad c = \frac{3\bar{\eta}^2}{Q}, \quad d = -b$$

where $Q = 1 + 3\bar{\eta}^2$. With these values a solution of equation (4.4) yields

$$\bar{\eta} = \left[\bar{\xi}\left(\bar{\xi}^2 - 1\right) + \left|\bar{\xi}^2 - 1\right|\right]^{\frac{1}{3}} + \left[\bar{\xi}\left(\bar{\xi}^2 - 1\right) - \left|\bar{\xi}^2 - 1\right|\right]^{\frac{1}{3}} + \bar{\xi}$$

and the value of the integral in equation (4.3) becomes

$$I = \int_{-1}^{1} f\left(\frac{(\eta - \bar{\eta})^3 + \bar{\eta}(\bar{\eta}^2 + 3)}{1 + 3\bar{\eta}^2}\right) \frac{3(\eta - \bar{\eta})^2}{1 + 3\bar{\eta}^2} d\eta \qquad (4.5)$$

The integrand in equation (4.5) is well-behaved in the neighbourhood of $\eta = \bar{\eta}$ and may be integrated using standard Gauss quadrature. As mentioned earlier, the effect of the transformation is to distribute the Gauss points so that they are bunched towards the singularity. In Figure 4.1 we show a geometrical transformation of a four-point quadrature rule in the case when $\bar{\eta} = 1$ with the relevant values in Table 4.1.

$$a = \frac{1}{4}, \quad b = -c = -d = -\frac{3}{4}$$

$$\xi = \frac{1}{4}\left[(\eta - 1)^3 + 4\right]$$

$$I = \frac{3}{4}\int_{-1}^{1} f\left(\frac{1}{4}\left[(\eta - 1)^3 + 4\right]\right)(\eta - 1)^2 \, d\eta$$

The Telles scheme is self-adaptive in that the effect of concentrating the quadrature points towards $\bar{\eta}$ is less marked as the singular point moves outside the domain of integration, *i.e.* as $|\bar{\eta}| > 1$. In fact as $|\bar{\eta}| \to \infty$ we have, from equation (4.5),

$$I \to \int_{-1}^{1} f(\eta) d\eta$$

and the integral degenerates to the standard form as in equation (4.3). Hence the Telles transformation could be used as a general numerical quadrature rule which deals automatically with regular, near singular and singular integrals.

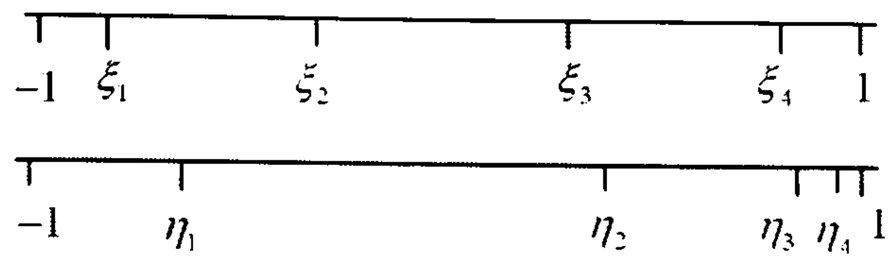

Figure 4.1: Transformation of the quadrature points for a four-point Gauss rule in the case $\bar{\eta} = 1$

Table 4.1: Quadrature points for a four-point Gauss rule and equivalent Telles transformation

Gauss points ξ_g	Telles points η_g
−0.861136	−0.611664
−0.339981	0.398500
0.339981	0.928120
0.861136	0.999331

4.4 Subtracting the singularity

A third method for evaluating singular integrals is to subtract out the singularity in such a way that the remaining integrand is regular and the subtracted singular part can be integrated analytically. Thus we write

$$\int_{-1}^{1} f(\xi)d\xi = \int_{-1}^{1} [f(\xi) - F(\xi)]\, d\xi + \int_{-1}^{1} F(\xi)d\xi \tag{4.6}$$

where $F(\xi)$ is a function which has the same singularity as $f(\xi)$ but in a simpler form which can be integrated exactly and $f(\xi) - F(\xi)$ is not singular and therefore can be integrated accurately by Gauss quadrature (Aliabadi 2002).

Since we do not investigate this method any further, we shall leave it here.

4.5 Automatic differentiation for the evaluation of singular integrals

In this section we seek the numerical evaluation of the singular integrals which occur when using quadratic elements. We consider the problem of evaluating the Taylor series for the Jacobian, J, of the transformation $s \to \xi$, $ds = J(\xi)d\xi$. We seek a sequence of numerical coefficient values, to an arbitrary order, without the explicit formulation of symbolic formulae to represent them (Crann, Christianson *et al.* 1997). The integral then becomes a finite sum of numerical coefficients multiplied by terms which may be integrated analytically. The accuracy of the value of the singular integral is determined by the degree of approximation in the Taylor series and does not depend on a numerical quadrature.

We use the ideas of Automatic Differention (AD) (Bartholomew-Biggs *et al.* 2000) in *fortran90*. Suitable data-types are defined in the form of coefficients of Taylor polynomials to an arbitrary degree and operator overloading is used to implement the computations. The usual numerical operators, plus, minus, multiplication, division *etc.*, are defined and algebraic manipulation is developed on the data-types. The module containing the algebraic constructs is shown in the appendix.

4.5.1 Laplace's equation

In two-dimensional boundary element calculations for potential problems the fundamental solution is $u^* = -\frac{1}{2\pi} \ln R$ and hence it is necessary to evaluate weakly singular integrals of the form

$$\int_{-1}^{1} w_j \left(s(\xi)\right) J(\xi) \ln(R(\xi)) d\xi$$

involving logarithmic singularities.

We shall consider the quadratic element with nodes 1, 2 and 3 whose

position vectors are

$$\mathbf{r}_1 = (x_1, y_1), \quad \mathbf{r}_2 = (x_2, y_2), \quad \mathbf{r}_3 = (x_3, y_3)$$

Using the local coordinate $\{\xi : -1 \leq \xi \leq 1\}$ and Lagrange quadratic interpolation polynomials

$$L_1(\xi) = \frac{1}{2}\xi(\xi - 1), \quad L_2(\xi) = 1 - \xi^2, \quad L_3(\xi) = \frac{1}{2}\xi(\xi + 1)$$

the equation which defines the geometry of the element is given by

$$\mathbf{r}(\xi) = \sum_{i=1}^{3} L_i(\xi)\mathbf{r}_i \quad i = 1, 2, 3 \tag{4.7}$$

If $\mathbf{R}_j(\xi) = \mathbf{r}(\xi) - \mathbf{r}_j$ is the position vector of a point, $\mathbf{r}(\xi)$, in the element relative to the base node $\mathbf{r}_j$, then we require the evaluation of the following nine singular integrals:

$$I_{ij} = \int_{-1}^{1} L_i(\xi) J(\xi) \ln R_j(\xi) d\xi \qquad i, j = 1, 2, 3 \tag{4.8}$$

where the Jacobian, $J(\xi)$, is given by

$$J(\xi) = \left[\mathbf{r}'(\xi).\,\mathbf{r}'(\xi)\right]^{\frac{1}{2}}$$

Suppose that the singularity occurs when $\xi = \xi_0$ *i.e.* $\mathbf{r}(\xi_0) = \mathbf{r}_j$, and let $\Delta\xi = \xi - \xi_0$ then

$$\begin{aligned} R_j(\xi) &= |\mathbf{R}_j(\xi)| \\ &= |\mathbf{r}(\xi) - \mathbf{r}_j| \\ &= |\mathbf{r}'(\xi_0)\Delta\xi + \tfrac{1}{2}\mathbf{r}''(\xi_0)\Delta\xi^2| \\ &= |\Delta\xi| \left[d_0 + \Delta\xi d_1 + \Delta\xi^2 d_2\right]^{\frac{1}{2}} \\ &= |\Delta\xi| \left[R_d(\xi)\right]^{\frac{1}{2}} \end{aligned}$$

where

$$d_0 = \mathbf{r}'(\xi_0).\,\mathbf{r}'(\xi_0) \tag{4.9}$$

$$d_1 = \mathbf{r}'(\xi_0).\,\mathbf{r}''(\xi_0) \tag{4.10}$$

$$d_2 = \frac{1}{4}\mathbf{r}''(\xi_0).\,\mathbf{r}''(\xi_0) \tag{4.11}$$

$$R_d(\xi) = d_0 + \Delta\xi d_1 + \Delta\xi^2 d_2 \tag{4.12}$$

Also

$$\begin{aligned} J(\xi) &= [\mathbf{r}'(\xi).\,\mathbf{r}'(\xi)]^{\frac{1}{2}} \\ &= \left[d_0 + 2d_1\Delta\xi + 4d_2\Delta\xi^2\right]^{\frac{1}{2}} \end{aligned}$$

We develop all the terms in the integrand, equation (4.8), as Taylor polynomials. This approach is similar to the direct factorisation technique described by Smith and Mason (1982).

The interpolation polynomials are easily written as second degree Taylor polynomials as follows:

$$\begin{aligned} L_i(\xi) &= L_i(\xi_0) + L_i'(\xi_0)\Delta\xi + \tfrac{1}{2}L_i''(\xi_0)\Delta\xi^2 \\ &= l_0 + l_1\Delta\xi + l_2\Delta\xi^2, \text{ say.} \end{aligned}$$

The Jacobian, $J(\xi)$, and the term $\ln R_d(\xi)$ may be expanded automatically as n^{th} degree polynomials

$$J(\xi) \approx j_0 + j_1\Delta\xi + j_2\Delta\xi^2 + \ldots + j_n\Delta\xi^n$$

and

$$\ln R_d(\xi) \approx b_0 + b_1\Delta\xi + b_2\Delta\xi^2 + \ldots + b_n\Delta\xi^n$$

Now we form the product of the two Taylor polynomials for $L_i(\xi)$ and $J(\xi)$ as

$$\begin{aligned} L_i(\xi)J(\xi) &\approx (l_0 + l_1\Delta\xi + l_2\Delta\xi^2)(j_0 + j_1\Delta\xi + j_2\Delta\xi^2 + \ldots + j_n\Delta\xi^n) \\ &\approx a_0^{(1)} + a_1^{(1)}\Delta\xi + \ldots + a_n^{(1)}\Delta\xi^n \end{aligned} \tag{4.13}$$

where we truncate the product at the $\mathcal{O}(\Delta\xi^n)$ term.

Similarly we determine

$$L_i(\xi)J(\xi)\ln\left[R_d(\xi)\right]^{\frac{1}{2}} = L_i(\xi)J(\xi)\frac{1}{2}\ln(R_d) \approx$$

$$\left(l_0 + l_1\Delta\xi + l_2\Delta\xi^2\right)\left(j_0 + j_1\Delta\xi + \ldots + j_n\Delta\xi^n\right) \times$$

$$\times \frac{1}{2}\left(b_0 + b_1\Delta\xi + \ldots + b_n\Delta\xi^n\right)$$

$$\approx a_0^{(2)} + a_1^{(2)}\Delta\xi + \ldots + a_n^{(2)}\Delta\xi^n \tag{4.14}$$

The approximate value of the integral may now be obtained from

$$\begin{aligned} I_{ij} &= \int_{-1}^{1} L_i(\xi) J(\xi) \ln R_j(\xi) d\xi \\ &= \int_{-1}^{1} L_i(\xi) J(\xi) \ln |\Delta\xi| d\xi + \int_{-1}^{1} L_i(\xi) J(\xi) \ln \left[R_d(\xi)^{\frac{1}{2}}\right] d\xi \\ &\approx \int_{-1}^{1} \sum_{k=0}^{n} a_k^{(1)} \Delta\xi^n \ln |\Delta\xi| d\xi + \int_{-1}^{1} \sum_{k=0}^{n} a_k^{(2)} \Delta\xi^n d\xi \\ &= \sum_{k=0}^{n} \left(a_k^{(1)}\alpha_k + a_k^{(2)}\beta_k\right) \end{aligned}$$

where

$$\alpha_k = \int_{-1}^{1} \Delta\xi^k \ln|\Delta\xi| d\xi$$

$$\beta_k = \int_{-1}^{1} \Delta\xi^k d\xi$$

and $a_k^{(1)}$ and $a_k^{(2)}$ are sequences of numerical coefficients for the Taylor polynomials truncated at the $\mathcal{O}(\Delta\xi^n)$ term.

There are three cases to consider:

1. Singularity at $\mathbf{r}_1$, *i.e.* $\xi_0 = -1$

$$\alpha_k = \frac{2^{k+1}}{k+1}\left\{\ln 2 - \frac{1}{k+1}\right\} \quad \beta_k = \frac{2^{k+1}}{k+1}$$

2. Singularity at $\mathbf{r}_2$, *i.e.* $\xi_0 = 0$

$$\alpha_k = \begin{cases} 0 & k \text{ odd} \\ -\dfrac{2}{(k+1)^2} & k \text{ even} \end{cases} \qquad \beta_k = \begin{cases} 0 & k \text{ odd} \\ \dfrac{2}{k+1} & k \text{ even} \end{cases}$$

3. Singularity at $\mathbf{r}_3$, *i.e.* $\xi_0 = +1$

$$\alpha_k = -\frac{(-2)^{k+1}}{k+1}\left\{\ln 2 - \frac{1}{k+1}\right\} \quad \beta_k = -\frac{(-2)^{k+1}}{k+1}$$

The convergence of the sequence as n increases requires that $|\Delta\xi| < \rho$ where ρ is the radius of convergence of the series. This condition forces a restriction on the placement of the position vectors $\mathbf{r}_1, \mathbf{r}_2$ and $\mathbf{r}_3$.

Before attempting to develop the Taylor polynomials we must ensure that $\mathbf{r}_2$ is suitably placed. Consider the situation shown in Figure 4.2 where we illustrate geometrically the definition of the co-ordinate (X, Y).

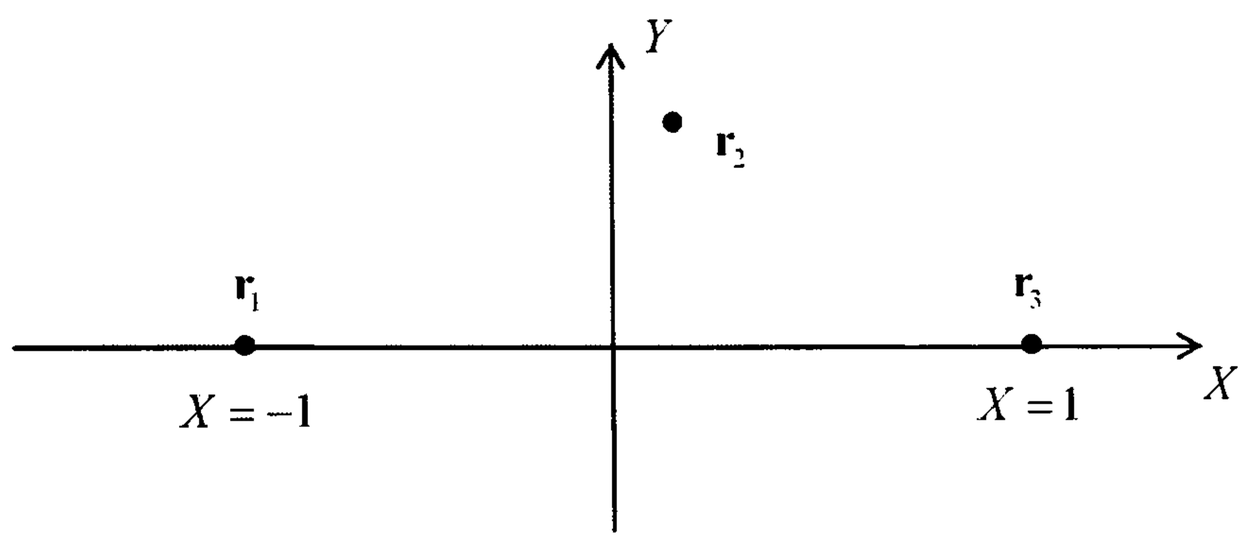

Figure 4.2: Definition of the co-ordinate (X, Y) in the quadratic element

From equation (4.7), using the definition of the Lagrange interpolation polynomials, we see that

$$\mathbf{r}'(\xi) = \mathbf{a} + 2\mathbf{b}\xi \text{ and } \mathbf{r}''(\xi) = 2\mathbf{b}$$

where

$$\mathbf{a} = \frac{1}{2}(\mathbf{r}_3 - \mathbf{r}_1) \quad \text{and} \quad \mathbf{b} = \frac{1}{2}(\mathbf{r}_1 - 2\mathbf{r}_2 + \mathbf{r}_3)$$

In (X, Y) co-ordinates we have

$$\mathbf{r}_1 = (-1, 0), \quad \mathbf{r}_2 = (X, Y), \quad \mathbf{r}_3 = (1, 0)$$

so that

$$\mathbf{a} = (1, 0) \quad \text{and} \quad \mathbf{b} = (-X, -Y)$$

and

$$\begin{aligned} J(\xi) &= |\mathbf{r}'(\xi).\,\mathbf{r}'(\xi)|^{\frac{1}{2}} \\ &= \left(1 - 4X\xi + 4(X^2 + Y^2)\xi^2\right)^{\frac{1}{2}} \end{aligned}$$

The Taylor series for the square root in $J(\xi)$ requires that, for convergence,

$$\left|4(X^2+Y^2)\xi^2 - 4X\xi\right| < 1 \quad \text{with } -1 \leq \xi \leq 1$$

which we may write as

$$\left|\left(X\xi - \frac{1}{2}\right)^2 + (Y\xi)^2 - \frac{1}{4}\right| < \frac{1}{4}$$

i.e.

$$0 < \left(X\xi - \frac{1}{2}\right)^2 + (Y\xi)^2 < \frac{1}{2}$$

The worst case corresponds to $\xi = \pm 1$ so that

$$0 < \left(X \pm \frac{1}{2}\right)^2 + Y^2 < \frac{1}{2}$$

and this region is the intersection of the two circles with radius $\frac{1}{\sqrt{2}}$ and centres at $(\pm\frac{1}{2}, 0)$. Hence the point $\mathbf{r}_2$ must lie in the shaded region in Figure 4.3.

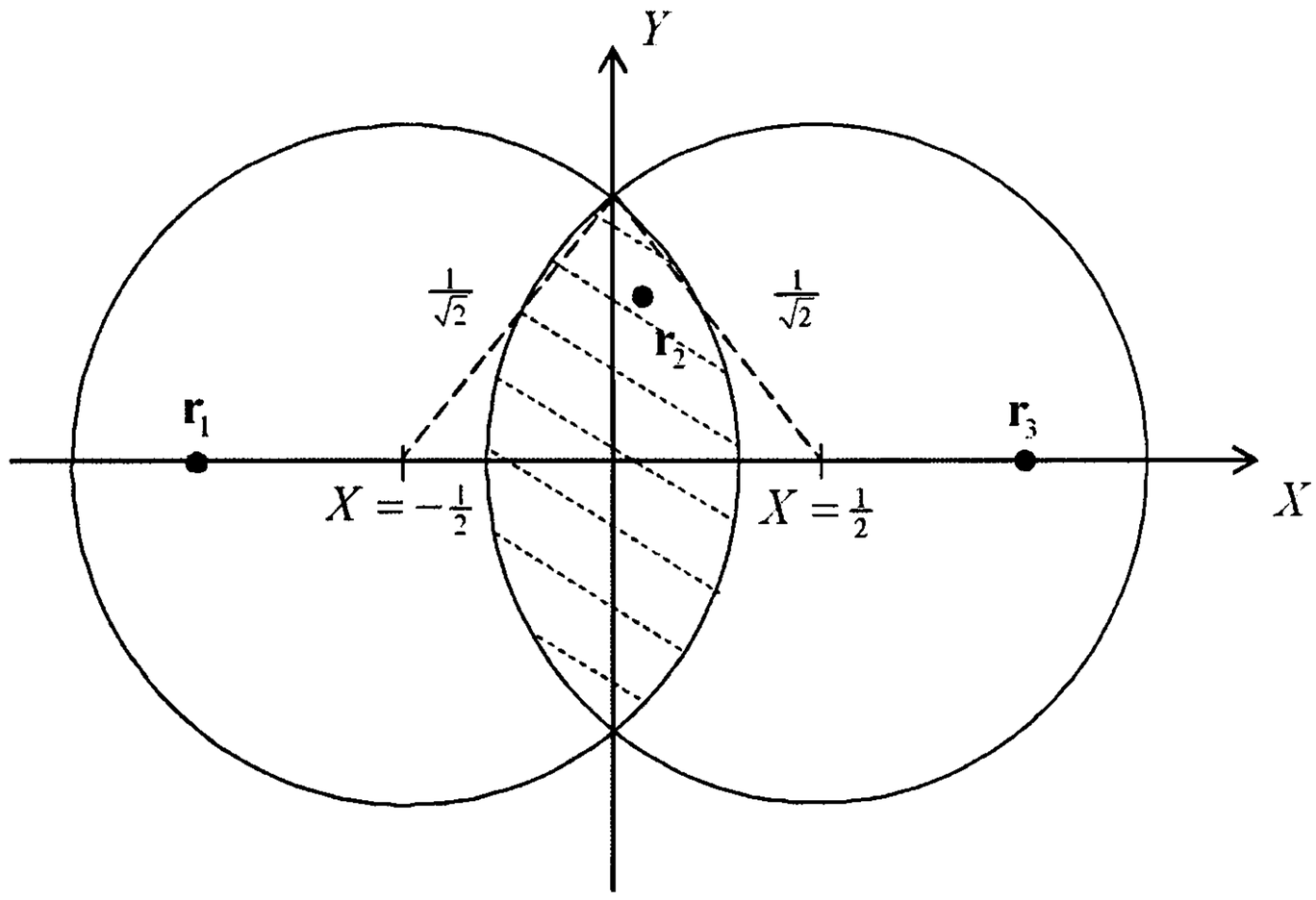

Figure 4.3: Region for the position of point $\mathbf{r}_2$ for convergence of the Taylor series

Also we perform a Taylor series expansion for $\ln R_d(\xi)$. Using equations (4.9), (4.10) and (4.11) we have

$$d_0 = (1 - 2X\xi_0)^2 + 4Y^2\xi_0^2$$

$$d_1 = -2X + 4X^2\xi_0 + 4Y^2\xi_0$$

$$d_2 = X^2 + Y^2$$

so that using equation (4.12)

$$\ln R_d(\xi) = \ln \left[1 - 4X\xi_0 + 4X^2\xi_0^2 + 4Y^2\xi_0^2 \right.$$
$$\left. + \left(-2X + 4X^2\xi_0 + 4Y^2\xi_0\right)(\xi - \xi_0) + \left(X^2 + Y^2\right)(\xi - \xi_0)^2\right]$$

Now the Taylor series expansion of this expression for $\ln R_d(\xi)$ requires that, for convergence,

$$\left|-4X\xi_0 + 4X^2\xi_0^2 + 4Y^2\xi_0^2 \right.$$
$$\left. + \left(-2X + 4X^2\xi_0 + 4Y^2\xi_0\right)(\xi - \xi_0) + \left(X^2 + Y^2\right)(\xi - \xi_0)^2 \right| < 1$$

with $-1 \leq \xi \leq 1$

$$\left|[X(\xi + \xi_0) - 1]^2 - 1 + [Y(\xi + \xi_0)]^2\right| < 1$$

i.e.

$$0 < [X(\xi + \xi_0) - 1]^2 + [Y(\xi + \xi_0)]^2 < 2$$

which is always satisfied provided

$$(\pm 2X - 1)^2 + (2Y)^2 < 2$$

i.e.

$$0 < \left(X \pm \frac{1}{2}\right)^2 + Y^2 < \frac{1}{2}$$

and this is the same restriction as for the convergence of the Jacobian.

Consequently for the convergence of the AD method it suffices that the point $\mathbf{r}_2$ lies in the shaded region in Figure 4.3.

So far we have established that for convergence it is sufficient that $\mathbf{r}_2$ is placed inside the shaded region in Figure 4.3. We now develop the parameter, σ, which we shall use to check convergence. If we consider the geometry in Figure 4.4 then provided

$$PQ < \min(PA, PB)$$

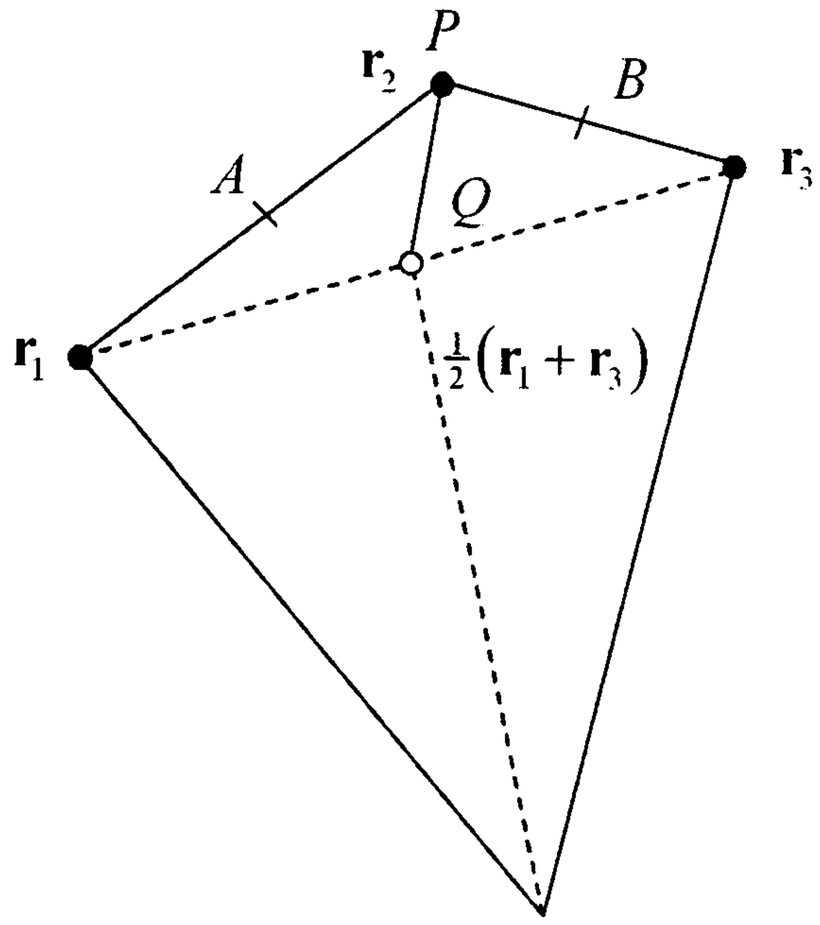

Figure 4.4: The geometry for $PQ < \min(PA, PB)$

$\mathbf{r}_2$ will be suitably placed.

We define

$$\sigma = \frac{\min\left\{\frac{1}{2}\left|\mathbf{r}_1 - \mathbf{r}_2\right|, \frac{1}{2}\left|\mathbf{r}_3 - \mathbf{r}_2\right|\right\}}{\left|\mathbf{r}_2 - \frac{1}{2}(\mathbf{r}_1 + \mathbf{r}_3)\right|} \tag{4.15}$$

Convergence will occur provided σ is sufficiently large. A very crude discretisation of a quadrant of a circle of unit radius into two equal quadratric elements has $\sigma \approx 2.6$. A value of $\sigma = 3$ requires about twenty Taylor terms to produce an accuracy of about ten decimal places. From a practical point of view a value of σ greater than 3 is likely to be satisfactory.

4.5.2 Modified Helmholtz equation

In Chapter 6 the use of the Laplace transform leads to evaluating weakly singular integrals arising in the modified Helmholtz equation. Previous authors use the tables of Ramesh and Lean (1991) based on the formulae by Abramowitz and Stegun (1972) to evaluate the modified Bessel functions of the second kind and order zero. Using a *fortran90* approach we develop the Taylor series directly from the formula, in such a way that the coefficients can be extended to arbitrary order as required instead of restricting

ourselves to the normal seven coefficients used by Ramesh and Lean.

The modified Helmholtz equation is

$$\nabla^2 u - p^2 u = 0 \quad \text{in } D \tag{4.16}$$

subject to the usual boundary conditions on C.

The weakly singular integrals with quadratic elements analogous to equation (4.8) are of the form

$$I_{ij} = \int_{-1}^{1} L_i(\xi) J(\xi) K_0 \left(pR_j(\xi)\right) d\xi \quad i,j = 1,2,3 \tag{4.17}$$

where $K_0(x)$ is the modified Bessel function of the second kind and order zero.

Abramowitz and Stegun give the formula

$$K_0(x) = -I_0(x)\left(\ln\left(\frac{x}{2}\right) + \gamma\right) + \sum_{r=1}^{\infty} \frac{1}{(r!)^2}\phi(r)\left(\frac{x}{2}\right)^{2r} \tag{4.18}$$

with

$$I_0(x) = \sum_{r=0}^{\infty} \frac{1}{(r!)^2}\left(\frac{x}{2}\right)^{2r} \qquad \phi(r) = \sum_{s=1}^{r} \frac{1}{s}$$

and γ, Euler's constant, given by

$$\gamma = \lim_{n\to\infty}\{\phi(n) - \ln(n)\} = 0.5772156649\ldots$$

$I_0(x)$ is the modified Bessel function of the first kind and order zero, and, for small x, $I_0(x)$ is well-behaved so we have $K_0(x) \sim -\ln(x)$ as $x \to 0$.

Ramesh and Lean provide explicit values for the first seven coefficients in the power series in equation (4.18). Previous authors use these values together with a logarithmic Gauss quadrature to evaluate the singular integrals with constant elements (Rizzo and Shippy 1970). Ramesh and Lean use linear elements and give recursive expressions in each of which are analytic contributions to the integrals. Both sets of authors use expressions which are equivalent to truncating the series after seven terms *i.e.* the x^{12} term.

We use the expression given by Ramesh and Lean

$$K_0(px) = -\ln\left(\frac{px}{2}\right)\sum_{i=0}^{6} A_{2i}x^{2i} + \sum_{i=0}^{6} B_{2i}x^{2i} \quad 0 < px \leq 2 \tag{4.19}$$

where

$$A_{2i} = A'_i\left(\frac{p}{3.75}\right)^{2i}, \quad B_{2i} = B'_i\left(\frac{p}{2}\right)^{2i}, \quad A_{2i+1} = B_{2i+1} = 0$$

with A'_i and B'_i given in Table 4.2.

Table 4.2: Coefficients in the Ramesh and Lean series for $K_0(px)$

i	A'_i	B'_i
0	1.0000000	-0.57721677
1	3,5167229	0.42289420
2	3.0899424	0.23069756
3	1.2067492	0.03488590
4	0.2659732	0.00262698
5	0.0360768	0.00010750
6	0.0045813	0.00000740

The Taylor series development follows in a similar manner to that for the potential problem and we use the same notation with the Taylor polynomials being of order twelve *i.e.* $n = 0, \ldots, 12$ (13 terms).

From equation (4.17)

$$L_i(\xi)J(\xi)K_0\left(pR_j(\xi)\right) = L_i(\xi)J(\xi)K_0\left(p|\Delta\xi|\left[R_d(\xi)\right]^{\frac{1}{2}}\right) \tag{4.20}$$

$$= \left(l_0 + l_1\Delta\xi + l_2\Delta\xi^2\right)\left(j_0 + j_1\Delta\xi + \ldots + j_n\Delta\xi^n\right) \times \tag{4.21}$$

$$\times \left[\left(-\ln|\Delta\xi|\right) - \frac{1}{2}\ln\left(\frac{p^2}{4}R_d(\xi)\right)\sum_{i=0}^{6} A'_i\left(\frac{p}{3.75}\left[R_d(\xi)\right]^{\frac{1}{2}}\Delta\xi\right)^{2i}\right.$$

$$\left. + \sum_{i=0}^{6} B'_i\left(\frac{p}{2}\left[R_d(\xi)\right]^{\frac{1}{2}}\Delta\xi\right)^{2i}\right]$$

With

$$\ln\left(\frac{p^2}{4}R_d(\xi)\right) = \left(\ln\frac{p^2}{4}\right) + b_0 + b_1\Delta\xi + \ldots + b_n\Delta\xi^n$$

we may write

$$L_i(\xi)J(\xi)K_0\left(pR_j(\xi)\right)$$

$$= \left(b_0^{(1)} + b_1^{(1)}\Delta\xi + \ldots + b_n^{(1)}\Delta\xi^n\right)\ln|\Delta\xi|$$

$$+ \left(b_0^{(2)} + b_1^{(2)}\Delta\xi + \ldots + b_n^{(2)}\Delta\xi^n\right)$$

The coefficients $b_k^{(1)}$ and $b_k^{(2)}$ are obtained using the *fortran90* Taylor polynomial data types and operator overloading to evaluate the necessary operations of addition, multiplication and to evaluate the natural logarithm. We note that there is no need to use square root because $[R_d(\xi)]^{\frac{1}{2}}$ is always raised to an even power.

Finally, then

$$I_{ij} = \sum_{k=0}^{n}\left(b_k^{(1)}\alpha_k + b_k^{(2)}\beta_k\right)$$

where α_k and β_k are given by cases 1, 2 and 3 in Section 4.5.1.

We obtain an estimate of a bound on the error due to the truncation of the series for K_0 as follows:

$$\begin{aligned} I_{ij} &= \int_{-1}^{1} L_i(\xi)J(\xi)K_0\left(pR_j(\xi)\right)d\xi \\ &= \int_{-1}^{1} L_i(\xi)J(\xi)\left(\tilde{K}_0\left(pR_j(\xi)\right) + \varepsilon\right)d\xi \\ &= \tilde{I}_{ij} + e \end{aligned}$$

so that

$$\begin{aligned} |e| &\leq \int_{-1}^{1} |L_i(\xi)||J(\xi)||\varepsilon|d\xi \\ &\leq |\varepsilon|_{\max}\int_{\mathbf{r}_1}^{\mathbf{r}_3} ds \end{aligned}$$

The quadrant of the circle, on the straight line joining $\mathbf{r}_1$ and $\mathbf{r}_3$ as chord, has length $\frac{\pi l_{13}}{2\sqrt{2}}$, see Figure 4.5. Hence we have the bound

$$|e| < \frac{\pi l_{13}}{2\sqrt{2}}|\varepsilon|_{\max} \tag{4.22}$$

where Abramowicz and Stegun give $|\varepsilon|_{\max} \approx 10^{-7}$ provided that $0 < pR_j(\xi) < 2$.

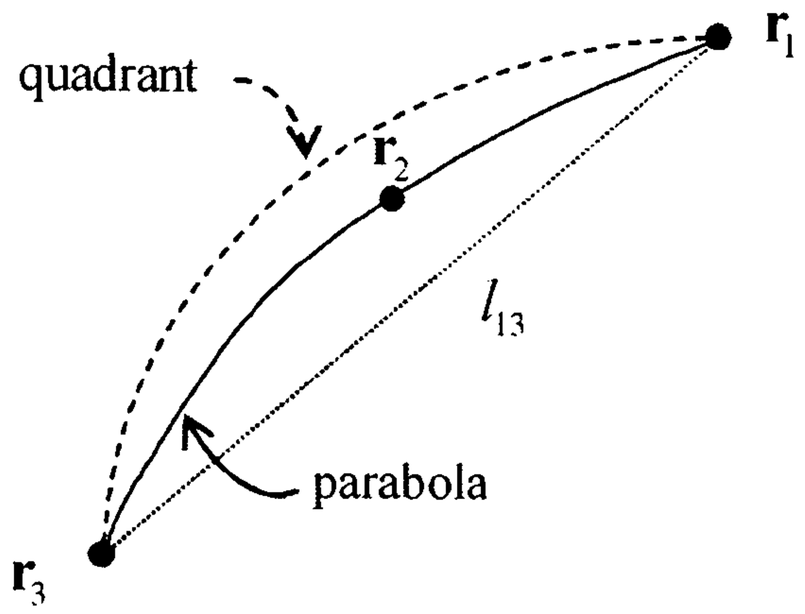

Figure 4.5: Quadrant of the circle, on the straight line joining $\mathbf{r}_1$ and $\mathbf{r}_3$

The Taylor expansions in equation (4.20) require that $\mathbf{r}_2$ is placed so that the Jacobian, $J(\xi)$, and the term $\ln R_d(\xi)$ both converge and we have developed the required condition in Section 4.5.1.

We need also consider the two finite sums in equation (4.21). Now, by virtue of equation (4.19) we require

$$0 < p^2 R_d |\Delta\xi|^2 < 4$$

which we may write, using the simplification in Section 4.5.1, as

$$0 < p^2 \left([X(\xi + \xi_0) - 1]^2 + [Y(\xi + \xi_0)]^2 \right) (\xi - \xi_0)^2 < 4$$

A sufficient condition is that

$$\left(X \pm \frac{1}{2} \right)^2 + Y^2 < \frac{1}{p^2}$$

To ensure that these two circles intersect we require $p < 2$, and if $p < \sqrt{2}$ the position of $\mathbf{r}_2$ to ensure convergence of $J(\xi)$ and $\ln R_d(\xi)$ will also be sufficient for the series in equation (4.19).

4.6 Other methods

4.6.1 Beale and Attwood's Correction method

Beale and Lai (2001) describe a method for the evaluation of singular integrals of the type $\ln|\xi|$ and Beale and Attwood (2002) extend this idea to near-singularities of the type $\ln|\xi^2 + a^2|$ where a is a small number. They apply standard rules of numerical integration that hold for smooth functions but fail in this case, then develop a correction term for the singularity or near-singularity that allows the evaluation of these integrals to third-order accuracy.

We consider the case of the singular integral of the form

$$I = \int_{-1}^{1} f(\xi)\ln|\xi|\, d\xi$$

which is approximated by

$$I \approx \sum_{j=-N}^{N} w_j f(\xi_j)\ln(\xi_j)$$

The numerical integration method is based on the Euler-Maclaurin summation formula with the set of weights $w_j, -N \leq j \leq N$ derived from Bernoulli numbers. The weights in the interior of the interval are all set to one while the outer points take different values according to the particular rule, *e.g.* when we choose $w_N = w_{-N} = \frac{1}{2}$ then we have the trapezoidal rule. Following Beale and Attwood we shall use the case

$$w_N = w_{-N} = \frac{5}{12} \quad w_{N-1} = w_{1-N} = \frac{13}{12} \quad w_j = 1 \text{ for } |j| < N-1$$

We observe the uncorrected term

$$S = \sum_{j=-N, j\neq 0}^{N} f(jh)\ln|jh|w_j h$$

where $h = 1/N$, and then add the correction term

$$h\ln\left(\frac{h}{2\pi}\right) f(0)$$

to obtain the approximation to I.

For such a relatively simple method, the corrected approximation converges to the value of the integral with $\mathcal{O}(h^3)$ accuracy.

4.7 Results for Laplace's equation

The examples in this section, together with those in the following section, were developed using *fortran90* with suitable data types for the Taylor polynomials and operator overloading to define operations on the polynomials. Details of the relevant *fortran90* constructs are given in Appendix.

Example 4.1

This example is considered by Smith (1996) and the element has nodal coordinates

$$\mathbf{r}_1 = (0.1, 0.1), \quad \mathbf{r}_2 = (0.2, 0.2 + \alpha), \quad \mathbf{r}_3 = (0.3, 0.3)$$

with $\alpha = 0,\ 0.02,\ 0.04\ 0.1$.

In Tables 4.3, 4.4, 4.5 and 4.6 we compare the results for the integral I_{i1} of our AD method using 6- and 20-degree Taylor polynomials with an accurate numerical approximation obtained using the adaptive numerical quadrature procedure available in the symbolic computation package *Maple* together with those calculated by Smith using Crow's method and those obtained using Gauss/log-Gauss 4- and 10-point quadrature. The convergence parameter σ is defined in equation (4.15).

For $\alpha = 0.0$ our AD results are as good approximations as those from the other methods. For $\alpha = 0.02$ our AD results for the 20-degree Taylor approximation are also as good as the others, but the 6-degree Taylor approximation is beginning to lose accuracy. For $\alpha = 0.04$ and $\alpha = 0.1$ our convergence parameter σ shows us that, at $\sigma < 2.6$, our results are

Table 4.3: Example 4.1 Values of $|I_{i1}|$ with α=0.0, $\sigma = \infty$

x-coordinate	0.1	0.2	0.3
y-co-ordinate	0.1	0.2	0.3
Maple	0.1930966	0.3952628	0.0516752
Smith	0.1930967	0.3952628	0.0516752
Gauss-4	0.1930966	0.3952628	0.0516753
Gauss-10	0.1930964	0.3952628	0.0516755
AD 6deg	0.1930966	0.3952628	0.0516752
AD 20deg	0.1930966	0.3952628	0.0516752

Table 4.4: Example 4.1 Values of $|I_{i1}|$ with α=0.02, $\sigma = 3.91$

x-coordinate	0.1	0.2	0.3
y-co-ordinate	0.1	0.22	0.3
Maple	0.2182667	0.3868547	0.0378031
Smith	0.2182668	0.3868548	0.0378030
Gauss-4	0.2182668	0.3868547	0.0378031
Gauss-10	0.2182665	0.3868546	0.0378033
AD 6deg	0.2176048	0.3884189	0.0368806
AD 20deg	0.2182667	0.3868547	0.0378031

Table 4.5: Example 4.1 Values of $|I_{i1}|$ with α=0.04, $\sigma = 2.15$

x-coordinate	0.1	0.2	0.3
y-co-ordinate	0.1	0.24	0.3
Maple	0.2438544	0.3806926	0.0270271
Smith	0.2438547	0.3806916	0.027028o
Gauss-4	0.2438548	0.3806905	0.0270296
Gauss-10	0.2438542	0.3806924	0.0270274
AD 6deg	0.2406218	0.3804058	0.0194077
AD 20deg	0.2438514	0.3807016	0.0270217

unlikely to be acceptable. However, our 20-degree Taylor approximation is still within 4 decimal places of the *Maple* approximation for $\alpha = 0.04$.

Table 4.6: Example 4.1 Values of $|I_{i1}|$ with α=0.1, $\sigma = 1.12$

x-coordinate	0.1	0.2	0.3
y-co-ordinate	0.1	0.3	0.3
Maple	0.3159910	0.3780244	0.0203769
Smith	0.3160620	0.3778107	0.0203340
Gauss-4	0.3161028	0.3774731	0.0206118
Gauss-10	0.3159909	0.3780238	0.0203771
AD 6deg	0.3200557	0.3841124	0.0366342
AD 20deg	0.3287444	0.316909	0.0330741

Example 4.2

We consider the curved element with nodes

$$\mathbf{r}_1 = (1,0) \qquad \mathbf{r}_2 = \left(0.5 + \alpha\sqrt{2},\, 0.5 + \alpha\sqrt{2}\right) \qquad \mathbf{r}_3 = (0,1)$$

In Tables 4.7 to 4.10 we compare the results with an accurate result obtained using the symbolic computation package *Maple*. The tables show the absolute value of the integral for each method, Gauss/log-Gauss quadrature 10 point, Telles self-adaptive method 20 point, AD 20-degree Taylor polynomial and Beale and Attwood's method with $N = 1000$. We give results for α in the range $0 \leq \alpha \leq 0.1$, since in any reasonable discretisation α would not be outside this range and to agree with the convergence criterion value for σ we need $\alpha < 0.08$.

Table 4.7: Example 4.2 Values of $|I_{ij}|$ with $\alpha = 0.0$, $\sigma = \infty$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.5861349	0.4589215	0.1209719	0.1602556	1.583831
G/LG 10pt	0.5861339	0.4589211	0.1209707	0.1602562	1.583830
Telles 20pt	0.5861350	0.4589215	0.1209719	0.1602556	1.584883
AD 20deg	0.5861349	0.4589214	0.1209719	0.1602556	1.583831
Beale 1000pt	0.5861375	0.4589165	0.1209709	0.1602555	1.583832

We see as may be expected that, in general, accuracy decreases as α increases and this is much more pronounced for Beale and Attwood's method. However our AD approach has results which compare very well with the

Table 4.8: Example 4.2 Values of $|I_{ij}|$ with $\alpha = 0.001$, $\sigma = 176.8$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.5861380	0.4589190	0.1209726	0.1602563	1.583833
G/LG 10pt	0.5861370	0.4589188	0.1209714	0.1602568	1.583832
Telles 20pt	0.5861381	0.4589190	0.1209726	0.1602563	1.584885
AD 20deg	0.5861380	0.4589190	0.1209726	0.1602563	1.583833
Beale 1000pt	0.5861439	0.4589244	0.1209719	0.1602561	1.583834

Table 4.9: Example 4.2 Values of $|I_{ij}|$ with $\alpha = 0.01$, $\sigma = 17.7$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.5864408	0.4586762	0.1210427	0.1603226	1.584021
G/LG 10pt	0.5864398	0.4586761	0.1210415	0.1603232	1.584020
Telles 20pt	0.5864410	0.4586762	0.1210427	0.1603226	1.585073
AD 20deg	0.5864408	0.4586763	0.1210427	0.1603226	1.584021
Beale 1000pt	0.5867881	0.4596170	0.1210743	0.1603227	1.584030

Table 4.10: Example 4.2 Values of $|I_{ij}|$ with $\alpha = 0.1$, $\sigma = 1.8$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.6140498	0.4351688	0.1277617	0.1659600	1.601812
G/LG 10pt	0.6140489	0.4351685	0.1277605	0.1659606	1.601811
Telles 20pt	0.6140499	0.4351687	0.1277617	0.1659600	1.602864
AD 20deg	0.6138296	0.4354450	0.1277488	0.1659600	1.601812
Beale 1000pt	0.6450671	0.5220976	0.1307252	0.1659603	1.602733

other methods and even with the convergence criterion for $\alpha = 0.1$ being less than our acceptable value, σ being approximately 1.8, the AD results are within 10^{-3} accuracy.

4.8 Results for the Modified Helmholtz equation

Example 4.3

We consider the curved element with nodes

$$\mathbf{r}_1 = (0.5, 0) \qquad \mathbf{r}_2 = \left(0.25 + \alpha\sqrt{2},\, 0.25 + \alpha\sqrt{2}\right) \qquad \mathbf{r}_3 = (0, 0.5)$$

We present here in Tables 4.11 to 4.15 the results of a variety of tests against an accurate numerical evaluation using *Maple*. We consider Taylor polyno-

mials of degree 13 calculated by the Ramesh and Lean formula (RL) together with Taylor polynomials of degree 13, 21 and 31, calculated by the Abramowitz and Stegun formula (AS) directly. Further results can be found in Crann, Christianson *et al.* (1997, 1998).

Table 4.11: Example 4.3 Values of $|I_{ij}|$ with $\alpha = 0.0$, $\sigma = \infty$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.3864977	0.6452758	0.0564923	0.1807186	1.181271
RL Taylor-13	0.3864977	0.6452758	0.0564923	0.1807186	1.181271
AS Taylor-13	0.3864977	0.6452759	0.0564923	0.1807186	1.181271
AS Taylor-21	0.3864977	0.6452759	0.0564923	0.1807186	1.181271
AS Taylor-31	0.3864970	0.6452759	0.0564923	0.1807186	1.181271

Table 4.12: Example 4.3 Values of $|I_{ij}|$ with $\alpha = 0.001$, $\sigma = 76.1$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.3865075	0.6452769	0.0564954	0.1807239	1.181280
RL Taylor-13	0.3865074	0.6452768	0.0564953	0.1807239	1.181280
AS Taylor-13	0.3865070	0.6452806	0.0564993	0.1807239	1.181280
AS Taylor-21	0.3865070	0.6452806	0.0564993	0.1807239	1.181280
AS Taylor-31	0.3865070	0.6452806	0.0564993	0.1807239	1.181280

Table 4.13: Example 4.3 Values of $|I_{ij}|$ with $\alpha = 0.01$, $\sigma = 8.85$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.3874758	0.6453778	0.0568020	0.1812449	1.182166
RL Taylor-13	0.3874758	0.6453779	0.0568020	0.1812498	1.182166
AS Taylor-13	0.3874271	0.6457512	0.0571990	0.1812524	1.182168
AS Taylor-21	0.3874271	0.6457512	0.0571990	0.1812524	1.182168
AS Taylor-31	0.3874271	0.6457512	0.0571990	0.1812524	1.182168

Table 4.14: Example 4.3 Values of $|I_{ij}|$ with $\alpha = 0.05$, $\sigma = 1.84$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.4092240	0.6478030	0.0638339	0.1930659	1.202788
RL Taylor-13	0.4090073	0.6469381	0.0648255	0.1930664	1.202787
AS Taylor-13	0.4076398	0.6572410	0.0758860	0.1931376	1.202842
AS Taylor-21	0.4081165	0.6573899	0.0752118	0.1931370	1.202843
AS Taylor-31	0.4075470	0.6584936	0.0749358	0.1931370	1.202843

Table 4.15: Example 4.3 Values of $|I_{ij}|$ with $\alpha = 0.1$, $\sigma = 1.01$

	$I_{11} = I_{33}$	$I_{12} = I_{32}$	$I_{13} = I_{31}$	$I_{21} = I_{23}$	I_{22}
Maple	0.4627852	0.6549658	0.0823129	0.2218715	1.258965
RL Taylor-13	0.5511404	0.6150085	-0.014123	0.2234573	1.252695
AS Taylor-13	0.5591817	0.5785737	0.0829721	0.2238240	1.252934
AS Taylor-21	1.1836290	-0.934008	0.1468918	0.2233508	1.254540
AS Taylor-31	29.740850	-105.4296	19.699090	0.2208742	1.265079

We see that for $\alpha = 0.1$ the σ value is less than the acceptable test parameter of 3, and the results are meaningless. However it is surprising to note that the results are not too inaccurate for the $I_{2,i}$ integrals. Also, the σ test value for $\alpha = 0.05$, at 1.84, is less than the required value of 3 but the results are still very reasonable.

The results for the Ramesh and Lean formula are closer to the *Maple* results for greater values of α compared with the Abramowitz and Stegun formula results but this is not surprising since we suspect that the *Maple* package uses the Ramesh and Lean formula to evaluate these integrals.

Notice that using the bound (4.22) developed in Section 4.5 we have, in this case, $l_{13} = \frac{1}{\sqrt{2}}$ and the error due to the truncation of the modified Bessel function is of the order 8×10^{-8}. This is very small compared with errors due to the numerical quadrature and hence makes very little contribution to the error in the integral.

4.9 Efficiency of the methods for evaluating singular integrals

In Table 4.16 we show the operation count for each of the methods described in Example 4.2 and we see that the Gauss/log-Gauss integral requires significantly fewer operations than the others (Crann *et al.* 2003).

In terms of ease of implementation we note first that the AD approach would be adopted only in an environment which supports operator over-

Table 4.16: Operation count for each method

	L-G 10pt	Telles 20pt	A-D 20deg	Beale 1000pt
+ -	1,431	4,169	38,436	153,089
* /	1,962	7,634	15,051	210,139
sqrt,log	162	362	138	15,012
Total	3,555	12,165	53,625	378,240

loading and Taylor series data-types. Also there is a significant cost in code generation so a general user would be unlikely to adopt it even though it is a once only cost. However, its attraction to users is that the errors are due only to truncation errors in the Taylor series and not to a numerical quadrature rule. For smaller values of α, A-D gives the best accuracy. For the other three methods the code implementation costs are very similar. We also note here that Beale and Attwood's method is interesting because it does not require a data set of quadrature points which depend on the order of quadrature but is not as accurate as the other methods.

Comparing the four methods, in general we see that the Gauss/log-Gauss method provides the best overall approach in terms of accuracy, efficiency and ease of implementation.

4.10 Summary of Chapter 4

In this chapter we consider a variety of different methods for handling the singularity which arises in the evaluation of the integrals in BEM when the base node is in the target element. We develop a new method using the ideas of automatic differentiation with Taylor polymonial coefficients and use a number of examples to demonstrate its use with singular integrals in the solution of Laplace's equation and the modified Helmholtz equation. We also define a condition on the geometry of the integral to enable us to ensure convergence of the method.

The AD Taylor polynomial method in a *fortran90* environment provides

a suitable approach for evaluating the quadratic boundary element singular integrals. In terms of accuracy it compares well with alternative methods. However the attraction of the method lies in the fact that the Taylor coefficients are obtained without symbolic evaluation of derivatives. Indeed the approach offers a possibility for evaluating the significantly more difficult singular integrals which occur in boundary element computations.

Chapter 5

The Laplace Transform Method

5.1 Introduction

In the boundary element solution of problems which are parabolic in the time variable there are several numerical techniques with which the time variable can be handled. A time-dependent fundamental solution may be used directly to derive the BEM formulations over space and time (Chang *et al.* 1973). Another technique interprets the time derivative in the diffusion equation as a body force and solves the problem using the dual reciprocity method (Wrobel 2002). An early application of the finite difference method in the time variable was given by Curran *et al.* (1980) who consider both first and second order schemes. A variety of time-marching schemes for two and three-dimensional problems and for axisymmetric problems is decribed by Brebbia *et al.* (1984). There are possible problems with the finite difference method since there may be severe restrictions on the step-size to ensure accurary or, especially, stability (Smith 1978).

An alternative possibility is to take the Laplace transform in the time variable and solve the resulting elliptic problem using the BEM then invert-

ing back using a numerical inversion process. Rizzo and Shippy (1970) first used this method with an inversion method suggested by Schapery (1962). Their inversion method was a curve fitting process and presupposed knowledge of the expected solution. The Laplace space transform parameter was arbitrarily chosen and a poor choice resulted in unstable solutions or insufficient definition of the curve which therefore reduced accuracy.

Lachat and Combescure (1977) used the Laplace transform and boundary integral equation methods to applications of transient heat conduction problems and inverted using complex Legendre polynomials. They reported the method as being very ill-conditioned and limited in use to certain problems only.

Moridis and Reddell (1991a, b, c) describe a family of Laplace transform-based numerical methods, finite difference, finite element and boundary element methods, for diffusion-type partial differential equations in groundwater flow applications. The Black-Scholes equation provides a model for european options in computational finance and is of diffusion-type. Crann, Davies, Lai and Leong (1998) and Lai *et al.* (2005) use this in an innovative approach using the Laplace transform with Stehfest's inversion process, solving the space equation using the Finite Volume Method (Jameson and Mavriplis 1986). Zhu *et al.* (1994) also use the Laplace transform with the Stehfest inversion method with the BEM and dual reciprocity for diffusion problems and we shall discuss this approach later in Chapter 7.

The Laplace transform boundary element method for time-dependent problems is now well-established. It provides a technique for the solution of partial differential equations for initial boundary-value problems in which the number of independent variables is reduced by one. Ordinary differential equations become algebraic equations, equations such as the one-dimensional wave and diffusion equations become ordinary differential equations. Hyperbolic and parabolic problems in time are transformed into elliptic problems

in the transform space. The advantages of the method are that there is no time-step stability problem as occurs with the usual FDM and if the solution is required at just one time value then there is no need for the computation of solutions at intermediate times. After application of the Laplace transform a variety of techniques may be employed to solve the resulting elliptic problem. We shall illustrate, using a simple model problem, how a variety of elliptic solvers may be employed.

The difficulty associated with the method manifests itself in the inversion which is required after the transformed equation has been solved. If the transformed equations have suitable analytic solutions then the inversion may be effected either directly from tables (Davies and Crann 2004) or using the complex inversion formula (Davies 2002). If, however, such solutions are not suitable or if numerical solutions are obtained, then inversion can cause serious problems.

5.2 The Laplace transform

Suppose that $f(t)$ is defined and is of exponential order for $t \geq 0$ *i.e.* there exists A, $\gamma > 0$ and $t_0 > 0$ such that $|f(t)| < A\exp(\gamma t)$ for $t > t_0$. Then providing $\lambda > \gamma$ the Laplace transform, $\bar{f}(\lambda)$, exists and is given by

$$\mathcal{L}[f(t)] \equiv \bar{f}(\lambda) = \int_0^\infty f(t)\mathrm{e}^{-\lambda t}dt \tag{5.1}$$

The problem of finding $f(t)$ from $\bar{f}(\lambda)$ using equation (5.1)

$$f(t) = \mathcal{L}^{-1}\left[\bar{f}(\lambda)\right] \tag{5.2}$$

is a much more difficult situation. It is a Fredholm integral equation of the first kind and such equations are known to be ill-conditioned in their solution (Wing 1991). Also $e^{-\lambda t}$ smooths out the values of $f(t)$ for relatively large t and consequently recovery of the function from the transform is likely to be

difficult. We shall address this particular problem for periodic functions in the next section.

We now consider numerical methods for inverting the Laplace transform.

5.3 Laplace transform numerical inversion

No single algorithm is known which is universally applicable to all functions. Davies (2002) describes some important facts when considering the use of an appropriate algorithm:

1. the source of values of the transform, whether the available data has only real values,

2. the precision required for the particular problem,

3. the number of time values required, how expensive the computation will be,

4. reliability of the problem compared with a similar representative class of transforms.

An evaluation of many methods can be found in the paper by Davies and Martin (1979). They test a range of algorithms on a range of transforms whose exact inverses are known.

Most of the methods require evaluation at complex values of the transform parameter. However, since the methods which involve only real values of the transform parameter are relatively easy to implement and our problems all contain real variables, we have chosen to consider algorithms which require only real values. Davies and Martin suggest a number of such methods and report that Stehfest's method gives good results on a fairly wide range of functions. As well as Stehfest's method we also consider an extension, by Aral and Gülçat (1977), of the method introduced by Zakian and Littlewood (1973) based on shifted Legendre polynomials. Davies and

Martin consider a method using Legendre polynomials and report that it seldom gives high accuracy, but although they did test the shifted Legendre polynomials method they didn't feel that the results were a marked improvement.

5.3.1 Stehfest's numerical inversion

Stehfest (1970) developed an inversion formula which is a weighted sum of transform values at a discrete set of transform parameters and is derived from a stochastic inversion process described by Gaver (1966). We note here that Stehfest says that his method is unlikely to be accurate for problems in which $f(t)$ is oscillatory or for finding the inverse close to a discontinuity in $f(t)$. In Section 5.4 and in Chapter 8 we shall consider an approach using Stehfest's method which overcomes these difficulties.

If $\bar{f}(\lambda)$ is the Laplace transform of $f(t)$ then the inversion algorithm is as follows:

We seek the value, $f(T)$, for a specific value $t = T$.

Choose a discrete set of transform parameters

$$\lambda_j = j\frac{\ln 2}{T} \qquad j = 1, 2, \ldots, M \tag{5.3}$$

where M is even.

The approximate numerical inversion is given by

$$f(T) \approx \frac{\ln 2}{T} \sum_{j=1}^{M} w_j \bar{f}(\lambda_j) \tag{5.4}$$

where the weights, w_j, are given by

$$w_j = (-1)^{\frac{M}{2}+j} \sum_{k=[\frac{1}{2}(1+j)]}^{\min(j,\frac{M}{2})} \frac{(2k)!k^{\frac{M}{2}}}{(\frac{M}{2}-k)!k!(k-1)!(j-k)!(2k-j)!} \tag{5.5}$$

The user chooses a value of M and various authors have considered the most appropriate values. Stehfest suggests that for eight-digit accuracy a

Table 5.1: Stehfest's weights for $M = 6, 8, 10, 12$ and 14

$M = 6$	$M = 8$	$M = 10$	$M = 12$	$M = 14$
1	-1/3	1/12	-1/60	1/360
-49	145/3	-385/12	961/60	-461/72
366	-906	1279	-1247	18481/20
-858	16394/3	-46871/3	82663/3	-484371/14
810	-43130/3	505465/6	-1579685/6	486289/9
-270	18730	-473915/2	1324138.7	-131950391/30
	-35840/3	375912	-58375583/15	21087592
	8960/3	-340072	21159859/3	-63944913
		328125/2	-8005336.5	127597580
		-65625/2	5552830.5	-170137188
			-2155507.2	150327467
			359251.2	-84592161.5
				824366543/30
				-117766649/30

value of $M = 10$ should be used. Moridis and Reddell (1991c) suggest that the accuracy of the method is insensitive to changes in the value of M for $6 \leq M \leq 20$ and Crann (1996) suggests in general that accuracy decreases with increasing $M \geq 10$. Zhu *et al.* (1994) report that $M = 6$ gives the best accuracy. It is not possible to state what the optimum value might be since this is problem dependent but values in the region $6 \leq M \leq 10$ are usually satisfactory.

In Table 5.1 we compare values of the weights for $M = 6, 8, 10, 12$ and 14 and we see that the values of w_j become numerically very large as M increases. With numerically large values of w_j associated with larger values of M, it is likely that there will be round-off error problems in the inversion process.

5.3.2 Shifted Legendre polynomials (SLP)

Aral and Gülçat (1977) describe a solution of the wave equation with time dependent boundary conditions. They use the Laplace transform together with the finite element method and a numerical inversion process involving

shifted Legendre polynomials. This inversion process is based on a method reported by Zakian and Littlewood (1973).

We seek the value of $f(T)$ for a specific value $t = T$.

We choose the set of transform parameters

$$\lambda_i = \frac{k+1}{\tau} \qquad k = 0, 1, \ldots, M'$$

and first obtain the constants a_{kj} given by Aral and Gülçat

$$a_{kj} = (-1)^{k+j} \begin{pmatrix} k+j \\ k \end{pmatrix} \begin{pmatrix} k \\ j \end{pmatrix} \qquad 0 \leq j \leq k : k = 0, 1, \ldots, M'$$

then evaluate the shifted Legendre polynomial of degree k.

$$P_k^*(z) = a_{k0} + a_{k1}z + a_{k2}z^2 + \ldots + a_{kk}z^k \qquad \text{with } z = \mathrm{e}^{-T/\tau}$$

We then evaluate the weights C_k

$$C_k = (2k+1) \sum_{i=0}^{k} a_{ki} \bar{f}(\lambda_i)$$

Finally the solution is obtained in the form

$$f(T) \approx \frac{1}{\tau} \sum_{k=0}^{M'} C_k P_k^*(z)$$

Aral and Gülçat discuss the possible choices for the value of the arbitrary parameter, τ. They suggest using $\tau = T$, however we find that using $\tau = 1.0$ gives as good overall results as other values. We tested smaller and larger τ values but for our examples over our times, the changes made little difference to the tracking of the solution.

The choice of M' is made by the user. As M' approaches infinity the truncation error becomes zero but as M' increases, the magnitude of the coefficients a_{kj} increases, hence round-off errors in the computed value of C_k increase. This is a characteristic instability attributed to inverse Laplace transforms which we have already noted with Stehfest's transform parameter. Aral and Gülçat suggest the use of $M' = 12$ for the transform parameter

but we shall test a variety of values for M' to compare with similar values for Stehfest's method.

5.3.3 Examples of the inversion methods

A variety of test cases of Laplace transforms and their inversions have been tested (Crann 1996) and we consider here five examples using Stehfest's inversion method with parameter $M = 6, 8, 10, 12$ and 14 and the shifted Legendre polynomials technique with parameter $M' = 6, 8, 10, 12$ and 14. We compare the numerical results with the analytic values.

Example 5.1

This example is the Laplace transform and its inverted function

$$\bar{f}(\lambda) = \frac{1}{\lambda + 1}, \quad f(t) = e^{-t}$$

We see from Figure 5.1 and Table 5.2 that, using Stehfest's inversion method, the function has inverted very well and we recover good approximate values.

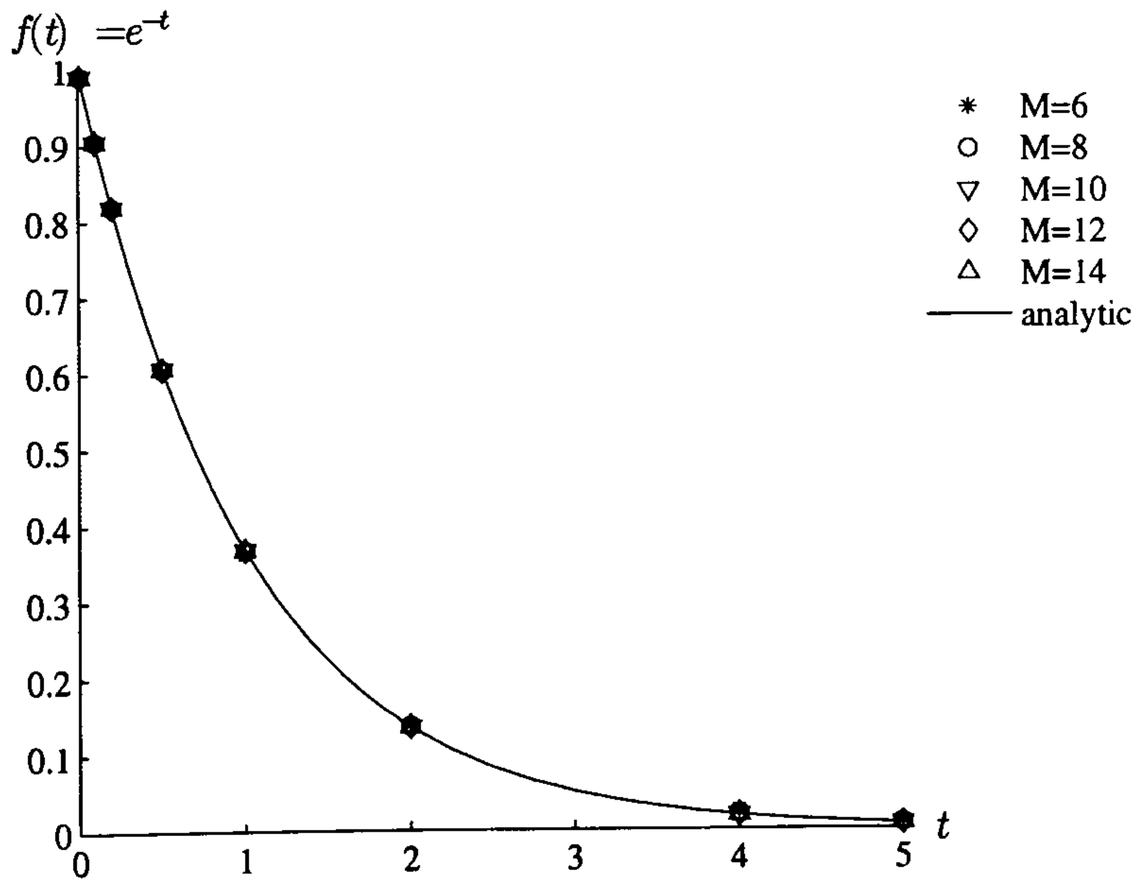

Figure 5.1: The numerical and analytic values of Example 5.1 using Stehfest's method

Table 5.2: Percentage errors for Stehfest's method for Example 5.1

time	$M = 6$	$M = 8$	$M = 10$	$M = 12$	$M = 14$
0.01	2.44E-03	8.42E-05	3.57E-05	1.39E-06	6.31E-07
0.1	2.42E-02	7.27E-04	3.07E-04	2.96E-05	1.32E-06
0.2	8.15E-03	2.72E-03	2.91E-04	7.12E-05	1.60E-06
0.5	5.21E-01	3.87E-02	2.27E-03	1.60E-04	1.48E-05
1.0	1.36E+00	1.96E-01	2.48E-02	2.73E-03	2.59E-04
2.0	2.22E+00	8.17E-01	2.07E-01	4.26E-02	7.51E-03
4.0	3.08E+01	6.67E+00	6.60E-01	1.61E-01	1.10E-01
5.0	4.67E+01	9.95E-01	4.35E+00	2.24E+00	7.41E-01
10.0	3.49E+03	1.99E+03	3.42E+02	1.22E+02	1.06E+02

In Figure 5.2 we show a graph of the analytic and numerical values for Example 5.1 using the SLP inversion method with $\tau = 1.0$ and we see that we recover very good results. In Table 5.3 we show the relative errors and see that the method has inverted very well. Values using $M' = 8$ or $M' = 10$ give the best results but then rounding errors from the very large numerically calculated weights start to take effect.

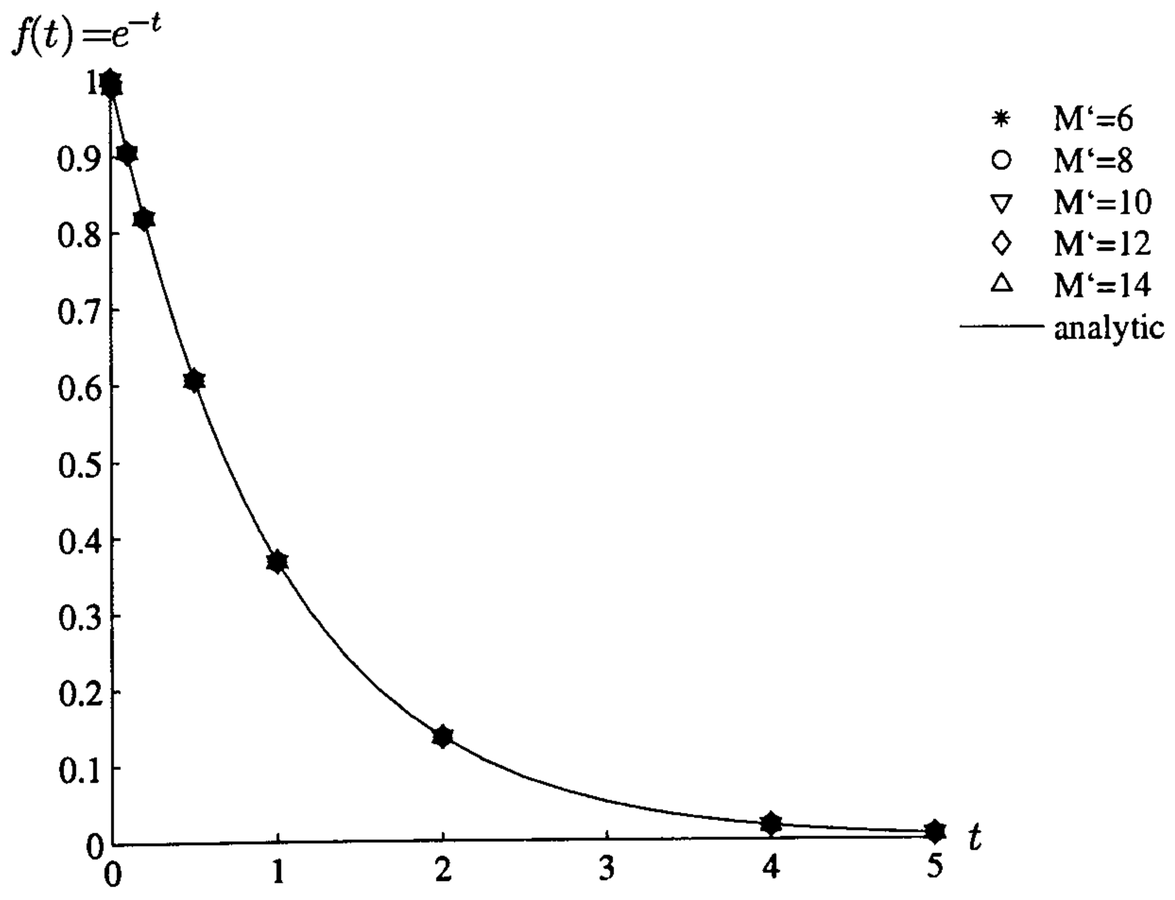

Figure 5.2: The numerical and analytic values of Example 5.1 using the SLP method

Table 5.3: Percentage errors for the SLP method for Example 5.1

time	$M' = 6$	$M' = 8$	$M' = 10$	$M' = 12$	$M' = 14$
0.01	9.09E-05	4.97E-09	4.97E-09	3.54E-08	2.19E-05
0.1	3.97E-09	3.97E-09	3.97E-09	1.81E-07	1.21E-05
0.2	9.52E-09	2.69E-09	2.69E-09	1.61E-07	2.28E-05
0.5	4.74E-08	2.08E-09	2.08E-09	2.45E-07	3.35E-05
1.0	4.66E-08	7.76E-09	7.76E-09	4.43E-07	4.20E-05
2.0	1.75E-07	2.71E-08	2.71E-08	2.71E-08	1.23E-04
4.0	6.07E-07	6.15E-08	6.15E-08	6.15E-08	1.95E-03
5.0	1.36E-08	1.36E-08	1.36E-08	1.36E-08	6.09E-05
10.0	6.56E-05	6.56E-05	6.56E-05	1.64E-02	1.92E+00

Example 5.2

This example is the Laplace transform and its inverted function

$$\bar{f}(\lambda) = \frac{\pi}{\lambda^2 + \pi^2}, \quad f(t) = \sin \pi t$$

The numerical approximations and analytic values are shown in Figure 5.3 for Stehfest's method.

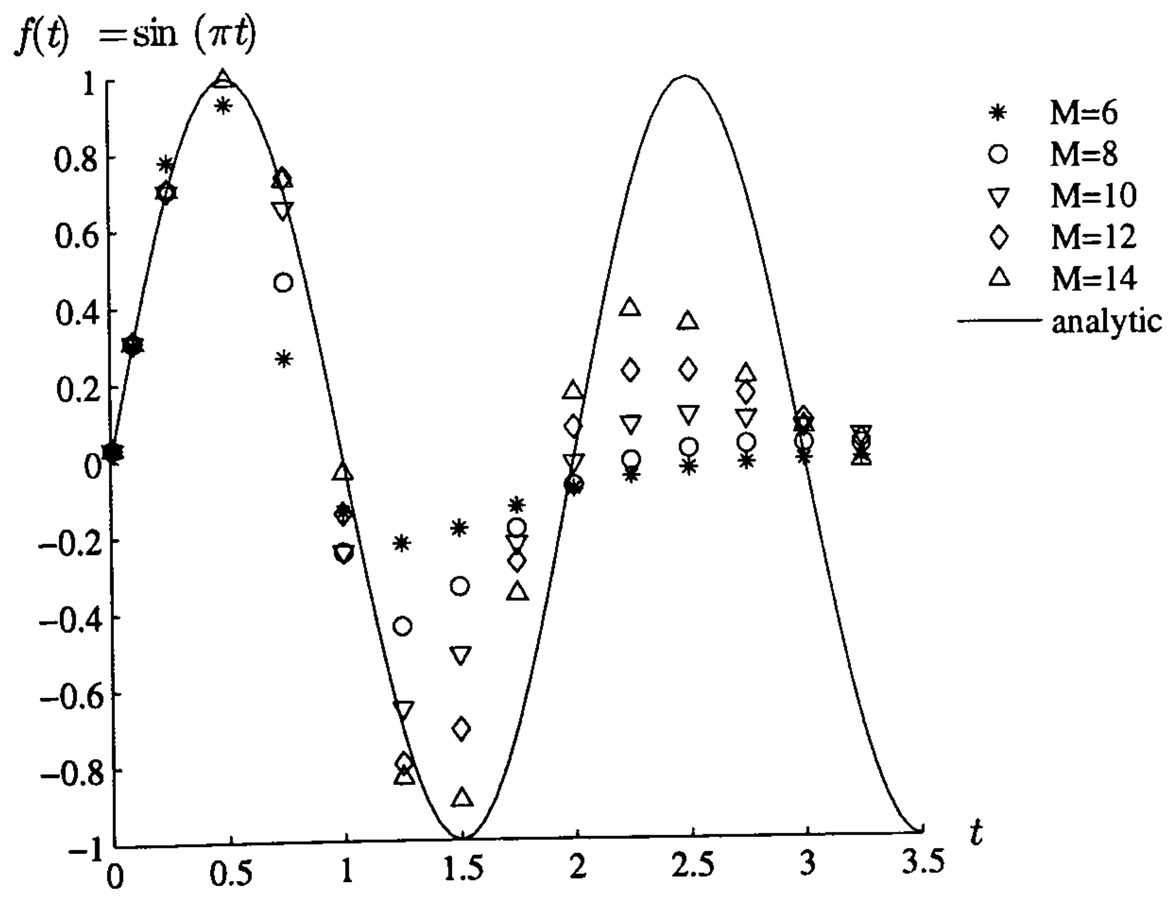

Figure 5.3: The numerical and analytic values of Example 5.2 using Stehfest's method

We see that the inversion approximation does not track the analytic value after $t > 0.5$ for any of the M-values for the Stehfest inversion method. The

function is oscillatory and we would not expect good results, as suggested by Stehfest. The method smooths out the oscillatory nature of the function.

In Figure 5.4, for the SLP method, we see that the numerical approximation tracks the analytic value closer for slightly longer, but eventually smooths out and loses the oscillatory nature of the function.

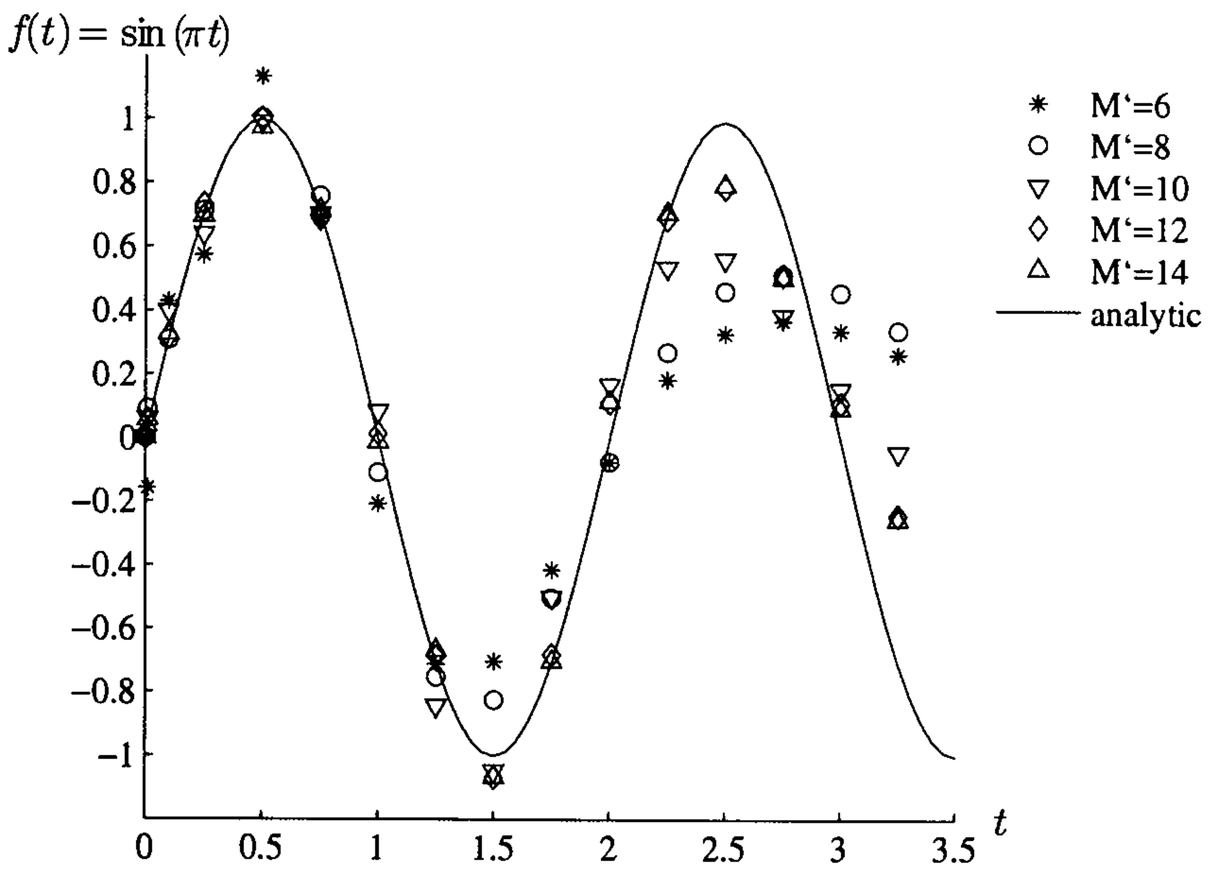

Figure 5.4: The numerical and analytic values of Example 5.2 using the SLP method

Example 5.3

This example is the Laplace transform and its inverted function

$$\bar{f}(\lambda) = \frac{e^{-\lambda}}{\lambda}, \quad f(t) = H(t-1)$$

where H is the Heaviside unit step function defined by

$$H(t-1) = \begin{cases} 0 & t < 1 \\ 1 & t \geq 1 \end{cases}$$

We show the approximate and analytic values in Figure 5.5 for Stehfest's inversion method and in Figure 5.6 for the SLP method.

Again, we see that the approximations do not track the analytic value very well. The inversions have problems with the discontinuity at $t = 1$ and

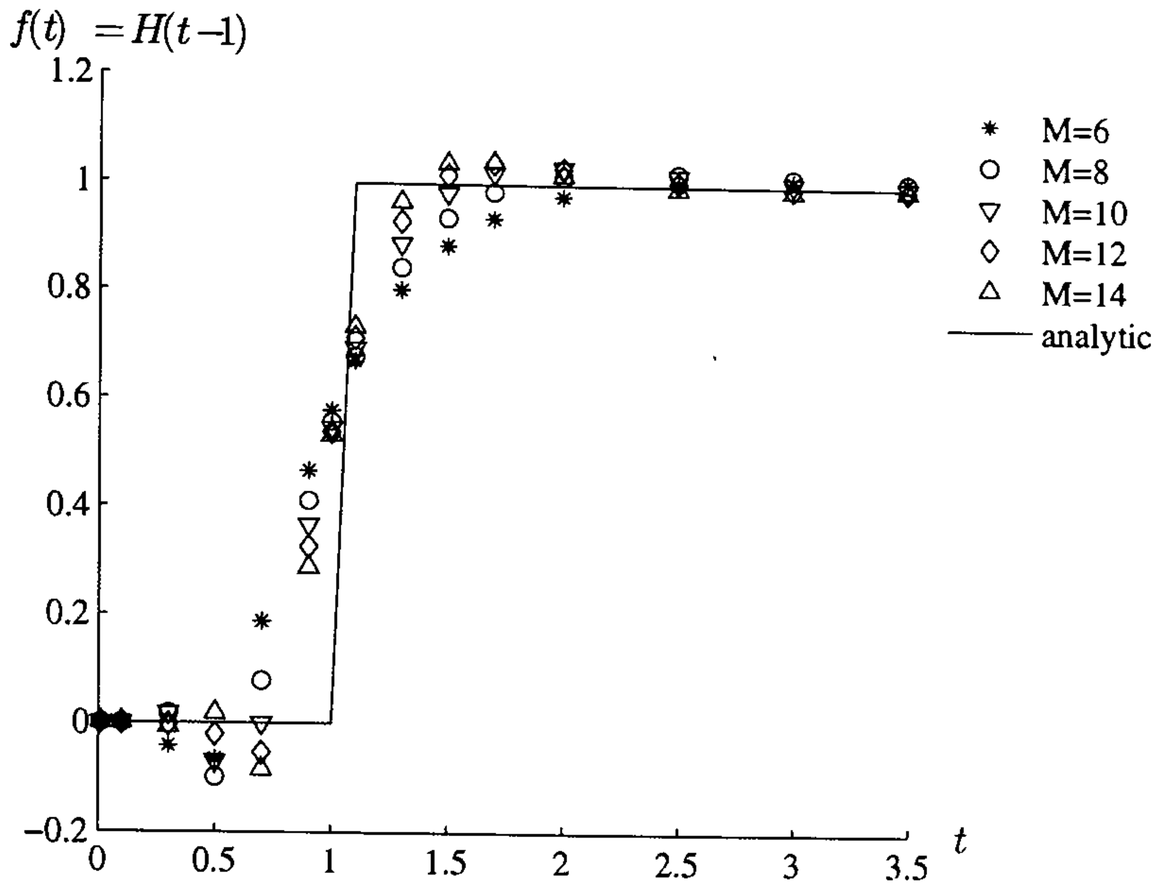

Figure 5.5: The numerical and analytic values of Example 5.3 using Stehfest's method

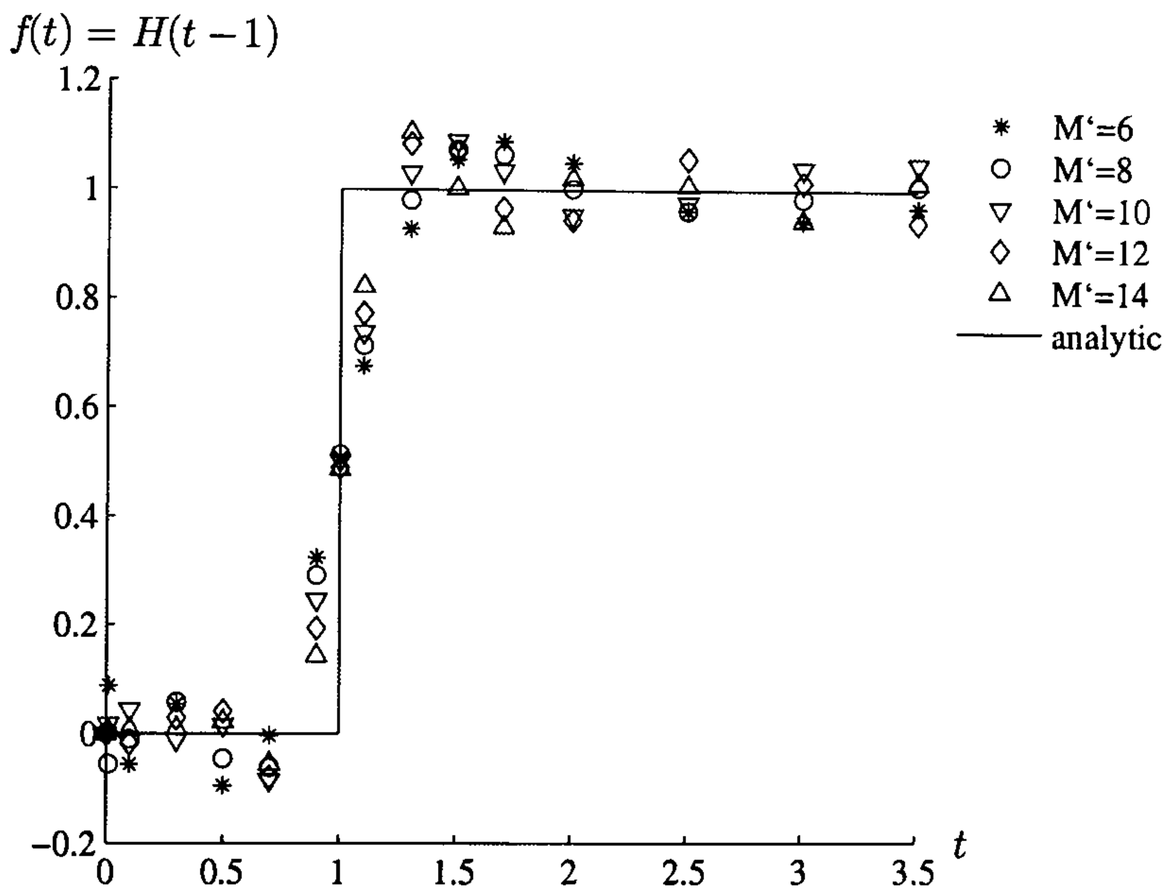

Figure 5.6: The numerical and analytic values of Example 5.3 using the SLP method

smooth out the function. By time $t = 2$, however, the inversions are satisfactory. We can see this in the numerical values in Table 5.4 for Stehfest's method and Table 5.5 for the SLP method.

Table 5.4: Numerical values for Stehfest's method for Example 5.3

time	analytic	$M = 6$	$M = 8$	$M = 10$	$M = 12$	$M = 14$
0.01	0.0	0.000000	0.000000	0.000000	0.000000	0.000000
0.1	0.0	0.000953	-0.000303	0.000066	-0.000009	0.000000
0.5	0.0	-0.065674	-0.099365	-0.070222	-0.020988	0.018487
0.9	0.0	0.465996	0.410959	0.365807	0.325568	0.288468
1.0	1.0	0.578125	0.557292	0.545492	0.537777	0.532315
1.1	1.0	0.670074	0.677453	0.693290	0.712680	0.733761
2.0	1.0	0.978445	1.016469	1.032599	1.030503	1.018024
3.0	1.0	1.012089	1.017257	1.007359	0.996731	0.992859
4.0	1.0	1.011121	1.006560	0.998203	0.995722	0.998288

Table 5.5: Numerical values for the SLP method for Example 5.3

time	analytic	$M' = 6$	$M' = 8$	$M' = 10$	$M' = 12$	$M' = 14$
0.01	0.0	0.087730	-0.055888	0.016950	0.001892	0.005480
0.1	0.0	-0.056353	-0.010093	0.043094	-0.018058	0.007989
0.5	0.0	-0.095115	-0.045845	0.015725	0.041129	0.022081
0.9	0.0	0.323143	0.290729	0.245495	0.193223	0.142926
1.0	1.0	0.507328	0.513388	0.501796	0.488717	0.486258
1.1	1.0	0.676341	0.713982	0.738302	0.773425	0.822785
2.0	1.0	1.050473	1.002588	0.953891	0.944916	1.021506
3.0	1.0	0.942677	0.984848	1.040608	1.014422	0.944189
4.0	1.0	1.005948	1.019276	1.020761	0.935713	1.067499

Example 5.4

This example is the Laplace transform and its associated inverted function

$$\bar{f}(\lambda) = \exp(-\sqrt{\lambda}), \quad f(t) = \frac{1}{2\sqrt{\pi t^3}} \exp\left(-\frac{1}{4t}\right)$$

and we show the values in Figure 5.7 and 5.8.

We can see that the numerical values are very good approximations for all values of M and M'. We show the percentage errors for these values in Tables 5.6 and 5.7. We notice that for small values of t none of the M and M' recovers a very good result, and we have found that this can often be a difficulty for these Laplace transform methods. However, for $t \geq 0.1$ we see that the approximations are very good, and for $M > 6$ for Stehfest's

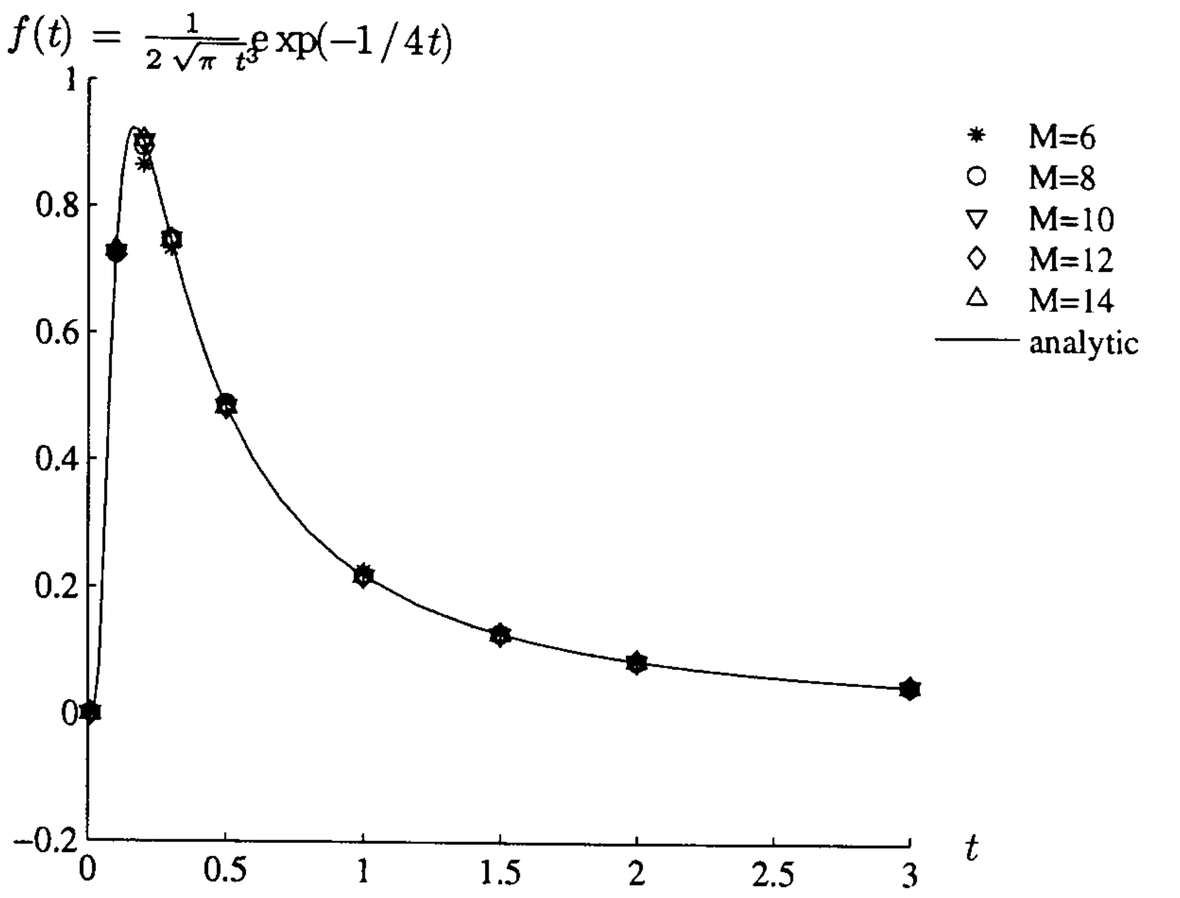

Figure 5.7: The numerical and analytic values of Example 5.4 using Stehfest's method

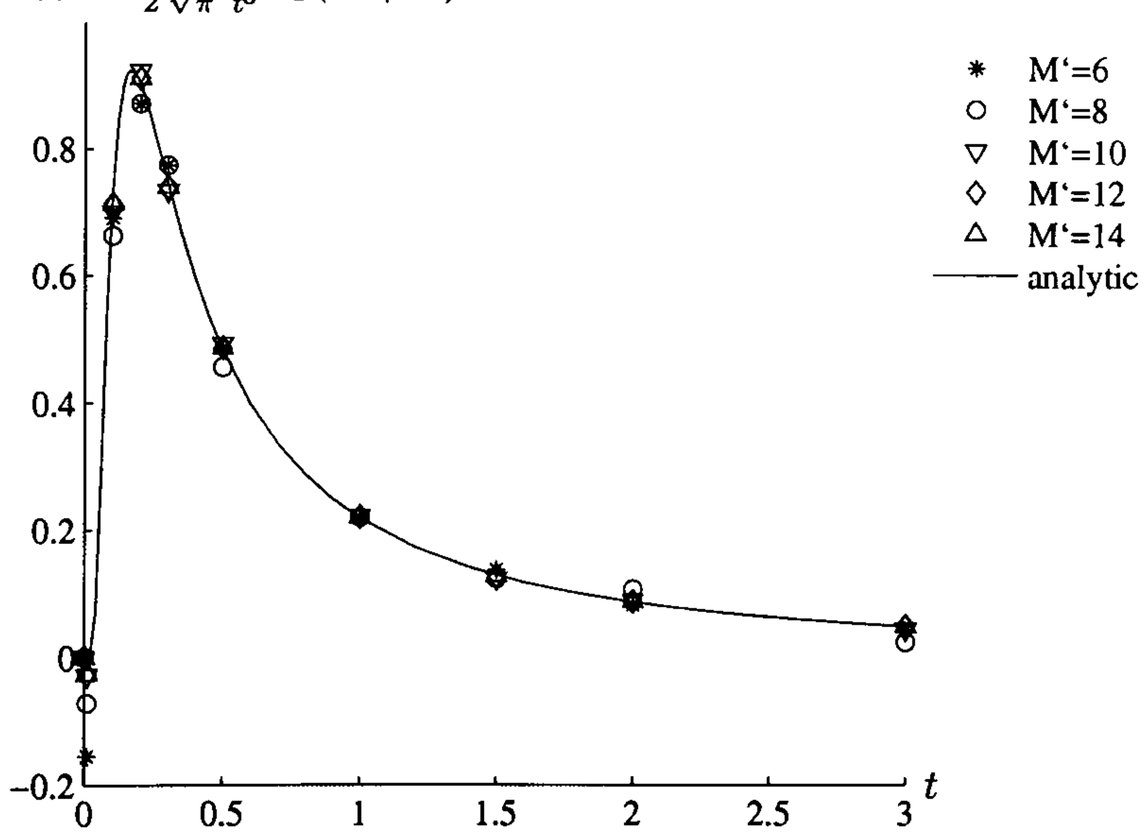

Figure 5.8: The numerical and analytic values of Example 5.4 using the SLP method

method the maximum error is one percent.

Table 5.6: Percentage errors for Stehfest's method for Example 5.4

time	$M = 6$	$M = 8$	$M = 10$	$M = 12$	$M = 14$
0.01	3.67E+07	3.91E+07	3.47E+06	2.01E+06	9.31E+05
0.1	7.78E-01	1.04E+00	6.90E-01	3.38E-01	1.29E-01
0.2	4.12E+00	9.98E-01	2.54E-02	1.25E-01	7.01E-02
0.3	1.65E+00	3.85E-01	4.14E-01	1.45E-01	9.56E-03
0.5	1.62E+00	1.16E+00	2.47E-01	5.41E-02	4.73E-02
1.0	3.27E+00	3.47E-01	2.84E-01	9.63E-02	1.31E-02
1.5	2.63E+00	3.86E-01	3.36E-01	4.18E-03	3.30E-02
2.0	1.73E+00	7.69E-01	2.53E-01	5.38E-02	2.72E-02
3.0	1.93E-01	1.00E+00	5.86E-02	8.88E-02	5.73E-03

Table 5.7: Percentage errors for the SLP method for Example 5.4

time	$M' = 6$	$M' = 8$	$M' = 10$	$M' = 12$	$M' = 14$
0.01	3.96E+09	1.83E+09	7.20E+08	6.70E+08	6.70E+08
0.1	5.57E+00	9.42E+00	4.37E+00	2.52E+00	2.52E+00
0.2	3.48E+00	3.48E+00	2.25E+00	9.40E-01	9.40E-01
0.3	3.86E+00	3.87E+00	1.56E+00	7.08E-01	7.08E-01
0.5	6.42E-01	5.56E+00	2.12E+00	6.23E-01	6.23E-01
1.0	3.38E-01	2.75E-01	1.16E+00	7.62E-01	7.62E-01
1.5	6.22E+00	4.09E+00	4.53E+00	3.18E-01	3.18E-01
2.0	5.77E+00	2.29E+01	1.93E+00	7.05E-01	7.05E-01
3.0	1.23E+01	4.93E+01	9.36E+00	7.26E-01	7.26E-01

Example 5.5

We now consider an example where, in order to take the Laplace transform, we need to take a series expansion of the function. Let

$$f(t) = \exp(-e^{-t})$$

and expand it as a series

$$f(t) = 1 - e^{-t} + \frac{e^{-2t}}{2!} - \frac{e^{-3t}}{3!} + \cdots$$

so that when we take the Laplace transform we obtain

$$\mathcal{L}[f(t)] = \frac{1}{\lambda} - \frac{1}{\lambda+1} + \frac{1}{2!(\lambda+2)} - \frac{1}{3!(\lambda+3)} + \cdots$$

We use both Stehfest and the SLP inversion techniques and show the percentage errors in Tables 5.8 and 5.9 for the truncated series.

Table 5.8: Percentage errors for Example 5.5 using Stehfest's method, $M = 8$, on the series truncated after the number of terms

time	2 terms	3 terms	4 terms	5 terms	6 terms
0.01	97.32	34.58	8.95	1.83	0.31
0.1	76.48	24.70	5.82	1.08	0.17
0.2	58.90	17.09	3.64	0.60	0.09
0.3	45.63	11.89	2.30	0.33	0.06
0.5	27.79	5.88	0.94	0.10	0.03
1.0	8.58	1.28	0.03	0.15	0.14
1.5	2.89	0.35	0.11	0.11	0.12
2.0	1.13	0.01	0.04	0.05	0.05
3.0	0.34	0.25	0.23	0.24	0.24

Table 5.9: Percentage errors for Example 5.5 using the SLP method, $M' = 8$, on the series truncated after the number of terms

time	2 terms	3 terms	4 terms	5 terms	6 terms
0.01	97.32	34.58	8.95	1.83	0.31
0.1	76.48	24.70	5.82	1.08	0.17
0.2	58.90	17.11	3.64	0.61	0.08
0.3	45.63	11.93	2.29	0.35	0.04
0.5	27.84	5.90	0.92	0.11	0.01
1.0	8.68	1.10	0.10	0.01	0.00
1.5	2.89	0.22	0.01	0.00	0.00
2.0	1.00	0.05	0.00	0.00	0.00
3.0	0.13	0.00	0.00	0.00	0.00

As we have often found before, the approximations for small values of time have the highest errors. However by the fifth term in the series the approximations are very good for both inversion methods. In Chapter 9 we shall use this process of approximating a function by a suitable series to effect a Laplace transform.

We see from these examples that both inversion processes recover the value of the transform very well and are straightforward to use. The methods were tested in a parallel environment (Crann, Davies and Mushtaq 1998) and computation times for the two algorithms were very similar. Since there is little to choose between the accuracy and computation time of the two methods, we shall choose Stehfest's method since it is slightly easier

to implement. For the choice of inversion parameter, our results in these test examples confirm the suggestions of other researchers mentioned in Section 5.3.1. Consequently we shall choose a value $M = 8$ for use in our applications.

5.4 The Laplace transform method for ordinary differential equations

In this section we consider initial-value problems associated with ordinary differential equations (Davies and Crann 1999). This will give us an indication of how to treat the time variable in diffusion-type problems described by a parabolic partial differential equation.

Example 5.6

This problem is defined as

$$\frac{d^2x}{dt^2} + 2\frac{dx}{dt} + 5x = e^{-t}\sin t$$

with initial conditions

$$x(0) = 0 \quad \text{and} \quad \frac{dx}{dt}(0) = 1$$

Taking the Lapace transform we obtain

$$\lambda^2\bar{x} - \lambda x(0) - \frac{dx}{dt}(0) + 2\left(\lambda\bar{x} - x(0)\right) + 5\bar{x} = \frac{1}{(\lambda+1)^2+1}$$

Rearranging and simplifying, this becomes

$$\bar{x} = \frac{1}{\lambda^2+2\lambda+5} + \frac{1}{(\lambda^2+2\lambda+5)(\lambda^2+2\lambda+2)}$$

This transform can be inverted using partial fractions and a set of tables (Davies and Crann 2004) as

$$x(t) = \frac{1}{3}e^{-t}(\sin t + \sin 2t)$$

The solution is oscillatory due to the sinusoidal terms. However the exponential term dominates the function $x(t)$ and has an amplitude less than 0.003 by $t = 3.0$. Using Stehfest's inversion method with parameter $M = 8$ we show the numerical and analytic solution in Figure 5.9 and we see that the approximation tracks the analytic solution in a satisfactory manner.

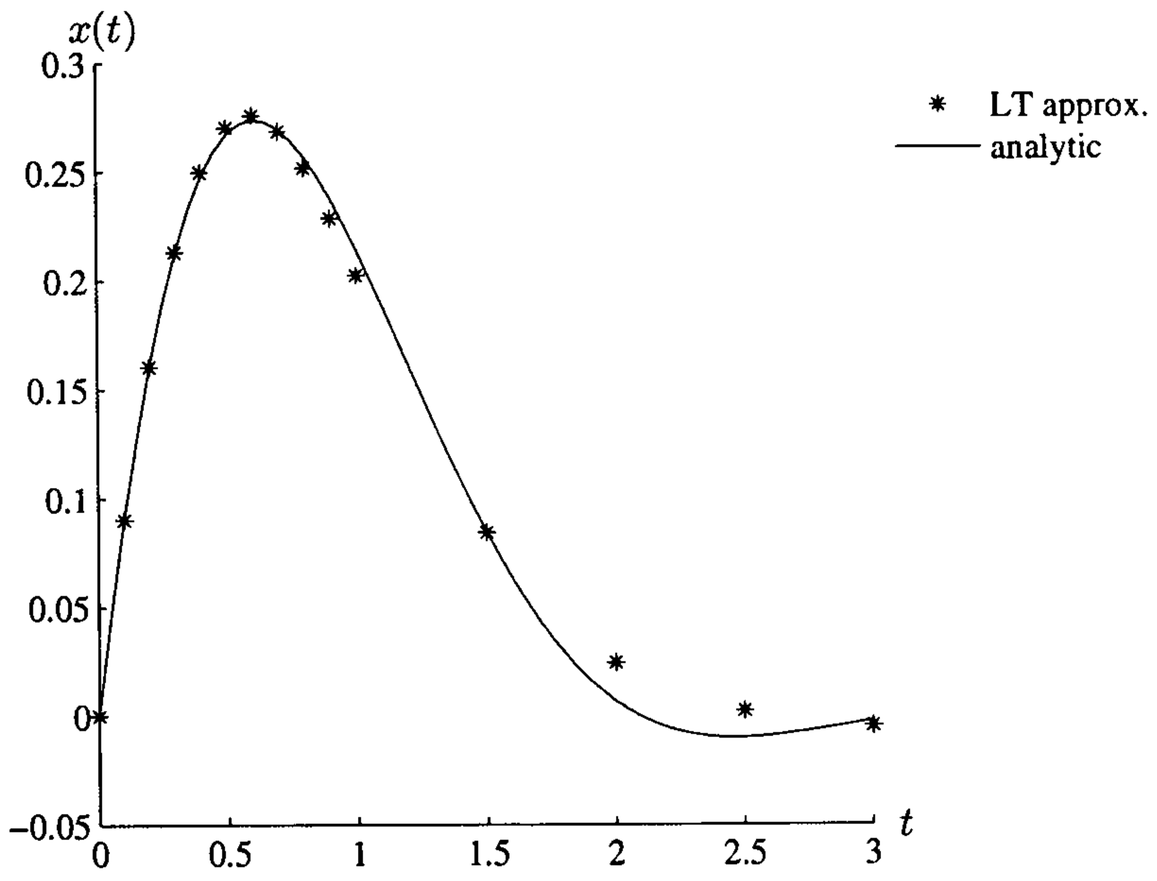

Figure 5.9: The numerical and analytic solution of Example 5.6 using Stehfest's inversion method

In Table 5.10 we present the analytic and numerical results together with the percentage errors for Stehfest's method. We see that the errors are quite small but the approximation fails to pick up the oscillatory nature of the solution.

Table 5.10: Numerical results for Example 5.6 using Stehfest's inversion method

time	analytic	approximation	error
0.1	0.090032	0.089965	7.51E-02
0.2	0.160495	0.160410	5.29E-02
0.3	0.212408	0.213237	3.90E-01
0.4	0.247298	0.249709	9.75E-01
0.5	0.267055	0.270244	1.19E+00
0.6	0.273799	0.275832	7.43E-01
0.7	0.269756	0.268692	3.95E-01
0.8	0.257155	0.251953	2.02E+00
0.9	0.238138	0.228967	3.85E+00
1.0	0.214691	0.202753	5.56E+00
1.5	0.084686	0.084291	4.67E-01
2.0	0.006879	0.024604	2.58E+02
2.5	-0.009863	0.002392	-1.24E+02
3.0	-0.002295	-0.004484	-9.54E+01

Example 5.7

In this example we consider a problem with a discontinuity in the data (Davies and Crann 1999):

$$\frac{dx}{dt} + x = H(t-1) \qquad x(0) = 1$$

where $H(t-1)$ is the Heaviside unit step function.

Taking the Laplace transform we obtain

$$(\lambda + 1)\bar{x} = 1 + \frac{e^{-\lambda}}{\lambda}$$

We call this the *Full Laplace transform method* (Full LT).

The analytic solution is

$$x(t) = e^{-t} + H(t-1)(1 - e^{1-t})$$

The numerical solution is compared with the analytic solution in Figure 5.10.

We notice that, as suggested by Stehfest, the numerical solution does not compare well with the analytic solution in the neighbourhood of the

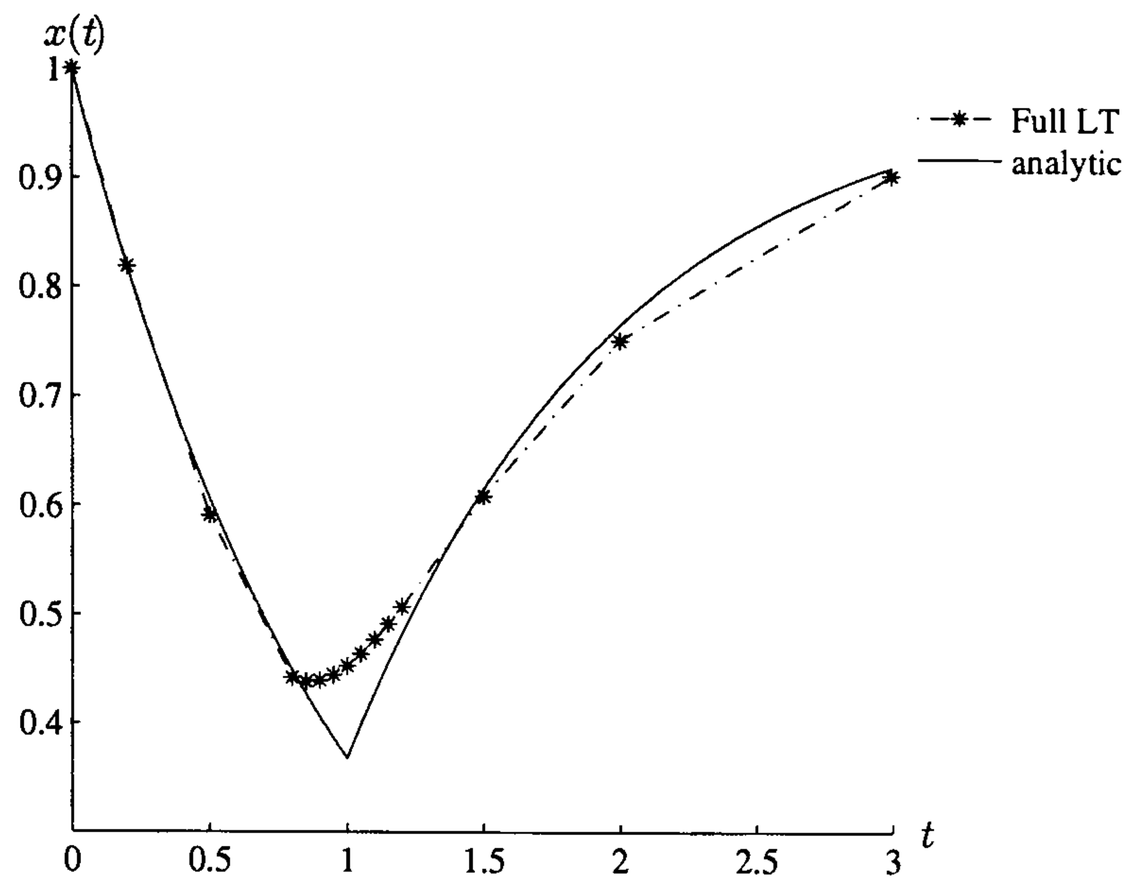

Figure 5.10: The numerical and analytic solution of Example 5.7 using the Full LT method

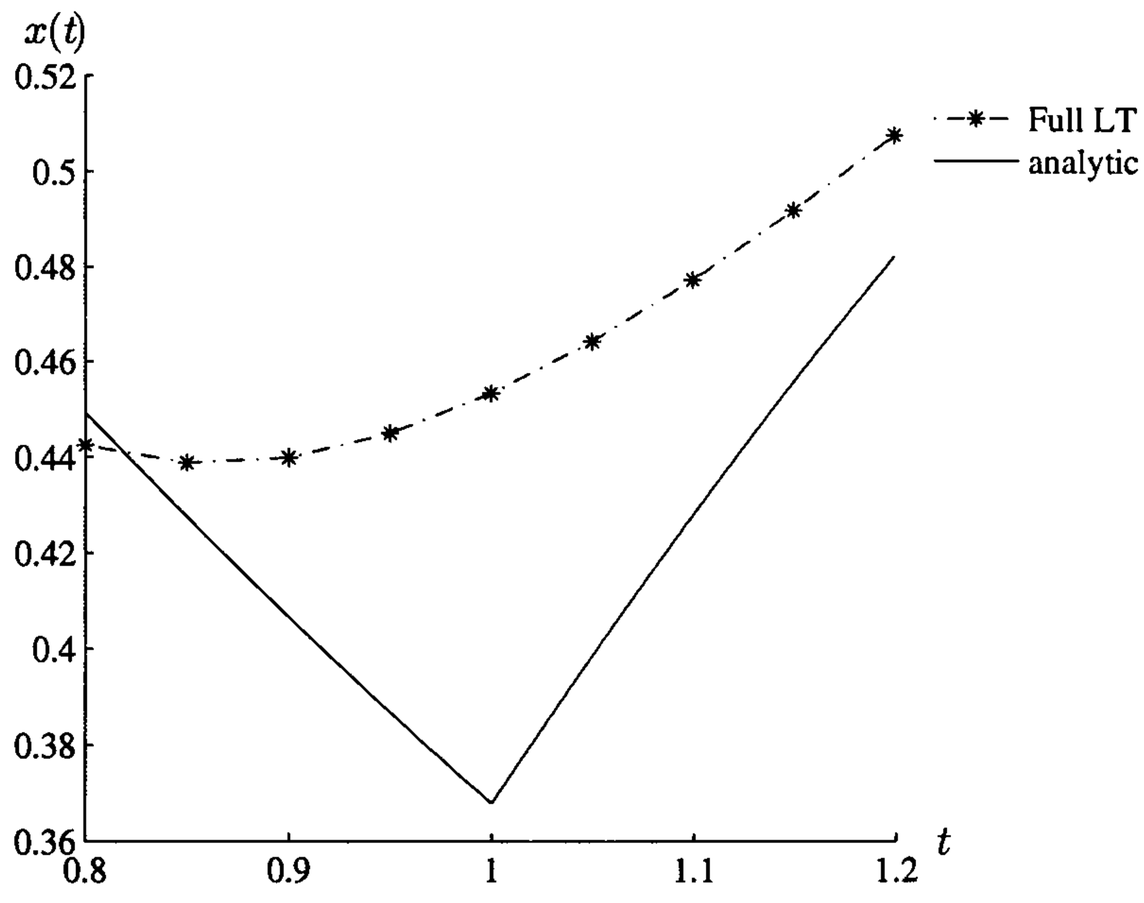

Figure 5.11: The numerical and analytic solution of Example 5.7 using the Full LT method, detail of region near $t = 1.0$

discontinuity at $t = 1$, see Figure 5.11. To overcome this we use the Laplace transform method to solve the equation

$$\frac{dx}{dt} + x = 0$$

subject to $x(0) = 1$ and obtain the value of $x(1)$. We make the change of

variable

$$x(t) = x_1(t-1) \quad t > 1$$

and then solve the equation

$$\frac{dx_1}{dt} + x_1 = 1$$

subject to $x_1(0) = x(1)$.

So we have

$$(\lambda + 1)\bar{x} = 1 \text{ to obtain } x(T) \text{ for } T \leq 1$$

and

$$(\lambda + 1)\bar{x}_1 = \frac{1}{\lambda} + x(1) \text{ to obtain } x(T) \text{ for } T > 1$$

We call this the *Step Laplace transform method* (Step LT).

The numerical and analytic solutions for the Step LT method are compared in Figures 5.12 and 5.13 in which we see that the numerical solution compares very well with the analytic solution. We shall use this idea again in Chapter 8 with discontinuities in the boundary conditions of partial differential equations.

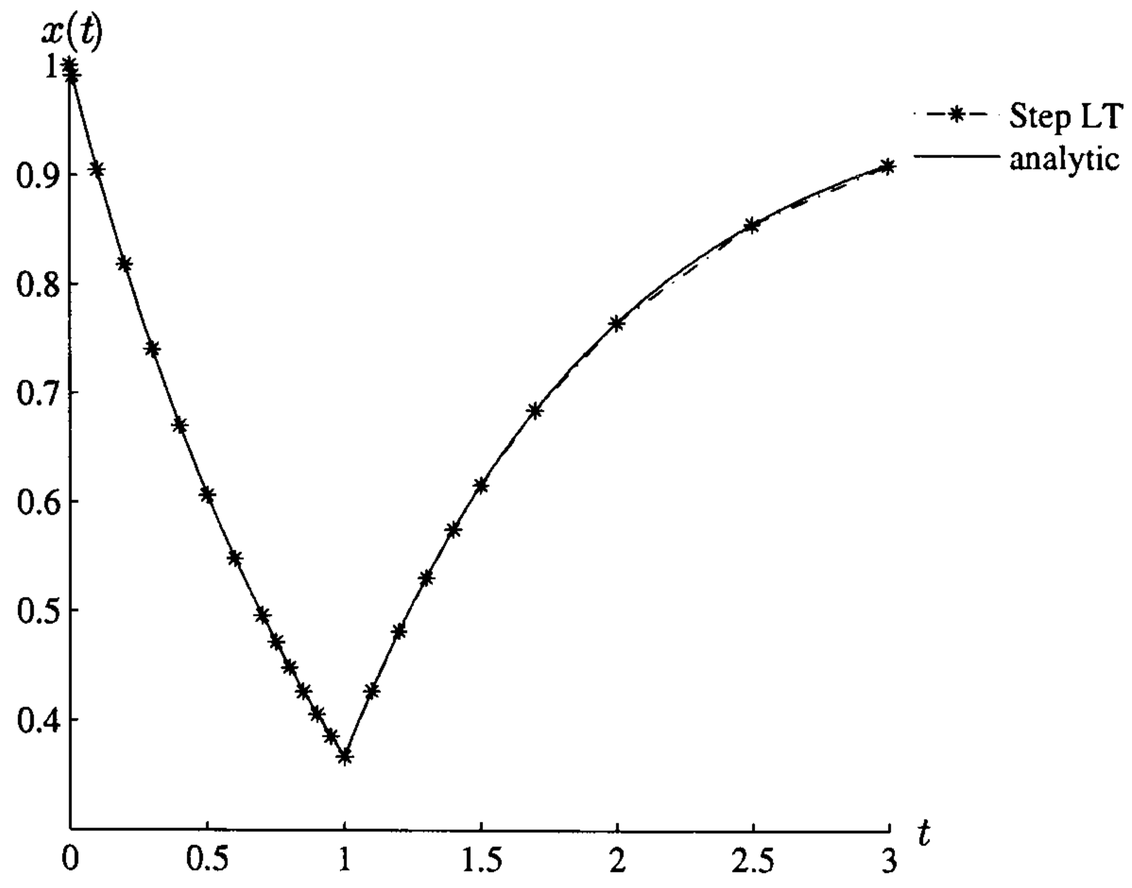

Figure 5.12: The numerical and analytic solution of Example 5.7 using the Step LT method

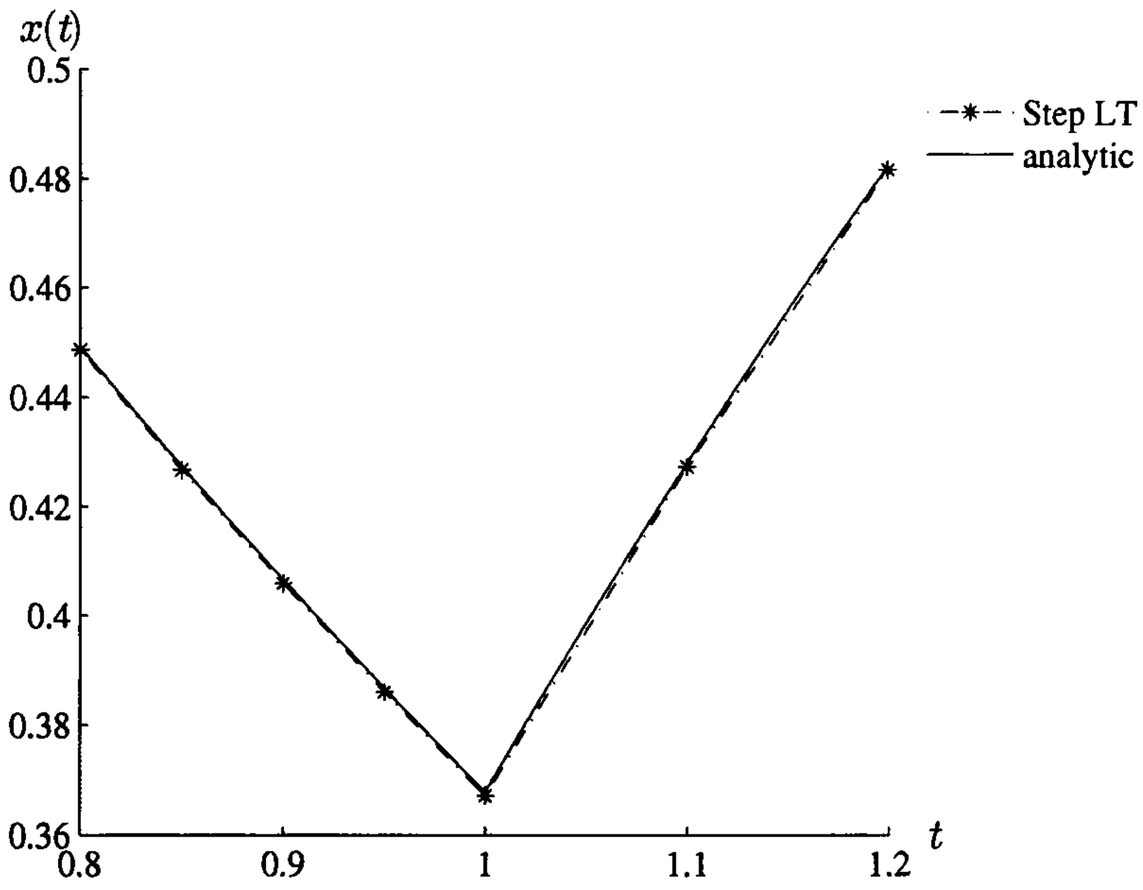

Figure 5.13: The numerical and analytic solution of Example 5.7 using the Step LT method, detail of region near $t = 1.0$

5.5 The Laplace transform method for parabolic problems

We shall describe the process with reference to an initial boundary-value problem defined in the two-dimensional region D bounded by the closed

curve $C = C_1 + C_2$, such as in Figure 3.1,

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \qquad \text{in } D \tag{5.6}$$

subject to the boundary conditions

$$u = u_1 \qquad \text{on } C_1 \tag{5.7}$$

$$q \equiv \frac{\partial u}{\partial n} = q_2 \qquad \text{on } C_2 \tag{5.8}$$

and the initial condition

$$u(\mathbf{r}, 0) = u_0(\mathbf{r}) \qquad \text{in } D \tag{5.9}$$

We now define the Laplace transform in time by

$$\bar{u}(\mathbf{r}; \lambda) = \int_0^\infty u(\mathbf{r}, t) e^{-\lambda t} dt \tag{5.10}$$

so that the initial boundary-value problem (5.6), (5.7), (5.8) and (5.9) becomes

$$\nabla^2 \bar{u} = \frac{1}{\alpha}(\lambda \bar{u} - u_0) \qquad \text{in } D \tag{5.11}$$

subject to

$$\bar{u} = \bar{u}_1 \qquad \text{on } C_1 \tag{5.12}$$

$$\bar{q} = \bar{q}_2 \qquad \text{on } C_2 \tag{5.13}$$

If $u_0 = 0$ this elliptic problem in the transformed plane comprises the modified Helmholtz equation (5.11) subject to constant Dirichlet and Neumann boundary conditions on C_1 and C_2 respectively and can be solved in a variety of ways, as suggested in Chapter 2, and inverted using one of the numerical inversion methods above. Although in later chapters we shall use the Laplace transform together with the boundary element method, we find it useful to compare solutions of a test problem using a variety of different elliptic equation solvers and this we shall do in Section 6.1.

5.6 Summary of Chapter 5

In this chapter we introduced a variety of ways of using the Laplace transform method and the boundary element method for the solution of time-dependent parabolic problems. The difficulty associated with the method is in the numerical inversion needed to bring the Laplace space solution back to the time domain and we have compared two possible techniques using real parameters rather than complex ones. Stehfest's numerical method is straightforward, easy to compute and gives good results for a variety of test transforms. Zakian and Littlewood's method gives results of similar accuracy but is a little more complicated to implement. Consequently we shall use Stehfest's method in our examples throughout this thesis. We shall use Stehfest's inversion method with $M = 8$ as suggested by Crann (1996) and in agreement with the suggestions of other authors.

In Chapter 6 we shall solve examples of initial boundary-value partial differential equations using the Laplace transform to reduce the time variable and solve the resulting elliptic problem by a variety of methods sequentially and in parallel.

Chapter 6

Using the Laplace Transform Method

6.1 Introduction

In Chapter 5 we described the Laplace transform method and how it can be used to solve time-dependent initial-value problems. In this chapter we demonstrate the method for time-dependent partial differential equations which have boundary and initial conditions. We also show that the method is ideally suited for use on parallel computers.

Example 6.1

We illustrate the solution process by solving the following two-dimensional heat conduction problem in the square $0 < x < 1$, $0 < y < 1$

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \tag{6.1}$$

subject to the boundary conditions

$$u(x, 0, t) = u(x, 1, t) = 20 \tag{6.2}$$

$$q(0, y, t) = q(1, y, t) = 0 \tag{6.3}$$

and the initial condition

$$u(x, y, 0) = 0 \tag{6.4}$$

see Figure 6.1.

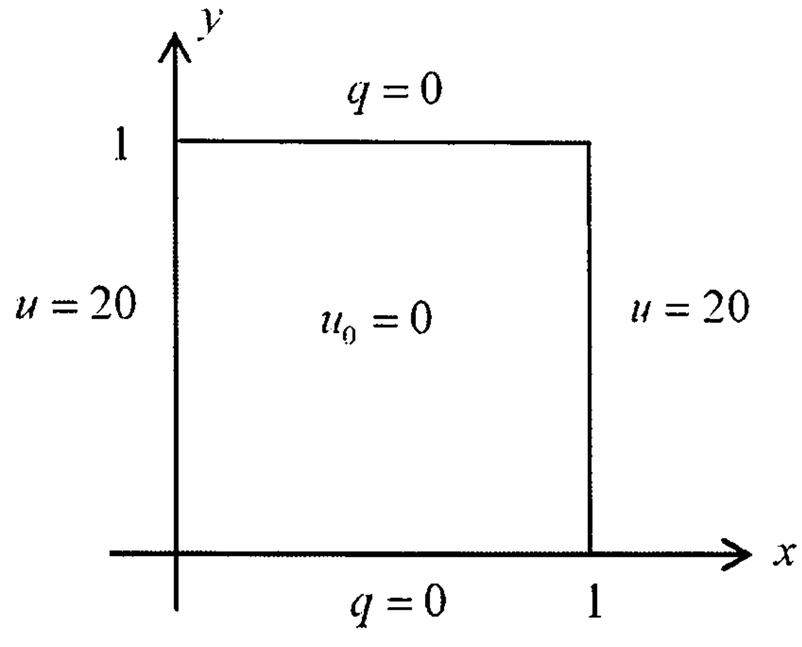

Figure 6.1: Boundary and initial conditions for Example 6.1

The problem is, in fact, essentially dependent only on x and the analytic solution is

$$u(x, y, t) = u(0, y, t) + (u_R - u_L)\frac{x}{l}$$

$$+ \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi x}{l}\right) \exp\left(-\alpha n^2 \pi^2 t / l^2\right)$$

where $u_L = u(x_L, y, t)$, $u_R = u(x_R, y, t)$, $l = x_R - x_L$ and

$$b_n = \frac{2}{n\pi}\left\{(u_0 - u_L)\left(1 - (-1)^n\right) + (u_R - u_L)\left(-1\right)^n\right\}$$

In this case $x_L = 0$ and $x_R = 1$.

Let $\bar{u}(x, y; \lambda)$ be the Laplace transform of $u(x, y, t)$.

The transform of equation (6.1) becomes

$$\mathcal{L}\left[\nabla^2 u\right] = \frac{1}{\alpha}\mathcal{L}\left[\frac{\partial u}{\partial t}\right]$$

so that

$$\nabla^2 \bar{u} = \frac{1}{\alpha}\left(\lambda \bar{u} - u(x, y, 0)\right)$$

Hence

$$\nabla^2 \bar{u} = \frac{\lambda}{\alpha}\bar{u}, \quad since u_0 = 0 \tag{6.5}$$

subject to the boundary conditions

$$\bar{u}(x,0;\lambda) = \bar{u}(x,1;\lambda) = 20/\lambda \tag{6.6}$$

and

$$\bar{q}(0,y;\lambda) = \bar{q}(1,y;\lambda) = 0 \tag{6.7}$$

In Laplace space, equations (6.1), (6.2), (6.3) and (6.4) become the transformed equations (6.5), (6.6) and (6.7) and we solve them in a variety of ways. We report a comparison of the results in Section 6.5.

6.2 Laplace transform finite difference method

We can define a uniform grid on the square $(x,y) : 0 < x < 1, 0 < y < 1$ and use the usual five-point formula for the Laplacian (Smith 1978), see Section 2.2.1

$$\bar{U}_{i,k} = \left(\bar{U}_{i-1,k} + \bar{U}_{i,k-1} + \bar{U}_{i+1,k} + \bar{U}_{i,k+1}\right) / \left(4 + \lambda_j h^2/\alpha\right)$$

We solve this equation for $\alpha = 0.1$ and mesh-size $h = 0.1$. We then use Stehfest's inversion procedure with parameter $M = 8$ to obtain the approximate solutions to the original problem stated in (6.1), (6.2), (6.3) and (6.4) and this numerical solution is compared with the analytic solution in Section 6.5.

6.3 Laplace transform finite element method

A graded mesh of 200 right-angled linear triangles is used to set the finite element equations in the form, (Davies 1985),

$$\mathbf{K}\bar{\mathbf{U}} - \frac{\lambda_i}{\alpha}\mathbf{M}\bar{\mathbf{U}} = \mathbf{f}$$

Again we use $M = 8$ in Stehfest's inversion process.

6.4 Laplace transform boundary element method

The partial differential equation (6.5) in the transform plane is the modified Helmholtz equation, which we can write in the form

$$\nabla^2 \bar{u} = p^2 \bar{u} \tag{6.8}$$

where $p^2 = \frac{\lambda}{\alpha}$.

Since

$$\mathcal{F}[\bar{u}] = 0$$

where

$$\mathcal{F} \equiv \nabla^2 - p^2$$

we can use equation (3.5) with fundamental solution (Kythe 1996)

$$\bar{u}^* = \frac{1}{2\pi} K_0(pR)$$

where R is the distance of the field point from the source point. K_i is the modified Bessel function of the second kind. The corresponding flux function, q^*, is given by

$$\bar{q}^* = \frac{\partial \bar{u}^*}{\partial n}$$

Since $\frac{d}{dx}\left(K_0(x)\right) = -K_1(x)$ (Abramowitz and Stegun 1972) it follows that

$$\bar{q}^* = -\frac{1}{2\pi} p K_1(pR) \frac{1}{R} \mathbf{R.n}$$

The boundary integral equation for the modified Helmholtz equation (6.5) is therefore given by

$$c_P \bar{u}_P = \frac{1}{2\pi} \oint \left(K_0(pR)\bar{q} + \bar{u} p K_1(pR) \frac{1}{R} \mathbf{R.n} \right) ds$$

The boundary element method is applied in the usual manner to set up the system of equations of the form

$$\mathbf{H}\bar{\mathbf{U}} + \mathbf{G}\bar{\mathbf{Q}} = \mathbf{0}$$

whose solution yields approximate values of $\bar{U}$ and $\bar{Q}$ at N nodes on the boundary.

We use the Laplace transform boundary element method to solve Example 6.1 for $\alpha = 0.1$ in the square discretised into 32 linear elements with eight-point Gauss quadrature and using Telles transformation method for the singular integrals. We once again use the inversion method of Stehfest with parameter $M = 8$. We show these results together with the analytic solution for a variety of times in the next section.

6.5 Results of the example using the Laplace transform method

In Figure 6.2 we show the results of Example 6.1 along the line $y = 0.5$ at times $t = 0.1, 0.3, 0.6$ and 1.0 for each of the three methods, FDM, FEM, BEM and the analytic solution. We see that for all methods the approximate solutions track the analytic solution very well.

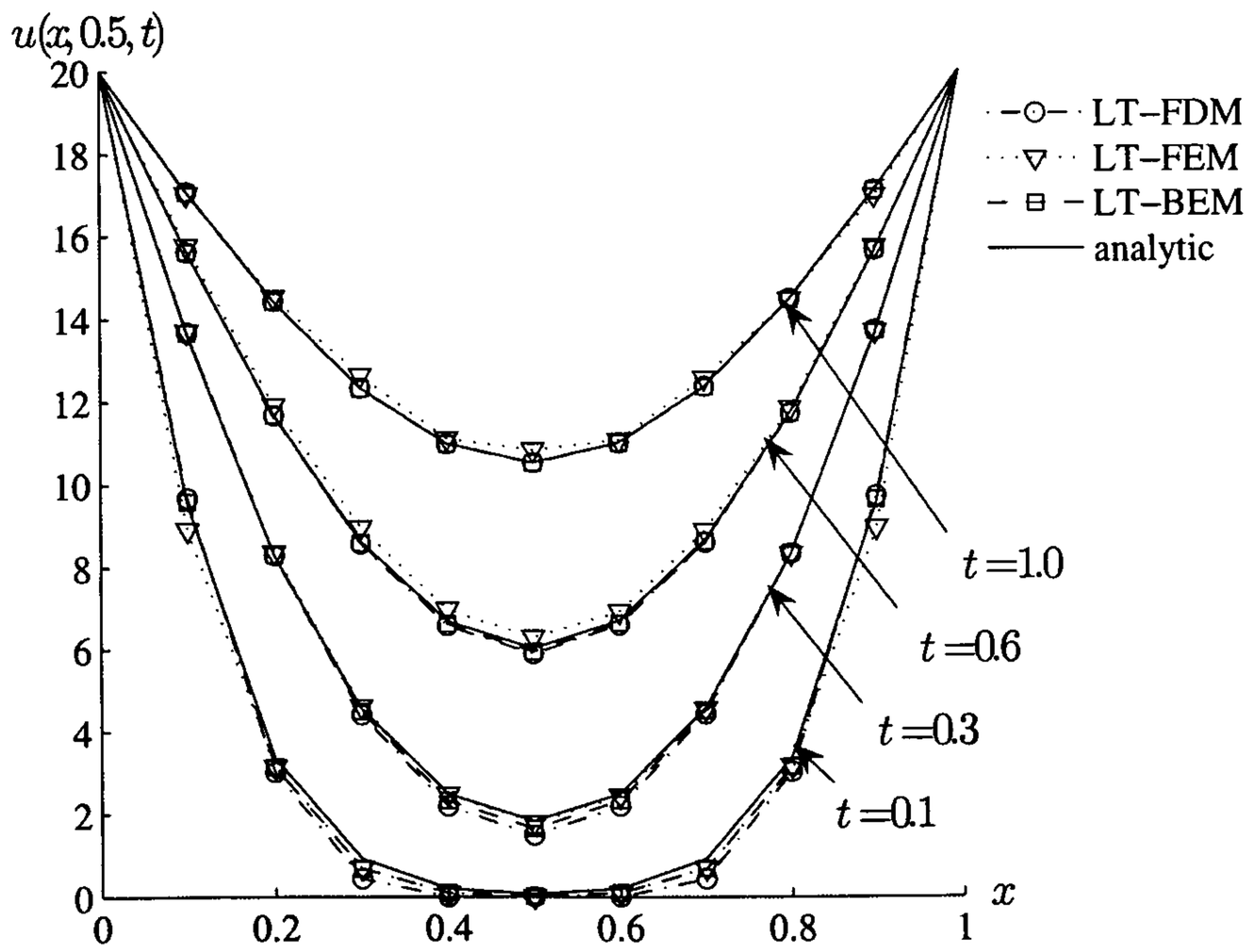

Figure 6.2: Space distribution of the solution for Example 6.1

In Tables 6.1 and 6.2 we see the numerical values of the approximations and the percentage errors from the analytic solution. We see that all three methods are accurate. The surprisingly relatively high errors in the LTBEM are likely to be due to the use of the series for the modified Bessel function, truncated according to Ramesh and Lean, as the fundamental solution.

Table 6.1: Analytic and approximate solutions at $t = 0.6$ for Example 6.1

x-value	Analytic	LTFDM	LTFEM	LTBEM
0.0	20.0000	20.0000	20.0000	20.0000
0.1	15.6142	15.6213	15.6162	15.7656
0.2	11.6818	11.7009	11.6768	11.9249
0.3	8.5922	8.6316	8.5678	8.9637
0.4	6.6284	6.6922	6.5836	6.9780
0.5	5.9560	6.0312	5.9028	6.3003
0.6	6.6284	6.6922	6.5836	6.9077
0.7	8.5922	8.6316	8.5678	8.8482
0.8	11.6818	11.7009	11.6768	11.8082
0.9	15.6142	15.6213	15.6162	15.6925
1.0	20.0000	20.0000	20.0000	20.0000

Table 6.2: Percentage errors at $t = 0.6$ for the results in Example 6.1

x-value	LTFDM	LTFEM	LTBEM
0.1	0.05	0.01	0.97
0.2	0.16	0.04	2.08
0.3	0.46	0.28	4.32
0.4	0.96	0.68	5.27
0.5	1.26	0.89	5.78
0.6	0.96	0.68	4.21
0.7	0.46	0.28	2.98
0.8	0.16	0.04	1.08
0.9	0.05	0.01	0.50

6.6 Implementation on a distributed memory architecture

The boundary element method has been shown to be very well-suited to parallel environments (Ingber and Davies 1997). These applications exploit the inherent parallelism in the integral formulation. In the Laplace transform method the space solutions for different time values are completely independent of the method used to solve the elliptic problem in the transform space and as such are ideally suited to be solved on different processors in a distributed system. There is no interprocessor communication during the solution and such an implementation has a very good load balance. The only interprocessor communication occurs during pre-processing when the data is broadcast from the host to all other processors and during post-processing when selected solution values are gathered on the host for the purpose of the display of the results.

Example 6.2

We illustrate the method by solving the following two-dimensional heat conduction problem (Moridis and Reddell 1991c):

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \quad -1 < x < 1,\ -1 < y < 1 \tag{6.9}$$

subject to the boundary conditions, see Figure 6.3

$$u(-1, y, t) = u(x, -1, t) = u(1, y, t) = u(x, 1, t) = 1 \tag{6.10}$$

and the initial condition

$$u(x, y, 0) = 0 \tag{6.11}$$

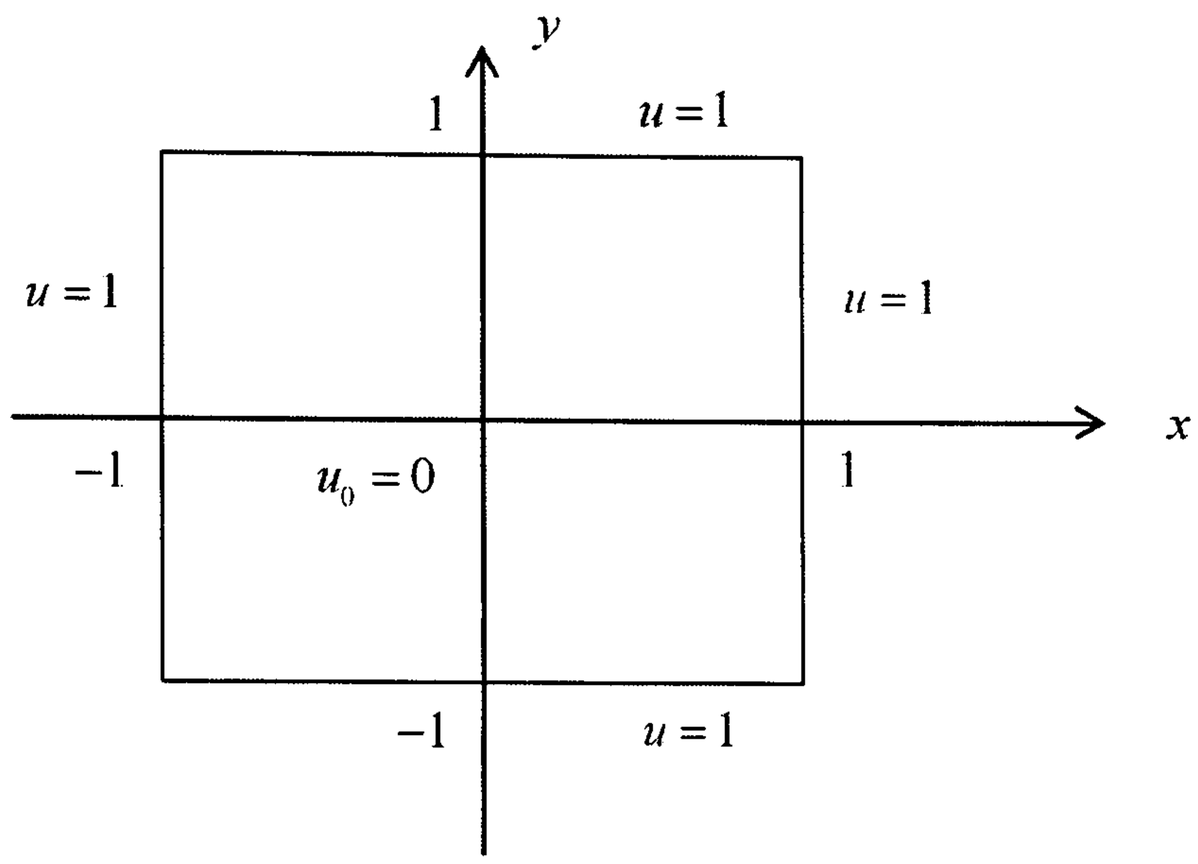

Figure 6.3: Boundary and initial conditions for Example 6.2

The analytic solution is

$$u(x,y,t) = 1 - \frac{16}{\pi^2}\sum_{n=0}^{\infty}\sum_{m=0}^{\infty}\frac{(-1)^{m+n}}{(2m+1)(2n+1)}\cos\left[\frac{(2n+1)\pi x}{2}\right]\times\ldots$$

$$\ldots\times\cos\left[\frac{(2m+1)\pi y}{2}\right]\exp\left(-\alpha\pi^2\left[(2m+1)^2+(2n+1)^2\right]t/4\right)$$

A parallel implementation involves seeking the approximation U to the solution u at the times T_p, $p = 1, 2, \ldots, P$ where P is the number of processors available. Each set of solutions U_p is evaluated on a different processor, $p = 1, 2, \ldots, P$, in parallel. The load balancing thus achieved is excellent. We measure the efficiency of the implementation in parallel by *speed-up*, which is defined as

$$S_P = \frac{\tau_1}{\tau_P}$$

where τ_i is the computing time on i processors.

Implementation on four T800 transputers

The problem defined above was solved using the five different methods from Chapter 2 for the modified Helmholtz equation in Laplace space (Davies *et al.* 1997).

1. **Finite difference method (FDM)** A uniform 16×16 grid on the square was used with the five-point formula for the Laplacian to define a Gauss-Seidel formulation

$$\bar{U}_{i,k} = \left(\bar{U}_{i-1,k} + \bar{U}_{i,k-1} + \bar{U}_{i+1,k} + \bar{U}_{i,k+1}\right) / \left(4 + \lambda_j h^2/\alpha\right)$$

with

$$\bar{U}_{0,k} = \bar{U}_{16,k} = \bar{U}_{i,0} = \bar{U}_{i,16} = 0$$

2. **Finite element method (FEM)** A graded 16×16 mesh was used to define a set of linearly triangular elements and the finite element system of equations was developed in the usual manner with an explicit formulation of the stiffness matrix, *i.e.* no numerical quadrature (Davies 1985). The equation solution was effected using Gauss elimination.

3. **Boundary element method (BEM)** The boundary element method was set up with 68 linear elements and eight-point Gauss quadrature was used to develop the system matrices. The singular integrals were effected using Telles transformation method.

4. **The method of fundamental solutions (MFS)** This method was set up with 68 uniformly distributed field points on the boundary and 69 source points distributed uniformly on the bounding circle.

5. **Kansa's multiquadric method (MQM)** 22 uniformly distributed points were placed on the boundary with 32 interior points.

The problems were solved on a network of processors comprising four T800 transputers and the solution was sought at eight time values:

$$T = 0.1,\ 0.2,\ 0.5,\ 1,\ 2,\ 5,\ 10 \text{ and } 20$$

Table 6.3: cpu times (s) for the five different methods for the solution of Example 6.2 on four T800 transputers

No. of processors	FDM	FEM	BEM	MFS	MQM
1	2537	2617	923	92.3	73.6
2	1269	1309	464	46.2	36.9
4	634	654	233	23.2	18.8

with Stehfest parameter $M = 6$. The computing times are shown in Table 6.3.

The speed-up in the five cases is indistinguishable; they all exhibit linear speed-up and a typical case is shown in Figure 6.4.

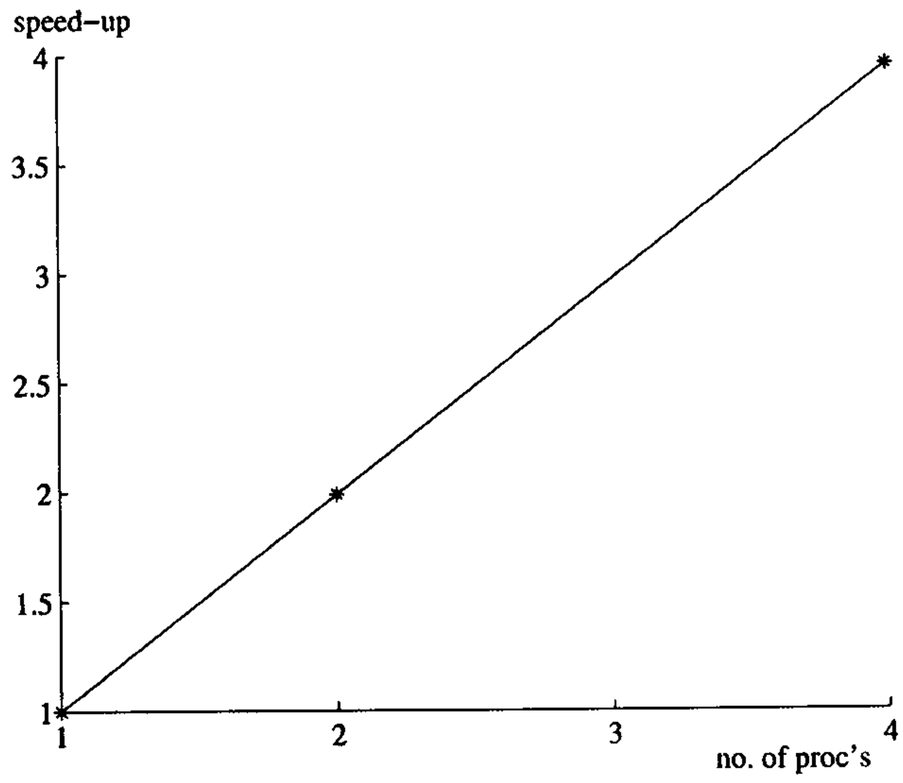

Figure 6.4: Speed-up for the solution of Example 6.2 on four T800 transputers

This linear speed-up, *i.e.* doubling the number of processors halves the computing time, is exactly what we should expect in this situation since there is no interprocessor communication during the solution process. Such communication occurs only in broadcasting the data to the processors and in gathering the results prior to post-processing and these require negligable computing time.

Implementation on a cluster of SUN workstations using PVM

The solution process to Example 6.2 using the Laplace transform boundary element method on the four transputers was compared with the solution using a cluster of eight SUN4 Sparcstations using the PVM message passing protocol. Computation times are shown in Tables 6.4 and 6.5 (Davies *et al.* 1996).

Table 6.4: Computation times for the transputer network

Stehfest M-parameter	Processors 1	Processors 2	Processors 4
6	923	464	233
8	1231	619	310
10	1539	773	388

Table 6.5: Computation times for the PVM SUN cluster

Stehfest M-parameter	Processors 1	Processors 2	Processors 4	Processors 8
6	263	137	70	70
8	352	188	170	91
10	463	232	116	112

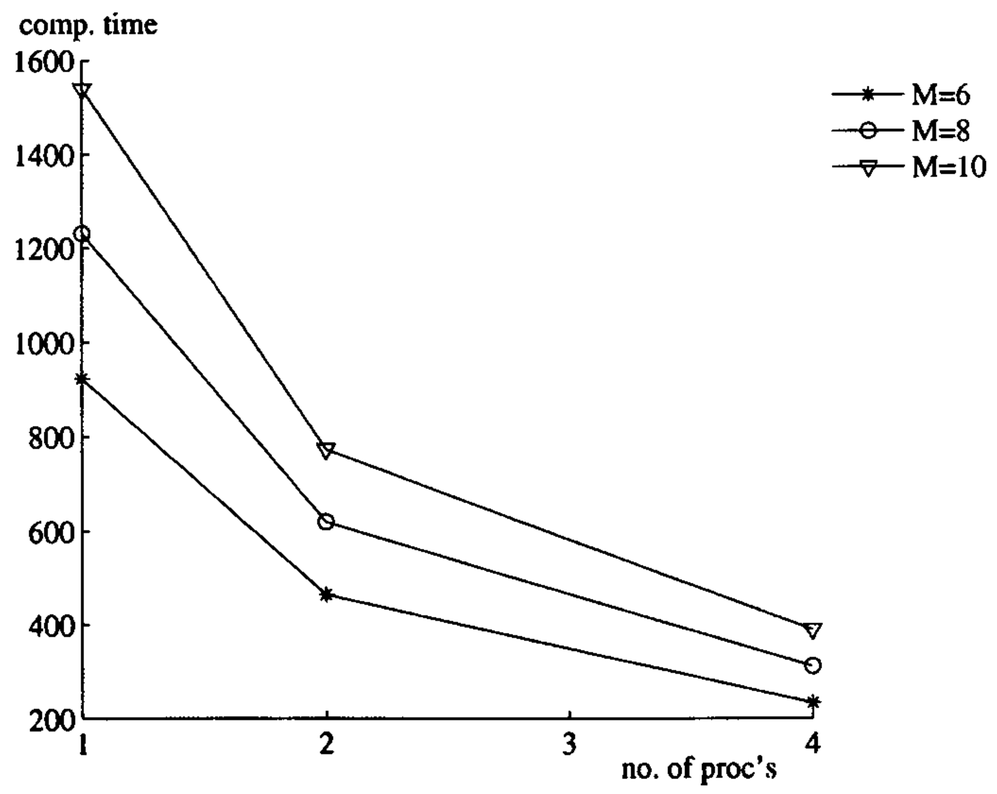

Figure 6.5: Computation time for the solution of Example 6.2 on the transputer network

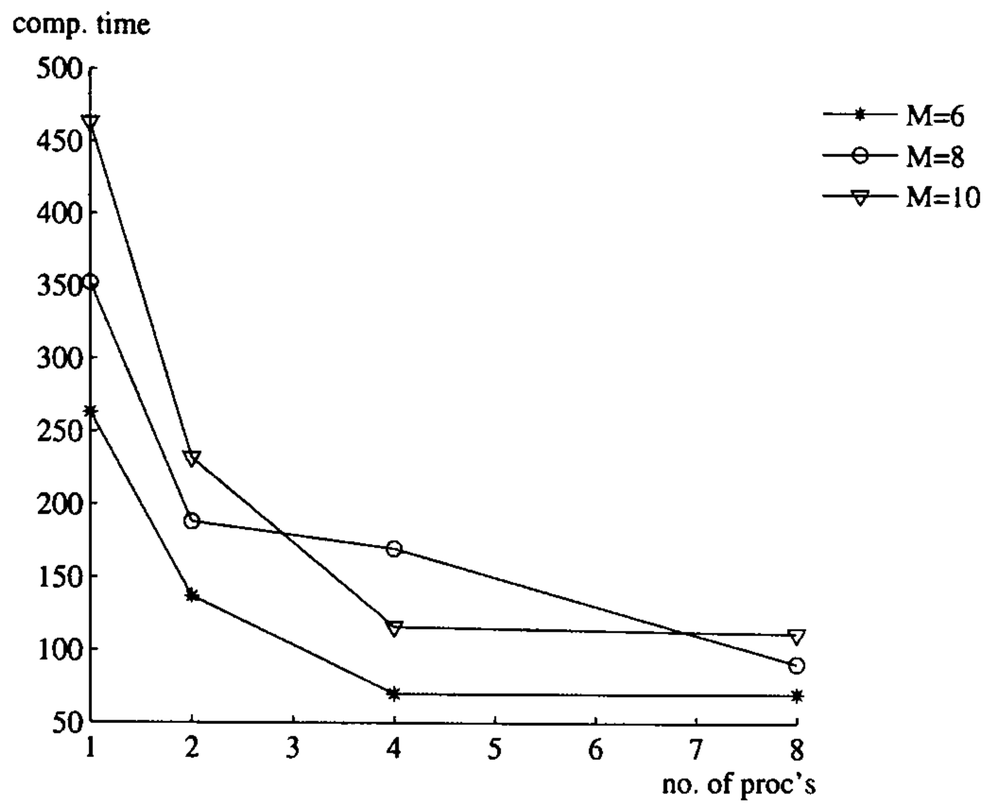

Figure 6.6: Computation time for the solution of Example 6.2 on the PVM SUN cluster

Figures 6.5 and 6.6 show the computation times to indicate the problems that occur on the SUN cluster. The results from the transputer network indicate almost perfect linear speed-up as we have already seen. The results from the PVM SUN cluster, however, are surprising. In the cases of $M = 6$ and $M = 10$ we have linear speed-up in going from 1 to 4 processors but there is almost no improvement in using 8 processors. In the case $M = 8$ the degradation in performance occurs when we go from 2 processors to 4. Overall in all three cases we have a speed-up by a factor of approximately 3.9, about fifty percent of what would be expected. The explanation is not obvious, we know that there is certainly no interprocessor communication during computation. However, there are suspicions that, even though there is no need for communication under PVM, the system is nevertheless preparing for such communication and so incurs the overhead unnecessarily.

The requirement to overcome the speed-up difficulties was removed by the acquisition of a sixty-four processor *nCube* machine.

Implementation on a sixty-four processor *nCube* parallel computer

Example 6.2 was again solved but this time on a sixty-four processor *nCube* parallel computer arranged in a hypercube configuration using the Laplace transform boundary element method (Davies and Crann 2001). The boundary was divided into 68 linear elements. Eight-point Gauss quadrature was used to develop the system matrices with the singular integrals evaluated using Telles transformation method. Solutions were obtained at 64 different times, $T_n = 0.1n;\ n = 1, \ldots, 64$ with the solution obtained using 1, 2, 4, 8, 16, 32 and 64 processors, *i.e.* hypercubes of dimensions $2^d :\ d = 0, \ldots, 6$.

In Figures 6.7 and 6.8 we show the speed-up for the solution of Example 6.2. We see, in Figure 6.7, that there is almost perfect linear speed-up as would be expected because the time-domain decomposition of the problem by the Laplace transform completely uncouples the calculation of the solution at each T_n. However, there is also a communication overhead associated with the implementation and this occurs in the so-called 'broadcast' and 'gather' of the data to and from the processors, *i.e.* in the passing of data and messages out to the processors and then retrieving data back again. If the 'broadcast' and 'gather' times are included then the speed-up is slightly less than linear as shown in Figure 6.8.

Crann, Davies and Mushtaq (1998) also compare the speed-up for the solution of this problem using Stehfest's inversion method with the method using shifted Legendre polynomials and report almost identical computation times and therefore almost perfect linear speed-up.

It is interesting to report here the results of a Laplace transform FDM approach on a distributed memory architecture. Davies *et al.* (2000) use a Jacobi iterative technique for the elliptic problem in Laplace space. We show that the computation effort increases with the parameter T in the Stehfest method as shown in Figure 6.9.

The behaviour of the convergence of the algorithm with respect to T

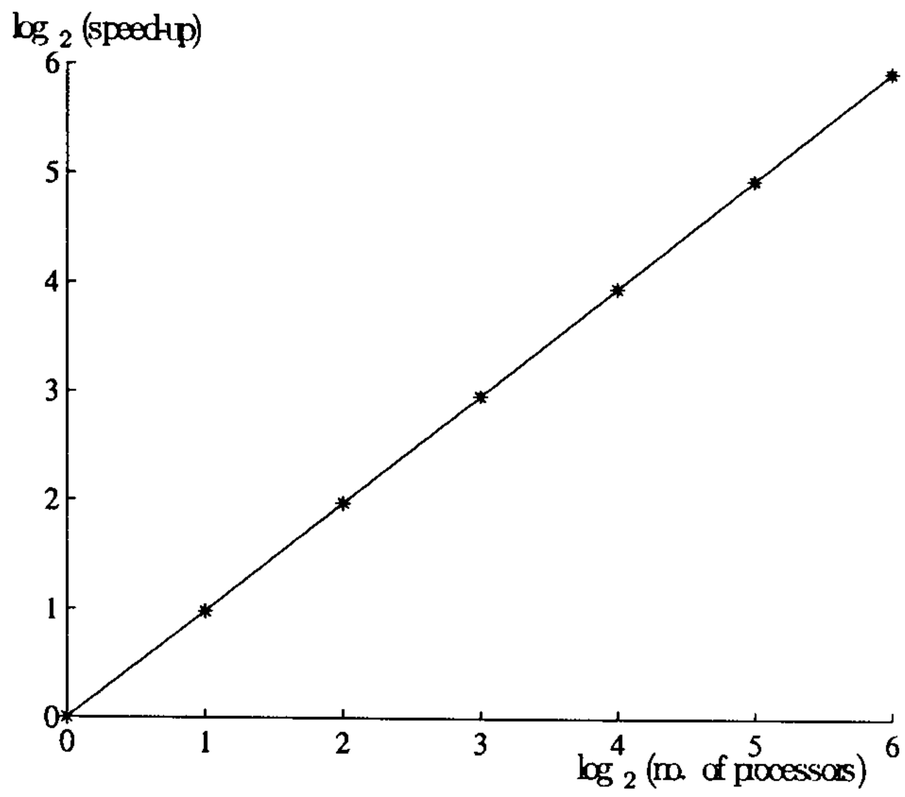

Figure 6.7: Speed-up for the solution of Example 6.2 on the *nCube*: without 'broadcast' and 'gather'

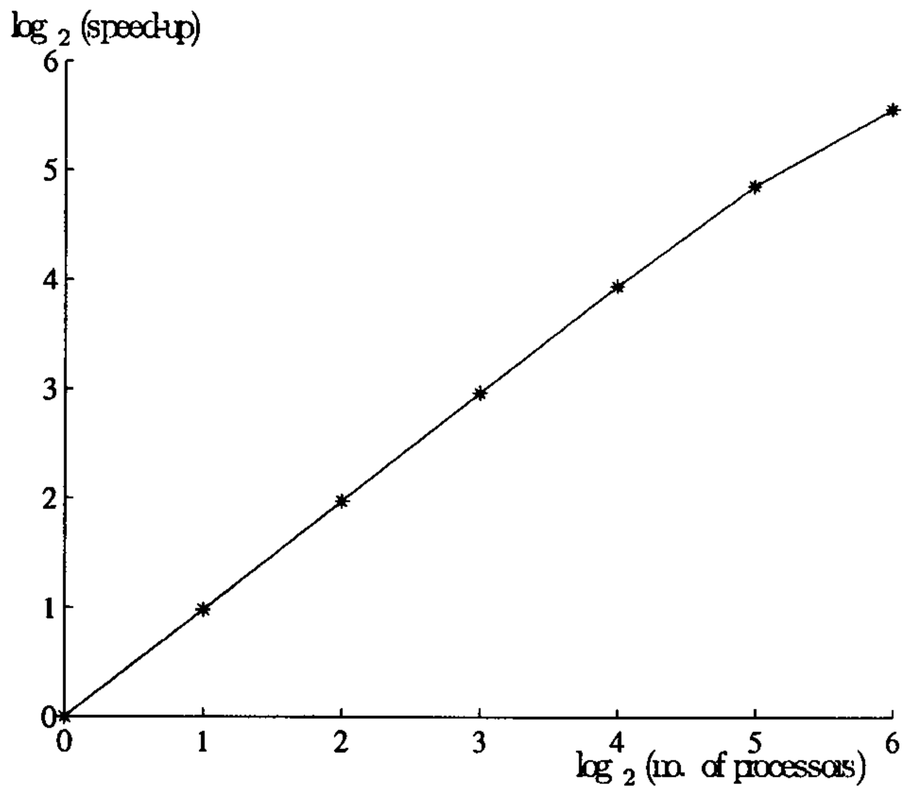

Figure 6.8: Speed-up for the solution of Example 6.2 on the *nCube*: with 'broadcast' and 'gather'

leaves us with a significant load balancing problem. We must allocate T values to the processors in such a way that the total work on each one is the same. The approximate linear nature of the relationship between work load and T provides a possible way forward since we can calculate suitable values of T so that the total work on each processor with different numbers of T values remains fixed. This load balancing difficulty is a feature of the fact that we are using an iterative, *i.e.* indirect, method to solve the system

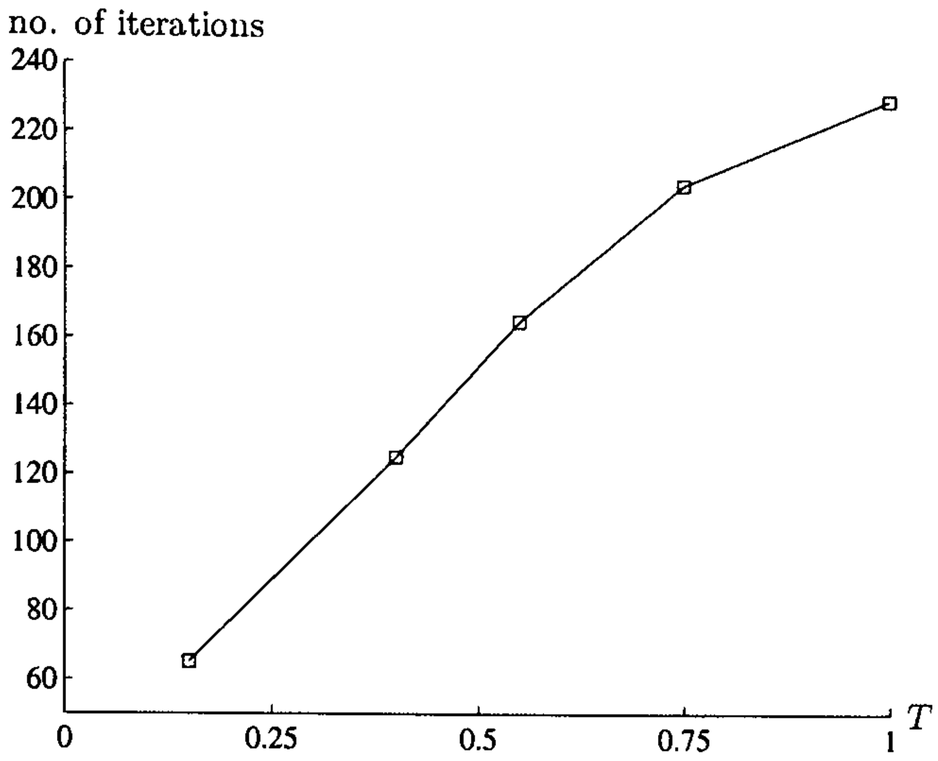

Figure 6.9: Number of iterations for convergence of the LTFDM as a function of T

of equations. The direct solution method that we usually use, *i.e.* Gauss elimination, does not exhibit the same problem.

6.7 Summary of Chapter 6

We have illustrated the Laplace transform method by solving initial-value problems. We used the transform to reduce the problem to an elliptic one in the space domain, solved this equation by a variety of methods, and then inverted back to a solution of the original problem using Stehfest's inversion method. We have compared the solutions to a test heat conduction problem using the Laplace transform together with the finite difference, finite element and boundary element methods and recorded good results.

We have also shown that the numerical Laplace transform method using Stehfest's inversion process is ideally suited to implementation on a distributed memory architecture. The user would choose the solver for the resulting elliptic problem according to which of the methods is preferred or available.

Chapter 7

The Laplace Transform Boundary Element Method with Dual Reciprocity

7.1 Introduction

In Chapter 3 we introduced the boundary element method for the solution of elliptic problems. In Chapters 5 and 6 we followed with the Laplace transform method for the solution of parabolic problems showing that they can be solved easily and accurately by a variety of methods when using the Laplace transform for the time variable including the boundary element method.

In the case that the initial condition, u_0, of the time-dependent problem is zero the resulting elliptic equation becomes homogeneous. Similarly if u_0 is harmonic in the two-dimensional region we can make a change in the dependent variable to obtain a homogeneous equation. In both cases the resulting elliptic equation, the modified Helmholtz equation (4.16) may be solved using the boundary element method for which a suitable fundamental

solution is

$$u^* = \frac{1}{2\pi} K_0 (pR) \tag{7.1}$$

before inverting back to the time domain using a suitable inversion process.

However if the elliptic equation is not homogeneous we must use a suitable approach to handle the non-homogeneity. The dual reciprocity method allows us to do this and at the same time use the simpler Laplacian fundamental solution

$$u^* = -\frac{1}{2\pi} \ln(R) \tag{7.2}$$

We note here that it is not essential to use equation (7.2). Zhu *et al.* (1994) use the fundamental solution equation (7.1) in association with the dual reciprocity method to solve such problems.

The dual reciprocity method was first proposed by Nardini and Brebbia (1982) for elastodynamic problems and extended by Nardini and Brebbia (1985), Partidge and Brebbia (1989) and Partridge and Wrobel (1990) and the first text book describing the ideas for general problems was presented by Partridge, Brebbia and Wrobel in 1992. It has proved to be a powerful technique for solving elliptic partial differential equations and its great advantage is that only boundary integrals need to be carried out, preserving the elegance of the traditional boundary element method.

Partridge *et al.* (1992) suggested, from computational experiments, that the number of internal points, L, and boundary points, N, should be chosen to ensure that $L \geq N/2$. Although it is still mentioned now and then by various authors, in all our examples we have not found this to be necessary and there seems to be no definitive rule nor analytic discussion in the literature.

7.2 The Laplace transform boundary element method with dual reciprocity

We shall describe the method in the context of the Laplace transform boundary element method with dual reciprocity as an additional scheme for handling the right-hand side of our non-homogeneous equation in Laplace space.

We consider the initial boundary-value problem defined in the two-dimensional region, D, bounded by the closed curve $C = C_1 + C_2$

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \quad \text{in } D \tag{7.3}$$

subject to the boundary conditions

$$u = u_1(x, y, t) \quad \text{on } C_1 \tag{7.4}$$

$$q \equiv \frac{\partial u}{\partial n} = q_2(x, y, t) \quad \text{on } C_2 \tag{7.5}$$

and the initial condition

$$u(x, y, 0) = u_0(x, y) \quad \text{in } D \tag{7.6}$$

We define the Laplace transform in the usual way so that the initial boundary-value problem becomes

$$\nabla^2 \bar{u} = \frac{1}{\alpha}(\lambda \bar{u} - u_0) \quad \text{in } D \tag{7.7}$$

subject to

$$\bar{u} = \bar{u}_1 \quad \text{on } C_1 \tag{7.8}$$

$$\bar{q} = \bar{q}_2 \quad \text{on } C_2 \tag{7.9}$$

If we write the right-hand side of equation (7.7) as $b = (x, y; \bar{u}, \lambda)$, then by using the fundamental solution and Green's theorem, equation (7.7) can be written in the usual integral form, see equation (3.5),

$$c_P \bar{u}_P + \oint_C \bar{q}^* \bar{u} ds - \oint_C \bar{u}^* \bar{q} ds = \int_D b \bar{u}^* dA \tag{7.10}$$

The right-hand side of equation (7.7), b, is expanded over D as a series of interpolation functions, f_j,

$$b \approx \sum_{j=1}^{N+L} \alpha_j f_j(R) \tag{7.11}$$

where α_j are coefficients to be determined by a collocation process using N boundary points and L interior points, see Figure 7.1. The interpolation functions, f_j, are chosen so that we can find a particular solution, $\hat{u}$, with the property $\nabla^2 \hat{\hat{u}}_j = f_j$.

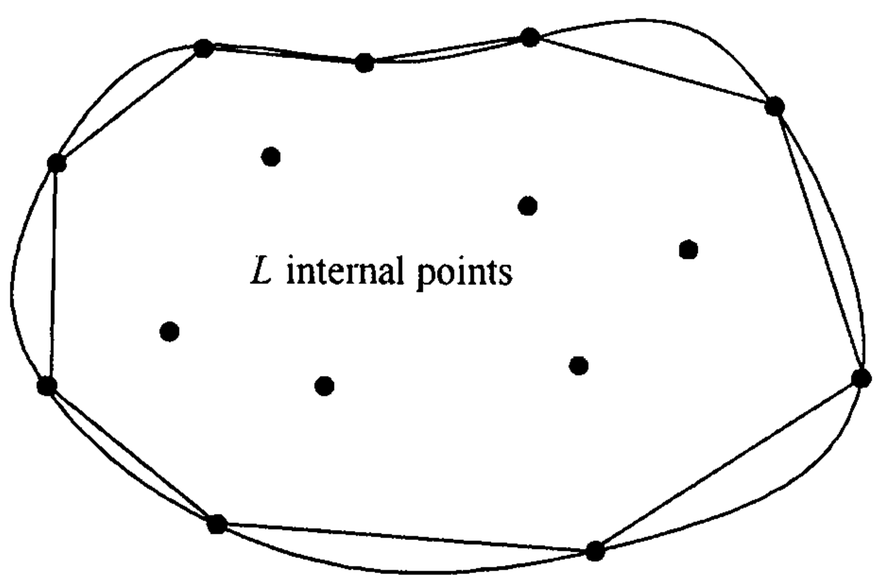

Figure 7.1: Boundary and internal nodes used in the dual reciprocity method.

Using equation (7.11) in the right-hand-side of equation (7.10) together with Green's theorem we obtain the boundary integral form

$$c_i \bar{u} + \oint_C \bar{q}^* \bar{u} ds - \oint_C \bar{u}^* \bar{q} ds = \sum_{j=1}^{N} \left\{ \alpha_j \left(c_i \hat{\bar{u}}_j + \oint_C \bar{q}^* \hat{\bar{u}}_j ds - \oint_C \bar{u}^* \hat{\bar{q}}_j ds \right) \right\} \tag{7.12}$$

Internal values are given by

$$c_k \bar{u} + \oint_C \bar{q}^* \bar{u} ds - \oint_C \bar{u}^* \bar{q} ds = \sum_{j=1}^{L} \left\{ \alpha_j \left(c_k \hat{\bar{u}}_j + \oint_C \bar{q}^* \hat{\bar{u}}_j ds - \oint_C \bar{u}^* \hat{\bar{q}}_j ds \right) \right\} \tag{7.13}$$

Note that equation (7.13) contains no domain integrals since the source term integral has been replaced by equivalent boundary integrals.

Combining equations (7.12) and (7.13) and collocating at the $N + L$ points, yields the overall set of equations

$$\begin{aligned} \mathbf{H}\bar{\mathbf{U}} + \mathbf{G}\bar{\mathbf{Q}} &= \left[\mathbf{H}\hat{\bar{\mathbf{U}}} + \mathbf{G}\hat{\bar{\mathbf{Q}}}\right] \boldsymbol{\alpha} \\ &= \left[\mathbf{H}\hat{\bar{\mathbf{U}}} + \mathbf{G}\hat{\bar{\mathbf{Q}}}\right] \mathbf{F}^{-1}\mathbf{b} \end{aligned} \tag{7.14}$$

using

$$\mathbf{b} = \mathbf{F}\boldsymbol{\alpha} \tag{7.15}$$

where the matrix $\mathbf{F} = [f_j(R_i)]$ is the collocation matrix from equation (7.11).

Defining

$$\mathbf{S} = \left[\mathbf{H}\hat{\bar{\mathbf{U}}} + \mathbf{G}\hat{\bar{\mathbf{Q}}}\right] \mathbf{F}^{-1} \tag{7.16}$$

equation (7.14) becomes

$$\mathbf{H}\bar{\mathbf{U}} + \mathbf{G}\bar{\mathbf{Q}} = \mathbf{S}\mathbf{b} \tag{7.17}$$

and $\mathbf{S}$ is obtained from known matrices which depend only on the geometry.

Although b in the right-hand-side of equation (7.10) is a function of $\bar{u}$, it is helpful to consider first the case $b = b(x, y)$ *i.e.* independent of u then $\mathbf{b}$ is known, and setting

$$\mathbf{R} = \mathbf{S}\mathbf{b}$$

in equation (7.17) we obtain

$$\mathbf{H}\bar{\mathbf{U}} + \mathbf{G}\bar{\mathbf{Q}} = \mathbf{R} \tag{7.18}$$

where $\mathbf{R}$ is known.

Applying the boundary conditions in equation (7.18), we obtain the system of equations in matrix form in a similar manner to that described in Section 3.3

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

where $\mathbf{x}$ is a matrix of unknown values of $\bar{\mathbf{U}}$ and $\bar{\mathbf{Q}}$ and $\mathbf{y}$ is a vector of known values.

We now return to the situation in equation (7.10) where b depends explicitly on $\bar{u}$. In fact since we shall need space derivatives in Chapter 9 as well we consider the case when b is of the form

$$b = p_1(x,y) + p_2(x,y)\bar{u} + p_3(x,y)\frac{\partial \bar{u}}{\partial x} + p_4(x,y)\frac{\partial \bar{u}}{\partial y} \tag{7.19}$$

The basic approximation for the dual reciprocity method is equation (7.11) and its discretised form equation (7.15). Similar equations may be written for $\bar{u}$

$$\bar{u} \approx \sum_{j=1}^{N+L} \beta_j f_j \tag{7.20}$$

$$\bar{\mathbf{U}} = \mathbf{F}\boldsymbol{\beta} \tag{7.21}$$

Differentiating equation (7.20) with respect to x produces

$$\frac{\partial \bar{u}}{\partial x} \approx \sum_{j=1}^{N+L} \beta_j \frac{\partial f_j}{\partial x} \tag{7.22}$$

Rewriting equation (7.21) as $\boldsymbol{\beta} = \mathbf{F}^{-1}\bar{\mathbf{U}}$, the discretised form of $\partial\bar{u}/\partial x$ becomes

$$\frac{\partial \mathbf{F}}{\partial x}\mathbf{F}^{-1}\bar{\mathbf{U}} \tag{7.23}$$

In a similar manner, for $\partial\bar{u}/\partial y$ we obtain the expression

$$\frac{\partial \mathbf{F}}{\partial y}\mathbf{F}^{-1}\bar{\mathbf{U}} \tag{7.24}$$

If

$$\mathbf{P}_i = \text{diag}\,[p_i(x_k, y_k)] \quad i = 2,3,4, \;\; k = 1,\ldots,L+N$$

$$\mathbf{p}_1 = [p_1(x_k, y_k)]$$

and $\mathbf{S}$ is given by equation (7.16) then in a similar manner to that which led to equations (7.16) to (7.18) we obtain

$$(\mathbf{H} - \mathbf{R}_2)\bar{\mathbf{U}} + \mathbf{G}\bar{\mathbf{Q}} = \mathbf{R}_1$$

with

$$\mathbf{R}_1 + \mathbf{R}_2 = \mathbf{R}$$

and

$$\mathbf{R}_1 = \mathbf{S}\mathbf{p}_1, \text{ a known function of position}$$

$$\text{and } \mathbf{R}_2 = \mathbf{S}\left[\mathbf{P}_2 + \mathbf{P}_3\frac{\partial \mathbf{F}}{\partial x}\mathbf{F}^{-1} + \mathbf{P}_4\frac{\partial \mathbf{F}}{\partial y}\mathbf{F}^{-1}\right]$$

Applying the boundary conditions and rearranging, we again obtain a system of equations in matrix form

$$\mathbf{A}\mathbf{x} = \mathbf{y}$$

We solve this system of equations and invert the transform using Stehfest's inversion technique to produce the numerical solution to our initial boundary-value problem.

7.2.1 Choice of approximation function, f

Many types of approximating function f have been suggested. Nardini and Brebbia (1982) first adopted the function $f = R$ where R is the distance function used in the definition of the fundamental solution. Later authors considered other functions from the series

$$f = 1 + R + R^2 + R^3 + \ldots + R^m \tag{7.25}$$

and Partridge *et al.* (1992) suggested the case $f = 1 + R$ to be generally recommended. Recent work related to the theory of mathematical interpolation based on the so-called radial basis functions, of which equation (7.25) is one particular case, has produced many other ideas, including the use of thin plate splines which we shall use later. However, as long as b is suitably well-behaved then the coefficients α_j are well-defined (Wrobel 2002).

We use the dual reciprocity code written by Toutip (2001) as a subroutine in our Laplace transform boundary element code. The dual reciprocity code considers the following Poisson-type equation:

$$\nabla^2 u = p_1 + p_2 u + p_3\frac{\partial u}{\partial x} + p_4\frac{\partial u}{\partial y}$$

If p_3 and p_4 are both zero then the code supports both $f = 1 + R$ and the augmented thin plate spline $f = R^2 \log R + a + bx + cy$ for the interpolation functions in equation (7.11). If one of p_3 or p_4 is non-zero then the code supports only $f = 1 + R$.

7.3 The solution of linear initial boundary-value problems

In this section we consider a variety of linear initial boundary-value problems, with b given by equation (7.19), to demonstrate the Laplace transform boundary element method using dual reciprocity to handle the right-hand side of the equation. We use $N = 32$ boundary points and $L = 9$ internal points, see Figure 7.2 and for the numerical Laplace transform we use Stehfest's inversion method with parameter value $M = 8$. We include in the following examples a term $h(x, y, t)$ to enable us to consider problems with simple analytic solutions.

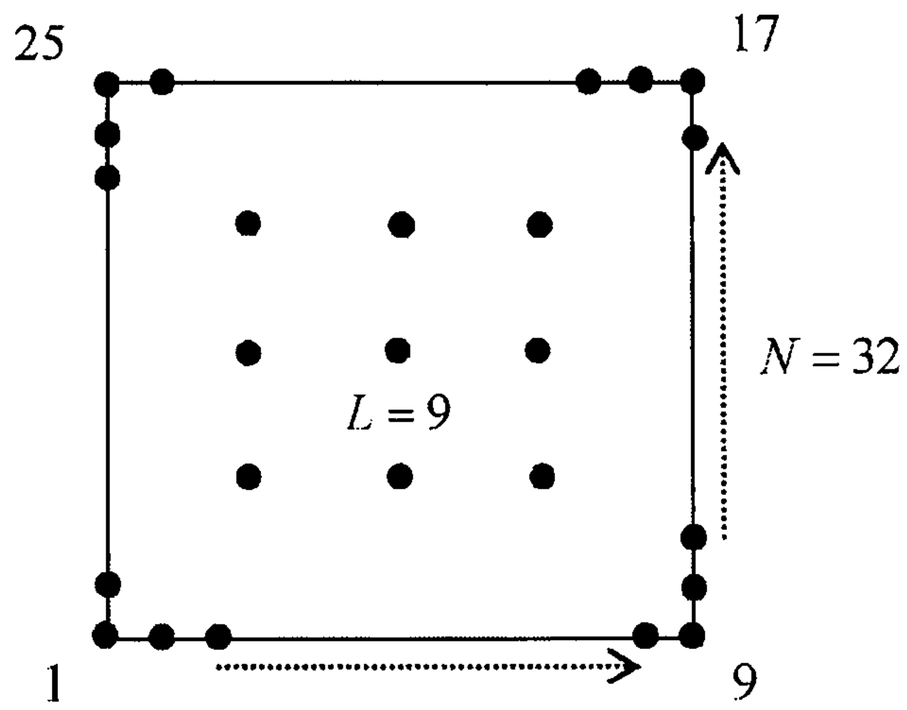

Figure 7.2: Distribution of boundary and internal nodes for a square geometry

Example 7.1

This example is defined in the unit square $\{(x, y) : 1 < x < 2,\ 1 < y < 2\}$, see Figure 7.3

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} + h \tag{7.26}$$

where

$$h = (2 + x^2)e^{-t}$$

with boundary conditions

$$u = e^{-t} \text{ on } x = 1, \quad u = 4e^{-t} \text{ on } x = 2$$

$$q = 0 \text{ on } y = 1 \text{ and } y = 2$$

and initial condition

$$u_0 = x^2$$

and we use $\alpha = 1.0$.

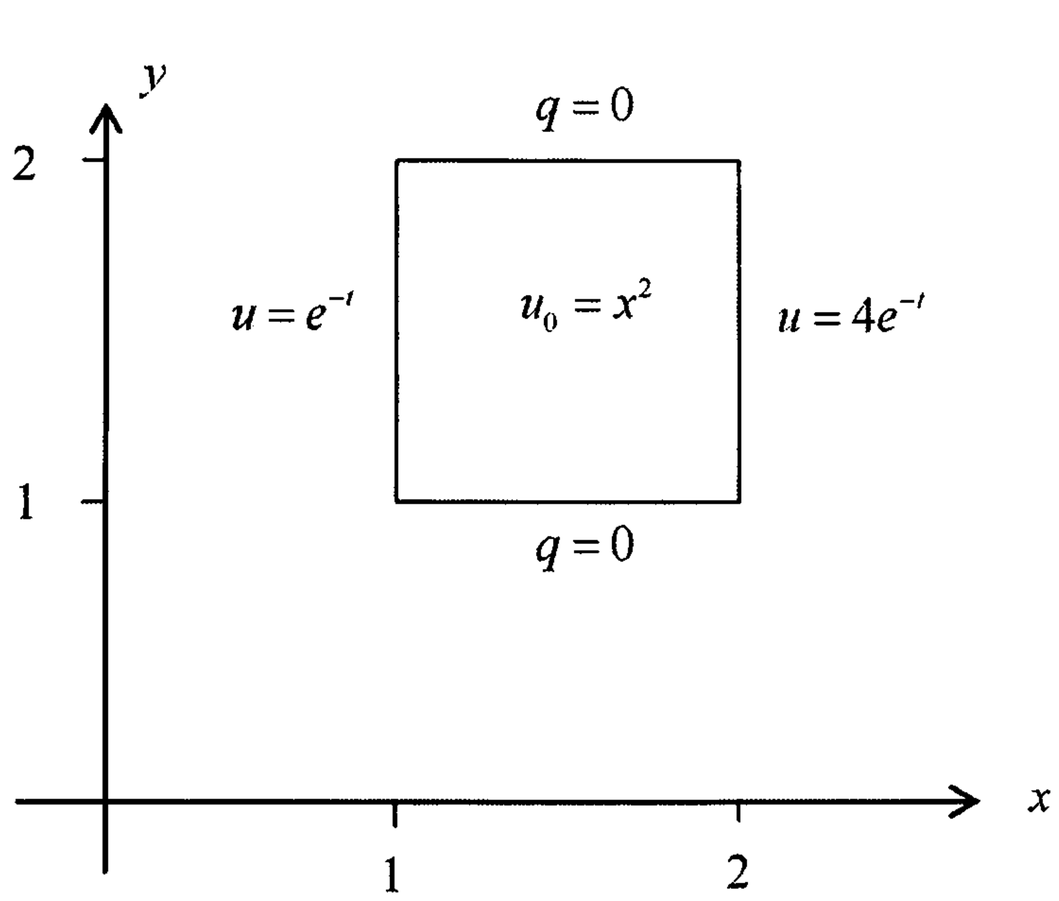

Figure 7.3: Boundary and initial conditions for Example 7.1

In Laplace space this problem is

$$\nabla^2 \bar{u} = \frac{1}{\alpha}\left(\lambda \bar{u} - x^2\right) + \bar{h}$$

with

$$\bar{h} = \frac{2 + x^2}{1 + \lambda}$$

and boundary conditions

$$\bar{u} = \frac{1}{1+\lambda} \text{ on } x = 1, \quad \bar{u} = \frac{4}{1+\lambda} \text{ on } x = 2$$

$$\bar{q} = 0 \text{ on } y = 1 \text{ and } y = 2$$

The analytic solution is

$$u = x^2 e^{-t}$$

and we compare this in Figure 7.4 with the numerical solution for which we used the interpolating function $f = 1 + R$ for the dual reciprocity method.

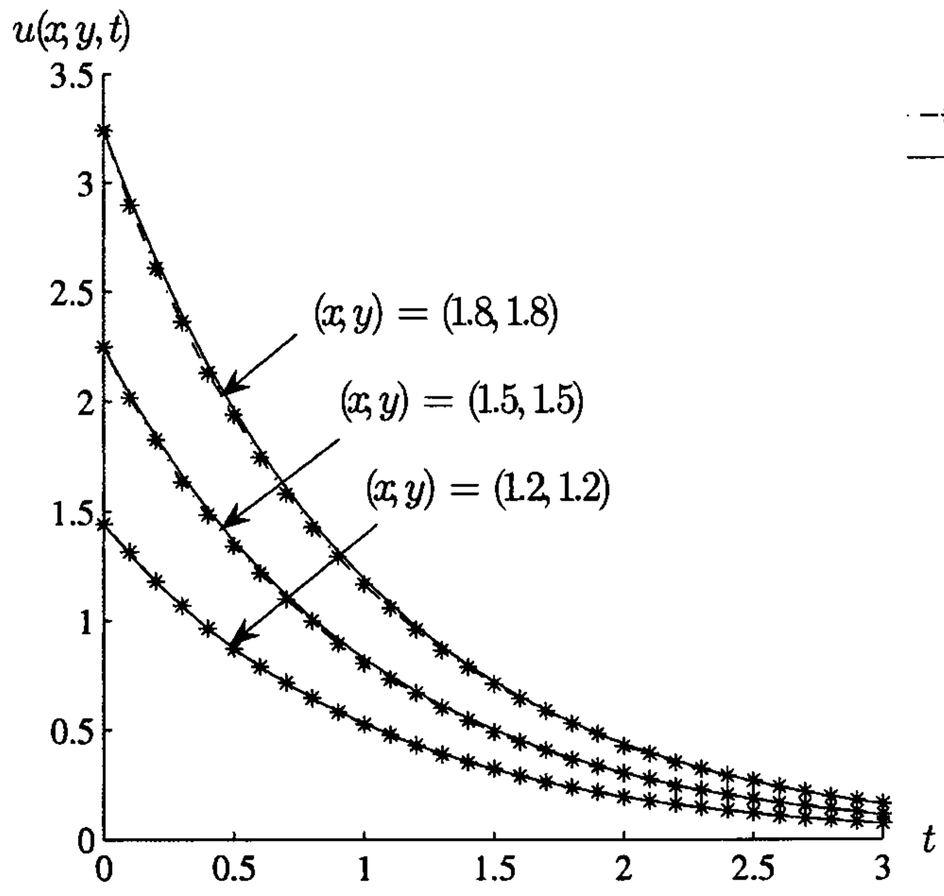

Figure 7.4: Time development of the solution for Example 7.1

We see that the numerical solution is a good approximation to the analytic solution. We show some of the numerical results in Table 7.1 and find that the maximum error is about five percent.

We also solved this problem using augmented thin plate splines and in Table 7.2 we show the results for the internal node $(1.5, 1.5)$ from $t = 0.1$ to 1.0 and compare the two methods with the analytic solution, showing

Table 7.1: Analytic and numerical solution for Example 7.1 in a unit square

time	analytic $x = 1.2$	approx $x = 1.2$	analytic $x = 1.5$	approx $x = 1.5$	analytic $x = 1.8$	approx $x = 1.8$
0.5	0.873404	0.874977	1.364694	1.365278	1.965159	1.971795
1.0	0.529746	0.527231	0.827729	0.822284	1.191929	1.190671
1.5	0.321307	0.322452	0.502043	0.497957	0.722941	0.728017
2.0	0.194882	0.195609	0.304504	0.308966	0.438486	0.446656
2.5	0.118202	0.121666	0.184691	0.192979	0.265955	0.270008
3.0	0.071693	0.077216	0.112021	0.116802	0.161310	0.169625

the percentage errors. We see that the method using augmented thin plate splines gives very poor results.

Table 7.2: Analytic and numerical solution for node $(1.5, 1.5)$ in Example 7.1, with percentage errors

time	analytic solution	$f = 1 + R$ solution	Aug TPS solution	$f = 1 + R$ % error	Aug TPS % error
0.1	2.035884	2.027869	2.085829	0.39	2.45
0.2	1.842144	1.845667	1.931223	0.19	4.84
0.3	1.666841	1.662987	1.695091	0.23	1.69
0.4	1.508220	1.509959	1.306139	0.12	13.40
0.5	1.364694	1.365278	1.189529	0.04	12.84
0.6	1.234826	1.232772	0.850058	0.17	31.16
0.7	1.117317	1.113685	-0.750861	0.33	167.20
0.8	1.010990	1.011307	15.573210	0.03	1440.39
0.9	0.914782	0.913680	8.720648	0.12	853.30
1.0	0.827729	0.822284	15.450520	0.66	1766.62

If we scale the geometry by a factor of two so that the problem domain is $\{(x, y), \frac{1}{2} \leq x \leq 1, \frac{1}{2} \leq y \leq 1\}$ we obtain the results in Table 7.3. The results for $f = 1 + R$ are similar to the previous ones but those for the augmented thin plate spline are now very good, in fact better than for the $f = 1 + R$ which we might expect.

Table 7.3: Analytic and numerical solution for node $(1.5, 1.5)$ in Example 7.1 with percentage errors, after scaling by a factor of 2

time	analytic solution	$f = 1 + R$ solution	Aug TPS solution	$f = 1 + R$ % error	Aug TPS % error
0.1	2.035884	2.036639	2.037439	0.04	0.08
0.2	1.842144	1.844345	1.842113	0.12	0.00
0.3	1.666841	1.665840	1.668133	0.06	0.08
0.4	1.508220	1.510100	1.504669	0.12	0.24
0.5	1.364694	1.366218	1.364577	0.11	0.01
0.6	1.234826	1.233334	1.232312	0.12	0.20
0.7	1.117317	1.119965	1.117729	0.24	0.04
0.8	1.010990	1.012913	1.009925	0.19	0.11
0.9	0.914782	0.918610	0.914851	0.42	0.01
1.0	0.827729	0.821593	0.826981	0.74	0.09

This problem was also solved in squares of size $\{(x, y) : 1 < x < 5,\ 1 < y < 5\}$ and $\{(x, y) : 1 < x < 9,\ 1 < y < 9\}$ to see if the size of the geometry affected the solution and we obtained the results in Tables 7.4, 7.5, 7.6 and 7.7.

Table 7.4: Solutions for node $(3.0, 3.0)$ in $\{(x, y) : 1 < x < 5,\ 1 < y < 5\}$ with percentage errors, before scaling

time	analytic solution	$f = 1 + R$ solution	Aug TPS solution	$f = 1 + R$ % error	Aug TPS % error
0.1	8.143537	8.098644	23.322310	0.55	186.39
0.2	7.368577	7.312969	16.863770	0.75	128.86
0.3	6.667364	6.680464	4.329823	0.20	35.06
0.4	6.032880	6.017361	-4.697392	0.26	177.86
0.5	5.458776	5.432860	27.059930	0.47	395.71
0.6	4.939305	4.958868	10.675640	0.40	116.14
0.7	4.469268	4.516536	-3.360534	1.06	175.19
0.8	4.043961	4.111164	22.983370	1.66	468.34
0.9	3.659127	3.658441	-7.487761	0.02	304.63
1.0	3.310915	3.310810	17.148700	0.00	417.94

Table 7.5: Solutions for node $(3.0, 3.0)$ in $\{(x, y) : 1 < x < 5,\ 1 < y < 5\}$ with percentage errors, after scaling by a factor of 5

time	analytic solution	$f = 1 + R$ solution	Aug TPS solution	$f = 1 + R$ % error	Aug TPS % error
0.1	8.143537	8.198797	8.173505	0.68	0.37
0.2	7.368577	7.379109	7.393033	0.14	0.33
0.3	6.667364	6.705882	6.684691	0.58	0.26
0.4	6.032880	6.056283	6.045638	0.39	0.21
0.5	5.458776	5.482986	5.456286	0.44	0.05
0.6	4.939305	4.954855	4.953763	0.31	0.29
0.7	4.469268	4.496972	4.484585	0.62	0.34
0.8	4.043961	4.082643	4.047904	0.96	0.10
0.9	3.659127	3.670136	3.651706	0.30	0.20
1.0	3.310915	3.328386	3.296805	0.53	0.43

To summarise, using $f = 1 + R$ the results are satisfactory without scaling but the larger the geometry becomes the better the results are after scaling. However for the augmented thin plate spline the results are very poor before scaling but very good afterwards, even better than the results

Table 7.6: Solutions for node $(5.0, 5.0)$ in $\{(x, y) : 1 < x < 9,\ 1 < y < 9\}$ with percentage errors, before scaling

time	analytic solution	$f = 1 + R$ solution	Aug TPS solution	$f = 1 + R$ % error	Aug TPS % error
0.1	22.620935	23.056060	-3955.8980	1.92	17587.77
0.2	20.468269	20.955060	-6297.1330	2.38	30865.34
0.3	18.520456	18.809710	-1644.1830	1.56	8977.66
0.4	16.758001	16.948760	-2378.2500	1.14	14291.73
0.5	15.163266	15.287830	1817.6260	0.82	11887.03
0.6	13.720291	13.780140	-2204.6360	0.44	16168.43
0.7	12.414633	12.472840	2229.6070	0.47	17859.51
0.8	11.233224	11.690050	-1342.5130	4.07	12051.27
0.9	10.164241	10.042650	-390.0205	1.20	3937.18
1.0	9.196986	9.248565	414.0055	0.56	4401.53

Table 7.7: Solutions for node $(5.0, 5.0)$ in $\{(x, y) : 1 < x < 9,\ 1 < y < 9\}$ with percentage errors, after scaling by a factor of 9

time	analytic solution	$f = 1 + R$ solution	Aug TPS solution	$f = 1 + R$ % error	Aug TPS % error
0.1	22.620935	22.593060	22.768480	0.12	0.65
0.2	20.468269	20.519190	20.542030	0.25	0.36
0.3	18.520456	18.614560	18.516730	0.51	0.02
0.4	16.758001	16.829330	16.874930	0.43	0.69
0.5	15.163266	15.205670	15.187000	0.28	0.16
0.6	13.720291	13.776530	13.771550	0.41	0.37
0.7	12.414633	12.444480	12.443230	0.24	0.23
0.8	11.233224	11.283360	11.256030	0.45	0.20
0.9	10.164241	10.233760	10.143770	0.68	0.20
1.0	9.196986	9.191519	9.200835	0.06	0.04

using $f = 1 + R$. This is also compatible with the perceived wisdom within boundary element literature that the dual reciprocity method requires suitable scaling of the geometry. There doesn't seem to be any definitive statement, however we find from time to time an aside within a paper which alludes to this general perception. Natalini and Popov (2005) discuss scaling the geometry although their particular interest is in computation cost rather than accuracy. Consequently in all our examples we shall usually restrict ourselves to regions which lie within $\{(x, y) : 0 \leq x \leq 1, 0 \leq y \leq 1\}$.

Example 7.2

We extend the problem in the previous section to include the term $\partial u/\partial x$. This example is defined in the unit square from $(0,0)$ to $(1,1)$, see Figure 7.5,

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} + h \tag{7.27}$$

where

$$h = (2 + 2x + x^2)e^{-t}$$

with boundary conditions

$$u = 0 \text{ on } x = 1, \quad u = e^{-t} \text{ on } x = 2$$

$$q = 0 \text{ on } y = 1 \text{ and } y = 2$$

initial condition

$$u_0 = x^2$$

and $\alpha = 1$.

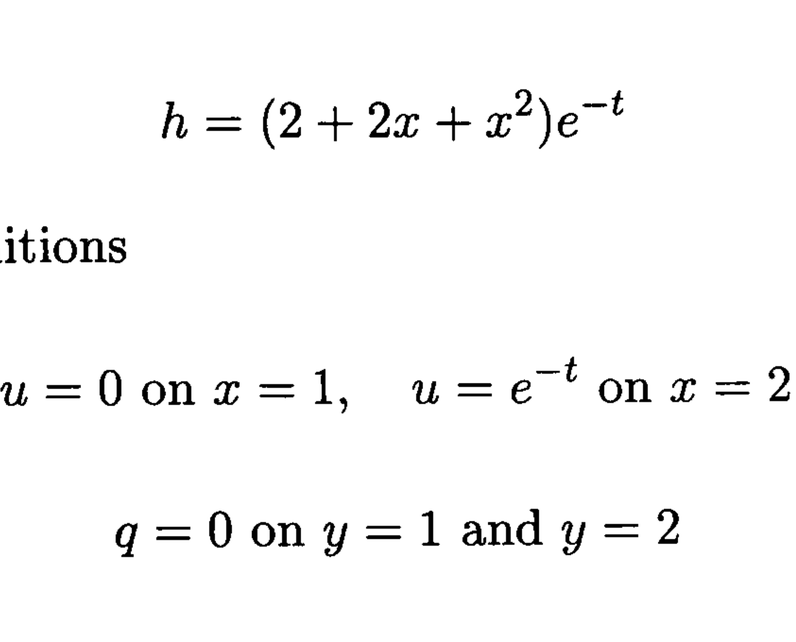

Figure 7.5: Boundary and initial conditions for Example 7.2

The problem is transformed in Laplace space to:

$$\nabla^2 \bar{u} = \frac{1}{\alpha}\left(\lambda \bar{u} - x^2\right) - \frac{\partial \bar{u}}{\partial x} + \bar{h} \tag{7.28}$$

where

$$\bar{h} = \frac{(2 + 2x + x^2)}{1 + \lambda}$$

with boundary conditions

$$\bar{u} = 0 \text{ on } x = 0, \quad u = \frac{1}{1+\lambda} \text{on } x = 1$$

$$\bar{q} = 0 \text{ on } y = 0 \text{ and } y = 1$$

The problem is solved using the boundary element method with dual reciprocity and inverted back using Stehfest's inversion method.

The analytic solution is the same as the previous example:

$$u = x^2 e^{-t}$$

The solution at three internal nodes, $(0.2, 0.2)$, $(0.5, 0.5)$ and $(0.8, 0.8)$ is shown in Figure 7.6 with the numerical results for time at intervals of 0.5 to 3.0 in Table 7.8. We see that the Laplace transform approximation tracks the analytic solution very well.

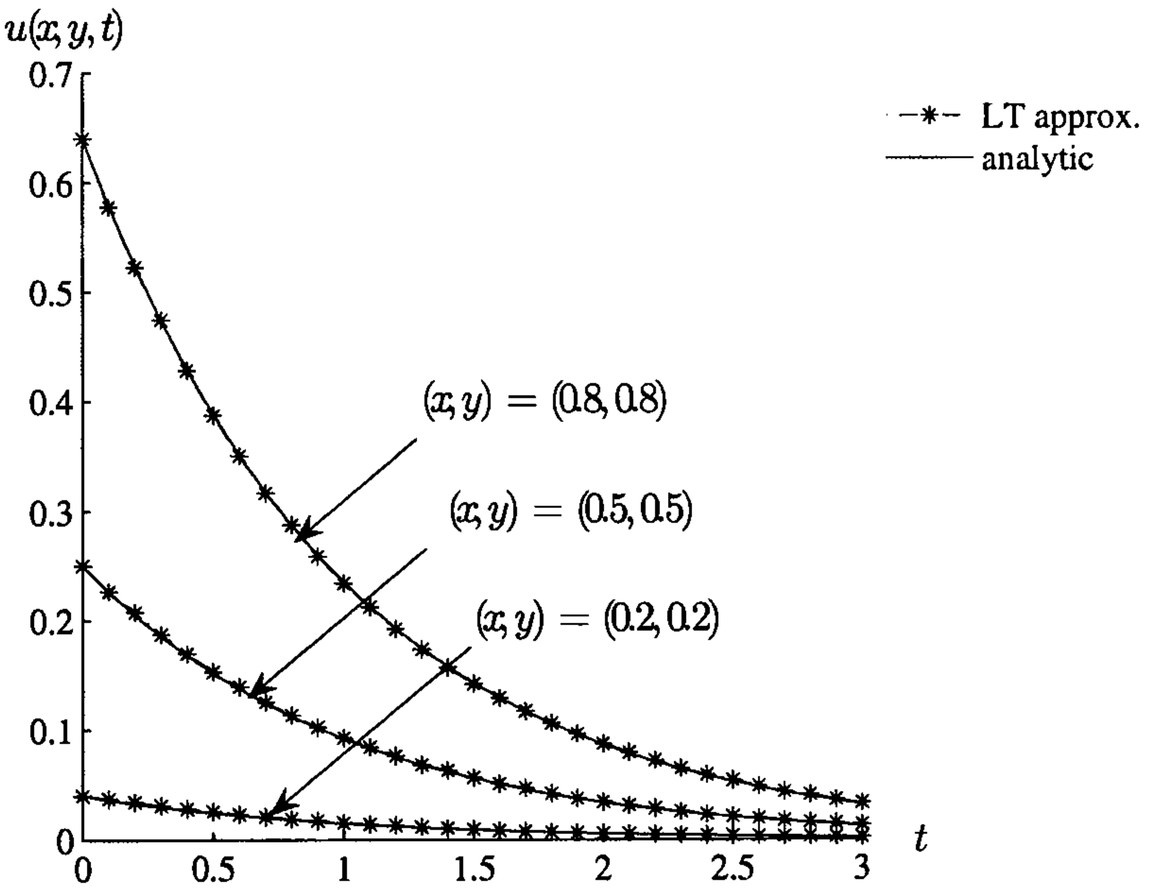

Figure 7.6: Time development of the solution for Example 7.2

Table 7.8: Analytic and numerical solution for Example 7.2

time	analytic $x = 0.2$	approx $x = 0.2$	analytic $x = 0.5$	approx $x = 0.5$	analytic $x = 0.8$	approx $x = 0.8$
0.5	0.024261	0.025936	0.151633	0.154400	0.388180	0.389557
1.0	0.014715	0.015862	0.091970	0.093249	0.235443	0.237373
1.5	0.008925	0.009725	0.055783	0.056919	0.142803	0.144136
2.0	0.005413	0.005874	0.033834	0.034959	0.086615	0.088427
2.5	0.003283	0.003544	0.020521	0.021291	0.052534	0.054107
3.0	0.001991	0.002190	0.012447	0.013073	0.031864	0.034084

Example 7.3

The previous examples were essentially one-dimensional in space. We now consider a problem in which the solution is explicitly dependent on both spatial variables. This example is again defined in the unit square from $(0,0)$ to $(1,1)$

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} + h \tag{7.29}$$

where

$$h = (4 + x^2 + y^2)e^{-t}$$

see Figure 7.7, with boundary conditions,

$$u = (1 + y^2)e^{-t} \text{ on } x = 1, \quad u = (1 + x^2)e^{-t} \text{ on } y = 1$$

$$q = 0 \text{ on } x = 0 \text{ and } y = 0$$

and initial condition

$$u_0 = x^2 + y^2$$

In Laplace space this problem is

$$\nabla^2 \bar{u} = \frac{1}{\alpha}\left(\lambda \bar{u} - (x^2 + y^2)\right) + \bar{h}$$

with

$$\bar{h} = \frac{4 + x^2 + y^2}{1 + \lambda}$$

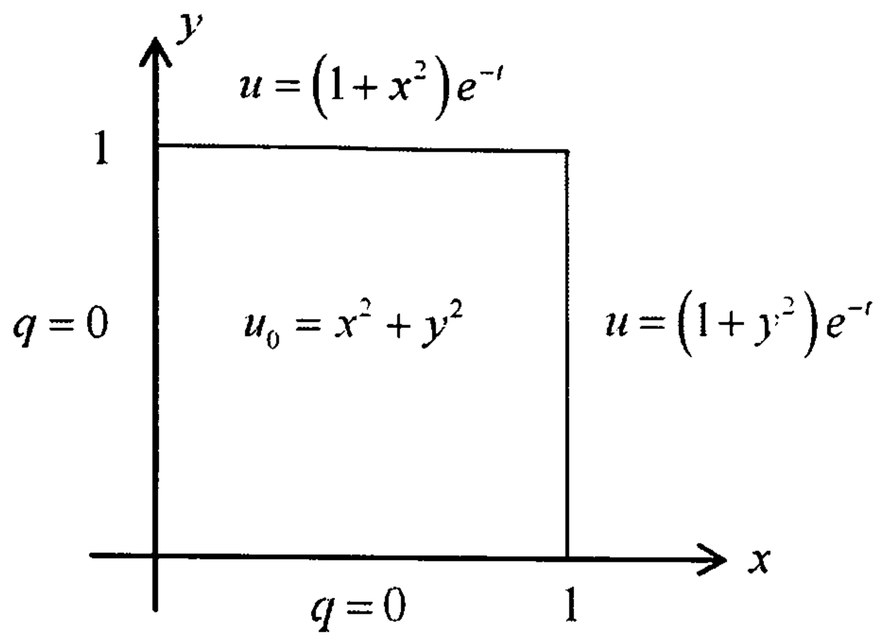

Figure 7.7: Boundary and initial conditions of Example 7.3

and boundary conditions

$$\bar{u} = \frac{1+y^2}{1+\lambda} \text{ on } x = 1, \quad \bar{u} = \frac{1+x^2}{1+\lambda} \text{ on } y = 1$$

$$\bar{q} = 0 \text{ on } x = 0 \text{ and } y = 0$$

The analytic solution is

$$u = (x^2 + y^2)e^{-t}$$

In Figure 7.8 we present the graphical solution of the analytic and approximate solutions and in Table 7.9 we show the numerical solutions for time at intervals 0.5 to 3.0. We see that the approximation is again very good.

Table 7.9: Analytic and numerical solution for Example 7.3

time	analytic (0.2, 0.2)	approx (0.2, 0.2)	analytic (0.5, 0.5)	approx (0.5, 0.5)	analytic (0.8, 0.8)	approx (0.8, 0.8)
0.5	0.048522	0.052398	0.303265	0.305371	0.776359	0.776239
1.0	0.029430	0.032681	0.183940	0.186203	0.470886	0.469709
1.5	0.017850	0.019922	0.111565	0.112316	0.285607	0.288195
2.0	0.010827	0.012715	0.067668	0.069300	0.173229	0.172084
2.5	0.006567	0.007510	0.041042	0.042540	0.105069	0.107481
3.0	0.003983	0.004681	0.024894	0.027323	0.063727	0.066657

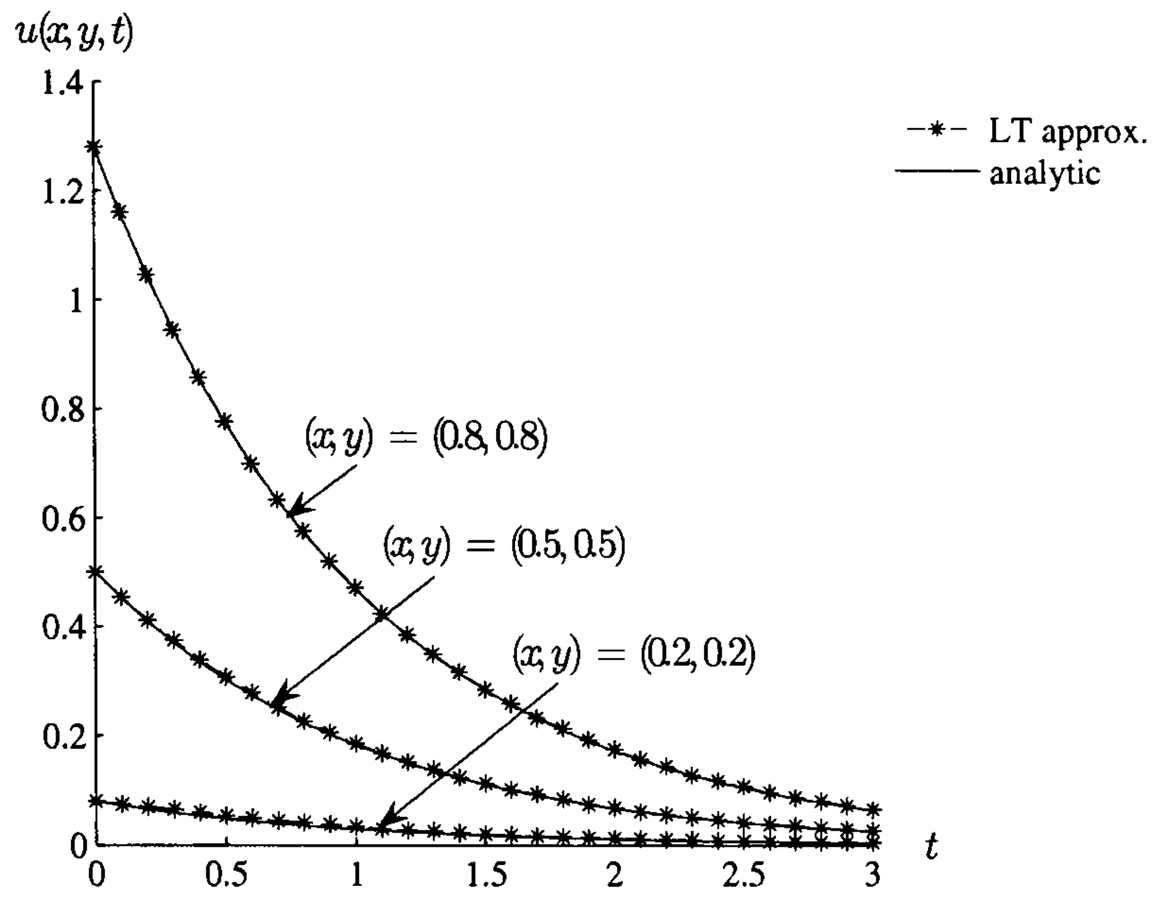

Figure 7.8: Time development of the solution for Example 7.3

Example 7.4

We now consider a similar problem but with curved geometry and a section of the boundary with a prescribed non-zero derivative. The problem is defined in the unit circle $x^2 + y^2 = 1$.

The partial differential equation is again

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} + h \tag{7.30}$$

where

$$h = (4 + x^2 + y^2)e^{-t}$$

with boundary conditions

$$u = e^{-t} \text{ on } x \geq 0$$

$$q = 2e^{-t} \text{ on } x < 0$$

and initial condition

$$u_0 = x^2 + y^2$$

Exploiting the symmetry of the geometry, we shall solve the problem in the upper half-plane introducing the boundary condition

$$q = 0 \text{ on } y = 0$$

see Figure 7.9, with 32 boundary nodes and 12 internal nodes, see Figure 7.10 and we shall use $\alpha = 1$.

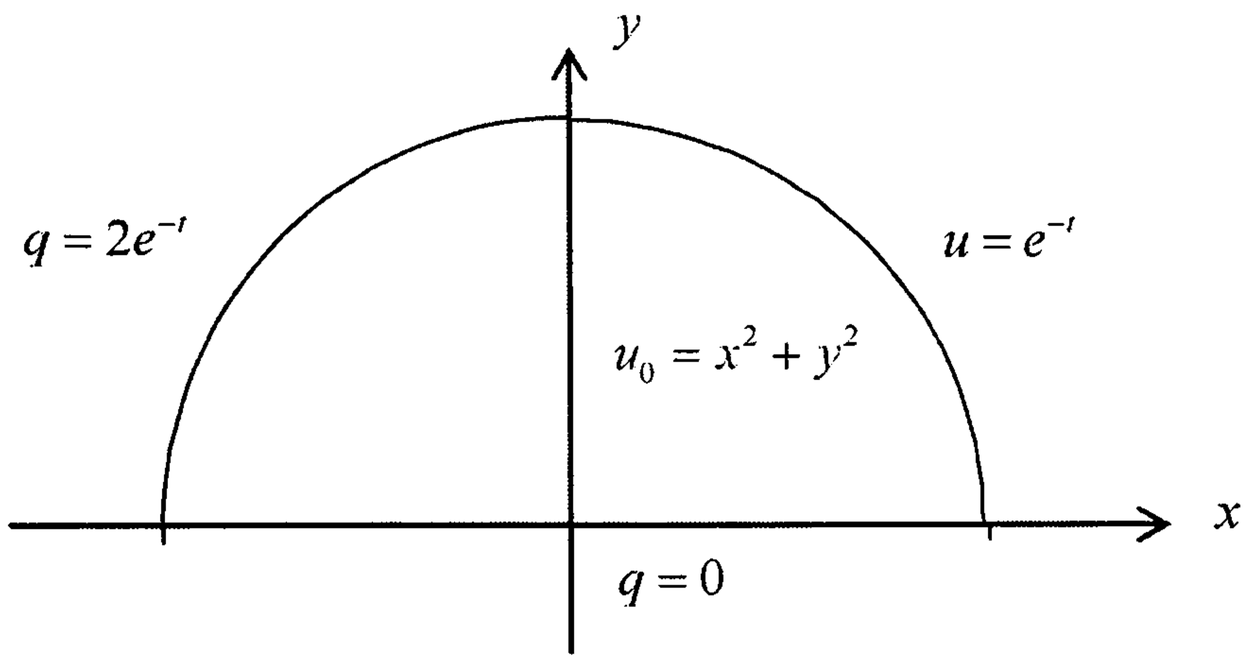

Figure 7.9: Boundary and initial conditions for Example 7.4

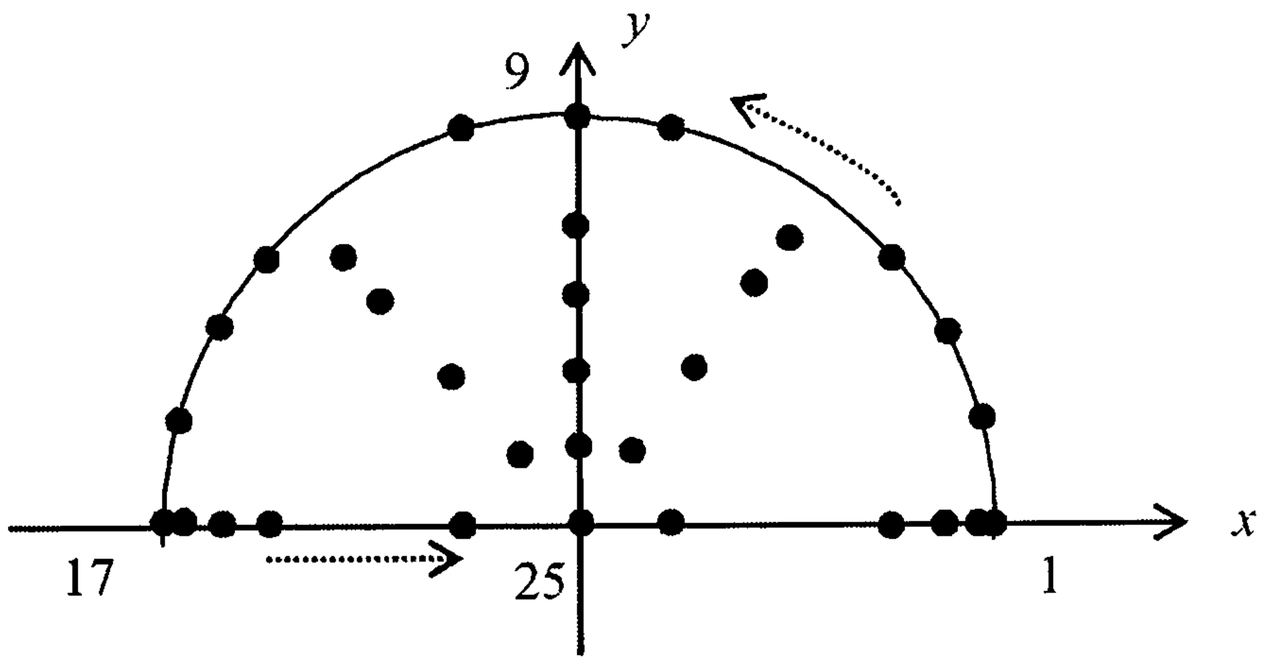

Figure 7.10: Boundary and internal nodes for Example 7.4

In Laplace space this problem is

$$\nabla^2 \bar{u} = \frac{1}{\alpha} \left(\lambda \bar{u} - (x^2 + y^2) \right) + \bar{h}$$

with

$$\bar{h} = \frac{4 + x^2 + y^2}{1 + \lambda}$$

and boundary conditions

$$\bar{u} = \frac{1}{1+\lambda} \text{ on the positive } x\text{-quadrant}$$

$$\bar{q} = \frac{2}{1+\lambda} \text{ on the negative } x\text{-quadrant}$$

$$\bar{q} = 0 \text{ on } y = 1 = 0$$

The analytic solution is again

$$u = (x^2 + y^2)e^{-t}$$

We show this together with the numerical solution in Figures 7.11 and 7.12 and the numerical results for the internal nodes $(0.2, 0.2), (0.4, 0.4), (0.6, 0.6)$ in Table 7.10 and internal nodes $(-0.2, 0.2), (-0.4, 0.4), (-0.6, 0.6)$ in Table 7.11.

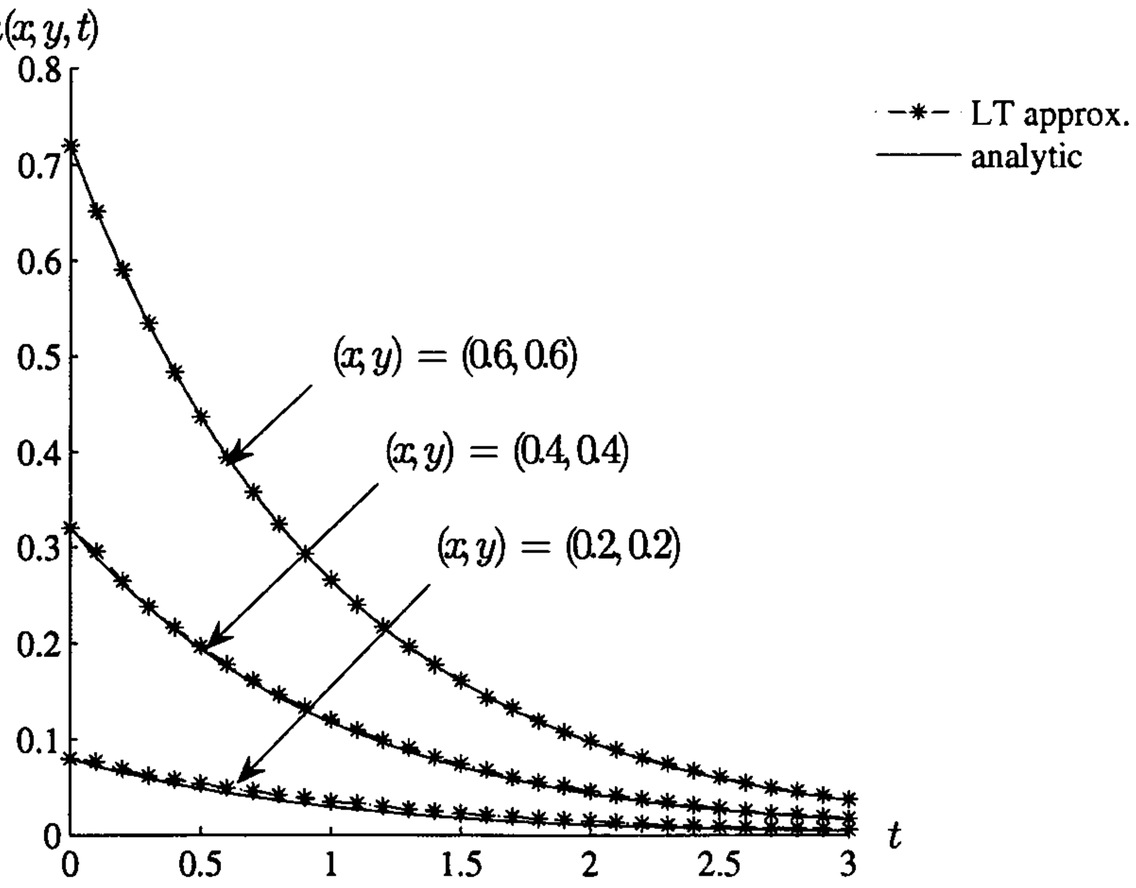

Figure 7.11: Time development of the solution for the positive x-values in Example 7.4

The results for the positive-x internal nodes are once again a very good approximation and although the negative-x errors are not quite as good the approximate solution still tracks the analytic solution very well. The numerical values in the left quadrant are slightly less accurate and this is

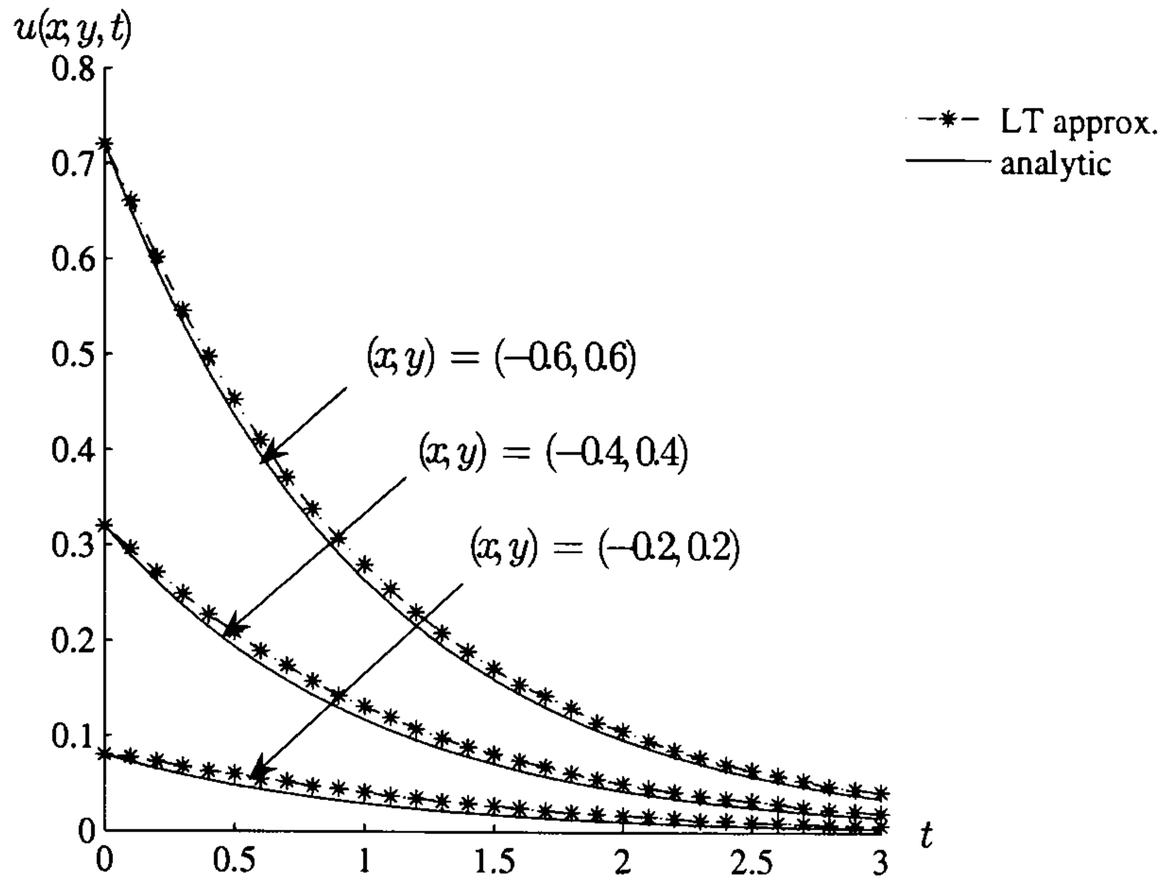

Figure 7.12: Time development of the solution for the negative x-values in Example 7.4

Table 7.10: Analytic and numerical solution for positive x-internal nodes for Example 7.4

time	analytic (0.2, 0.2)	approx (0.2, 0.2)	analytic (0.4, 0.4)	approx (0.4, 0.4)	analytic (0.6, 0.6)	approx (0.6, 0.6)
0.5	0.048522	0.053402	0.194090	0.196540	0.436702	0.436876
1.0	0.029430	0.035363	0.117721	0.120952	0.264873	0.266551
1.5	0.017850	0.022534	0.071402	0.073740	0.160654	0.161731
2.0	0.010827	0.014745	0.043307	0.046017	0.097441	0.097788
2.5	0.006567	0.008973	0.026267	0.028301	0.059101	0.061251
3.0	0.003983	0.006016	0.015932	0.017689	0.035847	0.038235

frequently the case for internal points near a boundary with a Neumann condition.

Table 7.11: Analytic and numerical solution for negative x-internal nodes for Example 7.4

time	analytic $(-0.2, 0.2)$	approx $(-0.2, 0.2)$	analytic $(-0.4, 0.4)$	approx $(-0.4, 0.4)$	analytic $(-0.6, 0.6)$	approx $(-0.6, 0.6)$
0.5	0.048522	0.060304	0.194090	0.209394	0.436702	0.453927
1.0	0.029430	0.041357	0.117721	0.131609	0.264873	0.280956
1.5	0.017850	0.027367	0.071402	0.082014	0.160654	0.172343
2.0	0.010827	0.017471	0.043307	0.050999	0.097441	0.107080
2.5	0.006567	0.010544	0.026267	0.032981	0.059101	0.065899
3.0	0.003983	0.007319	0.015932	0.020265	0.035847	0.042688

Example 7.5

We now consider a steady-state heat flow problem in a cylinder from Toutip (2001). Because of the symmetry in the problem, we can model it in a quarter annulus, using polar coordinates, bounded by the circles with radii $r \equiv \sqrt{x^2 + y^2 = 1}$, $r \equiv \sqrt{x^2 + y^2 = 2}$ and the lines $x = 0$ and $y = 0$, see Figure 7.13.

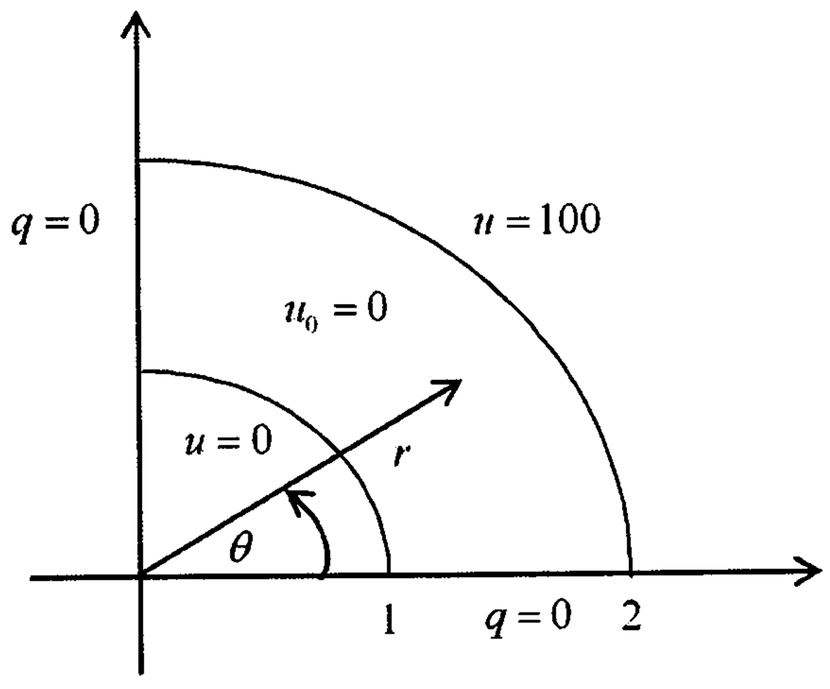

Figure 7.13: Boundary and initial conditions of Example 7.5

We use 40 boundary and 45 internal nodes discretised in the region as shown in Figure 7.14.

We consider two-dimensional transient heat conduction

$$\nabla.(k\nabla u) = \rho c \frac{\partial u}{\partial t} \tag{7.31}$$

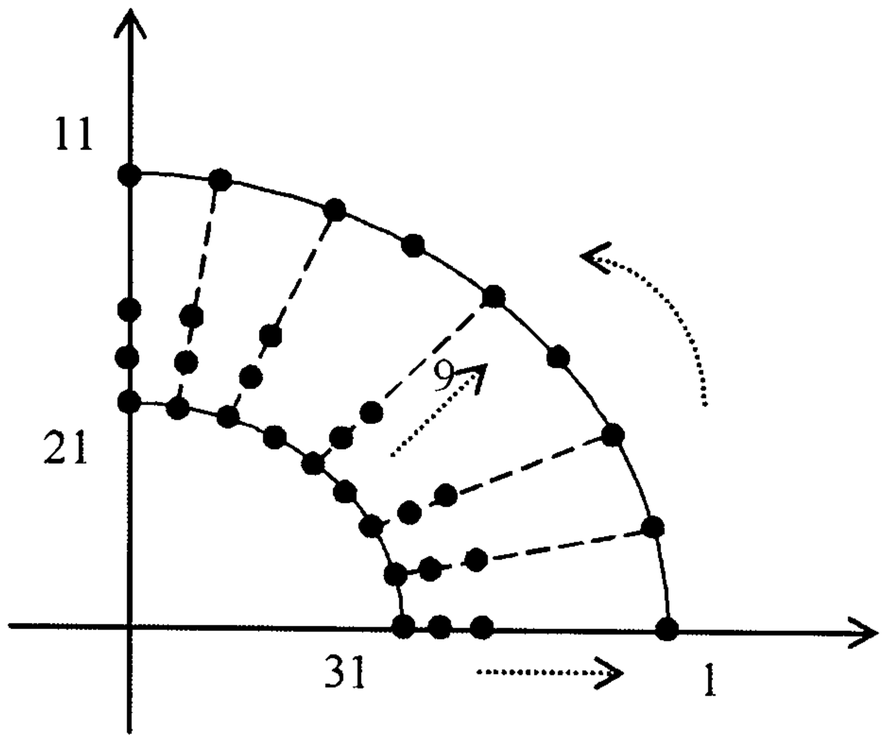

Figure 7.14: Boundary and internal node positions for Example 7.5

where k is the thermal conductivity, ρ is the density and c is the specific heat.

Equation (7.31) can be written in the form

$$k\nabla^2 u + \frac{\partial k}{\partial x}\frac{\partial u}{\partial x} + \frac{\partial k}{\partial y}\frac{\partial u}{\partial y} = \rho c \frac{\partial u}{\partial t}$$

which leads to

$$\nabla^2 u = \frac{1}{k}\left(\rho c \frac{\partial u}{\partial t} - \frac{\partial k}{\partial x}\frac{\partial u}{\partial x} - \frac{\partial k}{\partial y}\frac{\partial u}{\partial y}\right)$$

and, provided that k, ρ and c are independent of u, in Laplace space this becomes

$$\nabla^2 \bar{u} = \frac{1}{k}\left[\rho c\,(\lambda \bar{u} - u_0) - \frac{\partial k}{\partial x}\frac{\partial \bar{u}}{\partial x} - \frac{\partial k}{\partial y}\frac{\partial \bar{u}}{\partial y}\right] \tag{7.32}$$

Suppose that $k = 1$ and $\rho c = 1$. Carslaw and Jaeger (1959) provide an analytic solution with an infinite series of Bessel functions. However computation of the numerical values requires a significant amount of effort and an accurate FDM will serve our purpose. Using $\Delta r = 0.05$ and $\Delta t = 0.001$, we show in Figure 7.15 our results in time with the FDM solution along the lines $r = 1.2,\ 1.5,\ 1.8$. We see that our results match the FDM solution very well. The time curves rise steeply and the steady-state values are becoming clear at time $t = 0.5$.

In Figure 7.16 we show the space distribution for a variety of times.

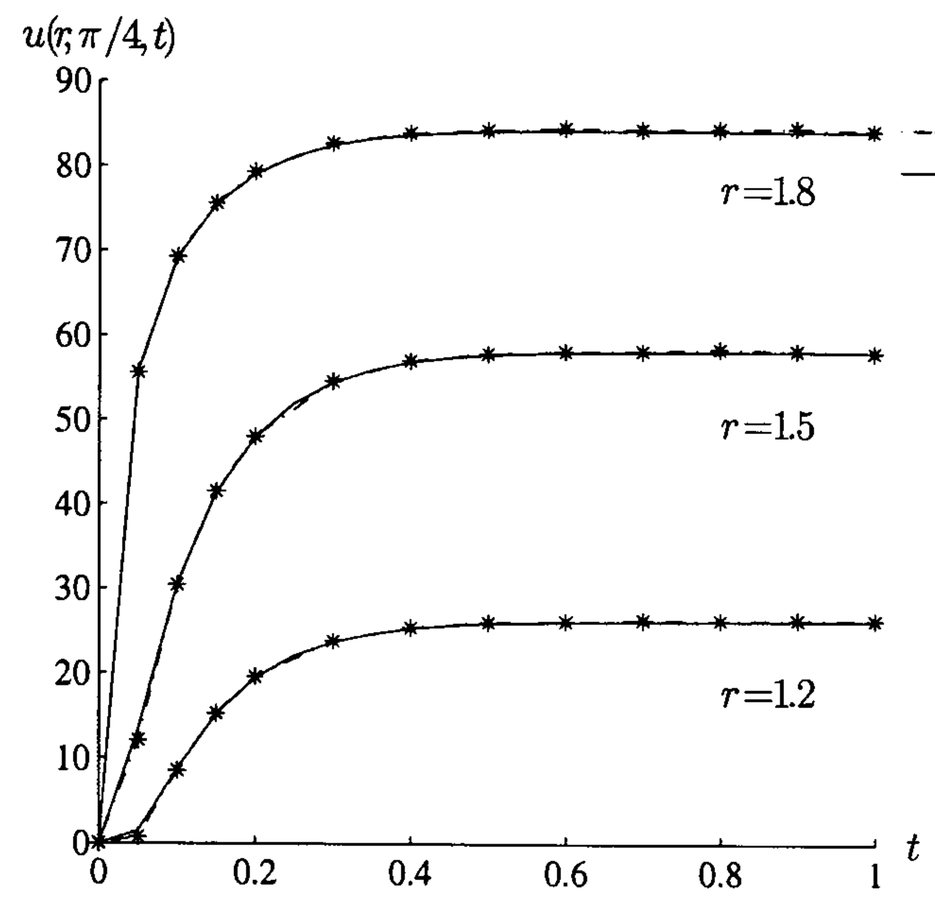

Figure 7.15: The solution of Example 7.5 in time

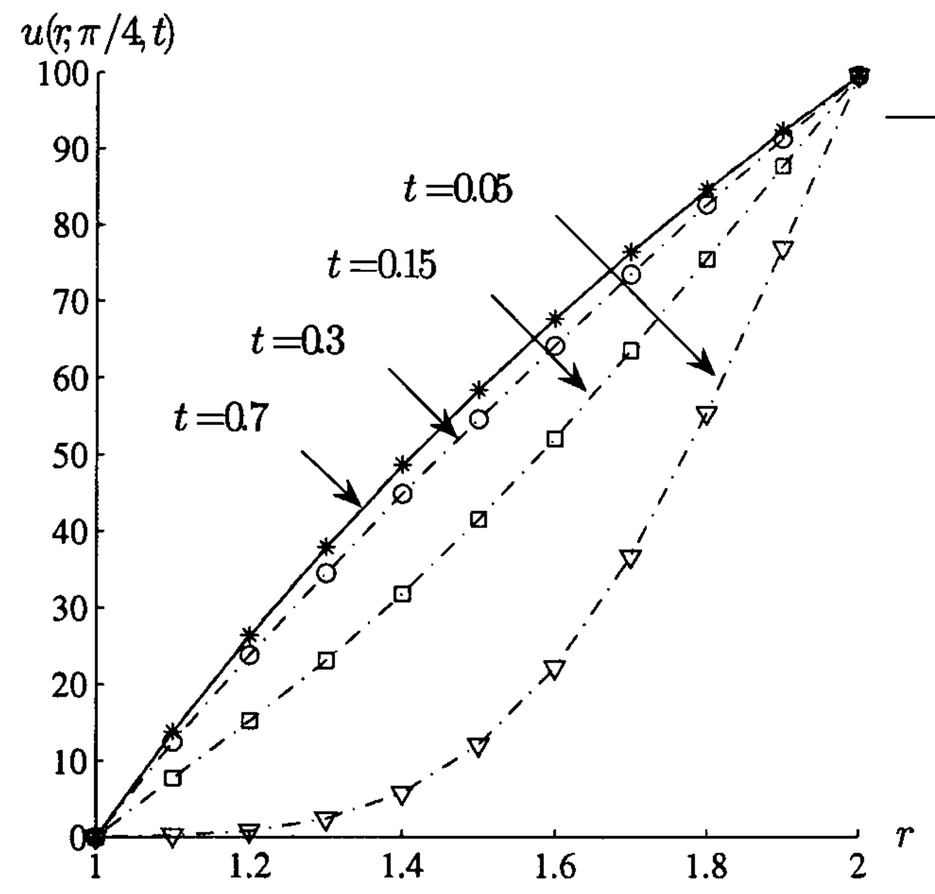

Figure 7.16: The solution of Example 7.5 in space

In the steady state, the analytic solution is

$$u = \frac{100}{\ln 2} \ln r$$

and in Table 7.12 we show our results with the analytic solution for values of r along $\theta = \pi/4$. We also report the errors of the approximation with the analytic value and show that our results are very satisfactory, having less

than 0.5 percent error.

Table 7.12: Steady state analytic and LT approximations for Example 7.5 with $k = 1.0$

r	1.10	1.20	1.30	1.40	1.50	1.60	1.70	1.80	1.90
Analytic	13.75	26.30	37.85	48.54	58.50	67.81	76.55	84.80	92.60
LT approx	13.80	26.35	37.96	48.58	58.60	67.73	76.68	84.80	92.72
LT % error	0.35	0.16	0.29	0.07	0.18	0.11	0.17	0.00	0.13

Example 7.6

This problem models heat conduction in a functionally graded material. In such materials physical properties vary rapidly over short distances thus allowing a smooth transition from one material to another without the possible problems which can occur at the interface between materials of, say, significantly differing thermal conductivity. Typically thermal conductivities vary exponentially in one variable only (Gray *et al.* 2005).

We consider again the problem in Example 7.5 but with an exponential thermal conductivity given by

$$k(x, y) = k(r) = 5e^{3r} = 5e^{3(x^2+y^2)^{\frac{1}{2}}}$$

as shown in Figure 7.17.

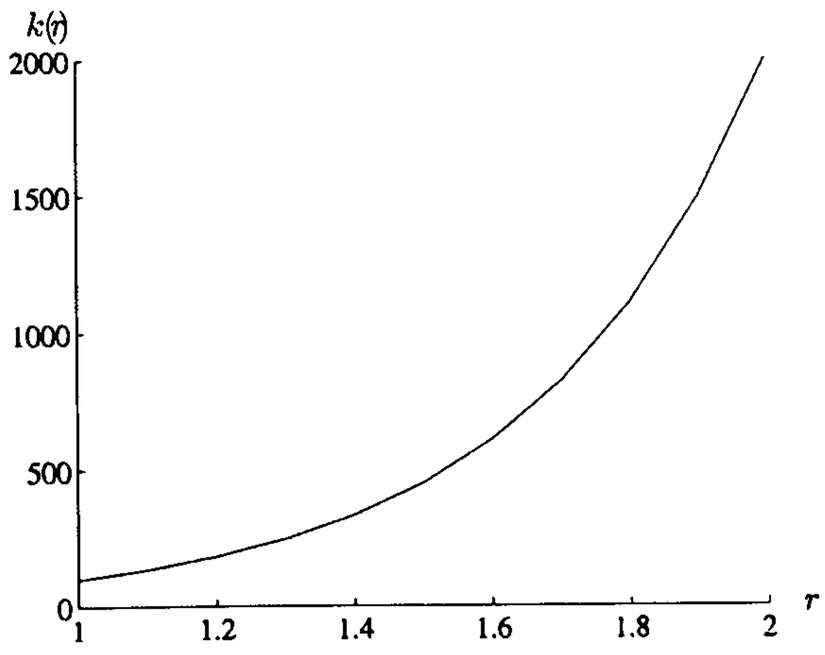

Figure 7.17: Graph of thermal conductivity $k(r)$ for Example 7.6

We see from Figure 7.17 that the thermal conductivity rises very sharply for $r > 1.6$, so that we would expect the temperature in the outer region would reach the steady state more quickly that the inner boundary.

We use equation (7.32) with

$$k = 5e^{3r}, \quad \frac{\partial k}{\partial x} = \frac{15x}{r}e^{3r}, \quad \frac{\partial k}{\partial y} = \frac{15y}{r}e^{3r}, \quad u_0 = 0, \quad \rho c = 1.0$$

There is no analytic solution for the problem and we again compare our results with an accurate FDM solution. This FDM solution, with $\Delta t = 0.000\,001$, shows that at a very short time, $t = 0.0005$, the solution is well on the way to the steady state, see Table 7.13. We would expect the steady state to be reached in a time which is too short for an accurate application of the Laplace transform.

Table 7.13: FDM solution for Example 7.6 at $t = 0.0005$

r	1.10	1.20	1.30	1.40	1.50	1.60	1.70	1.80	1.90
$t = 0.0005$	25.20	44.11	59.04	70.81	79.91	86.77	91.84	95.51	98.14
Steady-state	32.39	54.29	69.21	79.44	86.50	91.39	94.79	97.17	98.83

Consequently we expect our Laplace transform approach would recover the steady state since we would not use such very small values of time; therefore we report only the steady-state values. In Table 7.14 we show the FDM steady-state solution with our Laplace transform approximation and the percentage errors. For interest we also show those values reported by Toutip for his gradient dual reciprocity approach, using the radial basis function $f = 1 + R$, with 40 boundary nodes and 81 internal nodes. We see that our approximations compare very well with those reported by Toutip and with the FDM solutions.

Table 7.14: Steady state LT, FDM and Toutip approximations for Example 7.6 with $k = 5e^{3r}$, together with percentage error

r	1.10	1.20	1.30	1.40	1.50	1.60	1.70	1.80	1.90
LT	33.57	55.97	70.84	80.73	87.83	92.76	95.78	97.84	99.77
FDM	32.39	54.29	69.21	79.44	86.50	91.39	94.79	97.17	98.83
Toutip		55.80		80.79		92.46		98.06	
% error	3.64	3.10	2.36	1.63	1.53	1.50	1.04	0.69	0.95

Example 7.7

All our examples so far in this chapter have been bounded in time, *e.g.* they contain boundary conditions which are negative exponential in time. To test our method on a problem whose solution is increasing in time we consider the following

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} + h$$

where

$$h = 2t^2 - 2x^2 t$$

in the unit square with boundary and initial conditions as in Figure 7.18 and we use $\alpha = 1$.

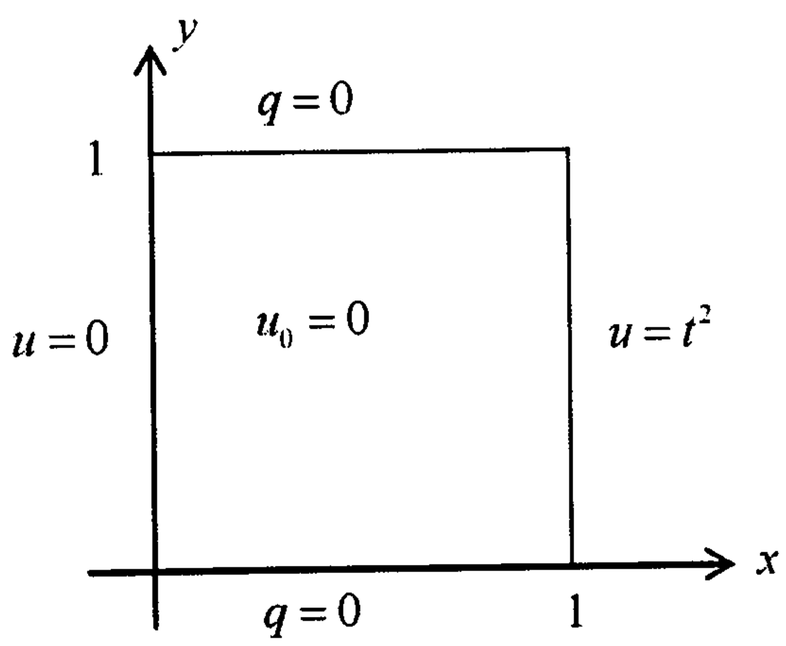

Figure 7.18: Boundary and initial conditions for Example 7.7

In Laplace space the problem is defined as

$$\nabla^2 \bar{u} = \frac{1}{\alpha}(\lambda \bar{u} - u_0) + \frac{4}{\lambda^3} - \frac{2x^2}{\lambda^2}$$

with boundary conditions

$$\bar{u} = 0 \text{ on } x = 0, \quad \bar{u} = \frac{2}{\lambda^3} \text{ on } x = 1 \quad \text{and}$$

$$\bar{q} = 0 \text{ on } y = 1 \text{ and } y = 1$$

The solution is shown in Figure 7.19 and we see that the approximation tracks the analytic solution, $u = x^2t^2$, very well.

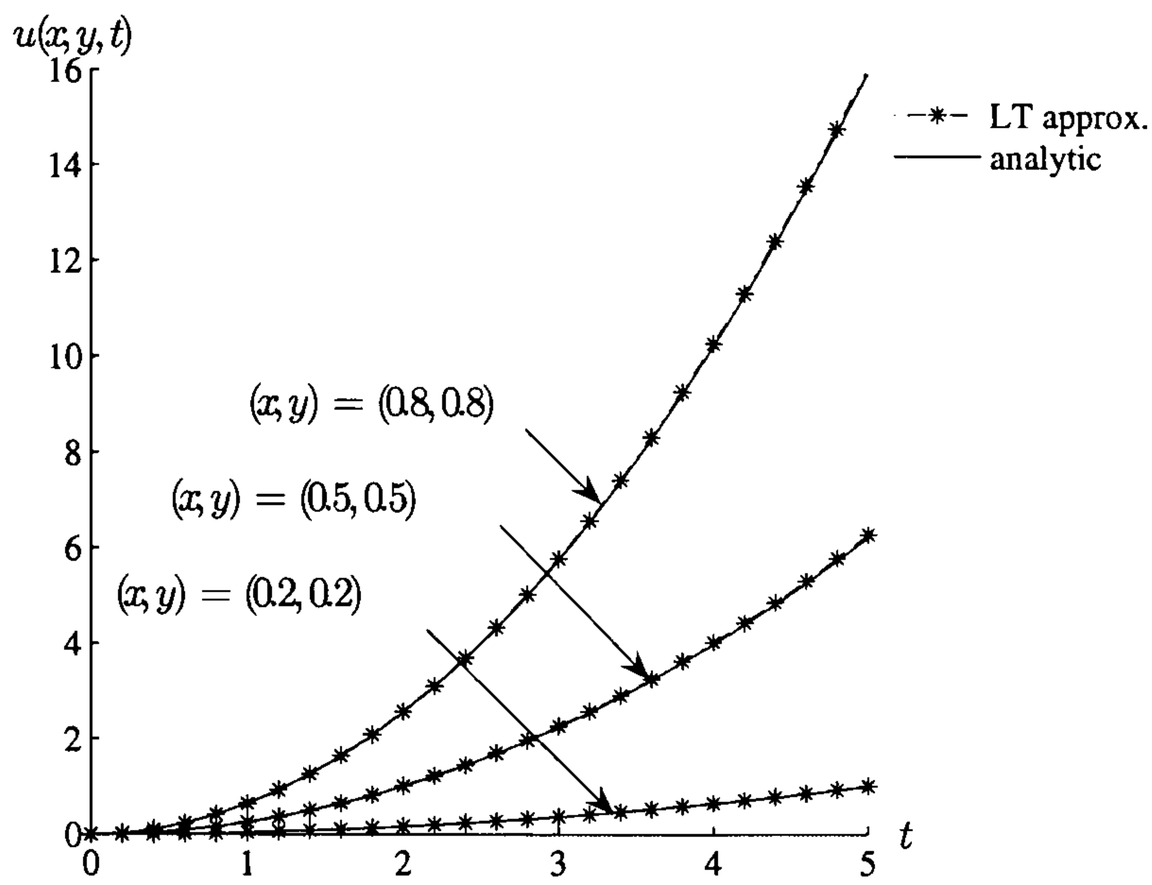

Figure 7.19: Time development of the solution for Example 7.7

The numerical results for the internal values at nodes $(0.2, 0.2), (0.5, 0.5), (0.8, 0.8)$ together with the analytic values are shown in Table 7.15 and the percentage errors in Table 7.16.

We see from Tables 7.15 and 7.16 that the solutions are very good, with maximum error less than three percent.

Table 7.15: Analytic and numerical solution for Example 7.7

time	analytic $x = 0.2$	approx $x = 0.2$	analytic $x = 0.5$	approx $x = 0.5$	analytic $x = 0.8$	approx $x = 0.8$
0.5	0.010000	0.010171	0.062500	0.063035	0.160000	0.161266
1.0	0.040000	0.040886	0.250000	0.252535	0.640000	0.645159
1.5	0.090000	0.092085	0.562500	0.568471	1.440000	1.451893
2.0	0.160000	0.163914	1.000000	1.011094	2.560000	2.582508
2.5	0.250000	0.256187	1.562500	1.580294	4.000000	4.035404
3.0	0.360000	0.369073	2.250000	2.275608	5.760000	5.809252
3.5	0.490000	0.502280	3.062500	3.097056	7.840000	7.908508
4.0	0.640000	0.656846	4.000000	4.044423	10.240000	10.324660
4.5	0.810000	0.831822	5.062500	5.120780	12.960000	13.073050
5.0	1.000000	1.025558	6.250000	6.322110	16.000000	16.141930

Table 7.16: Percentage errors for Example 7.7

time	$x = 0.2$	$x = 0.5$	$x = 0.8$
0.5	1.71	0.86	0.08
1.0	2.21	1.01	0.08
1.5	2.32	1.06	0.08
2.0	2.45	1.11	0.09
2.5	2.47	1.14	0.09
3.0	2.52	1.14	0.09
3.5	2.51	1.13	0.09
4.0	2.63	1.11	0.08
4.5	2.69	1.15	0.09
5.0	2.56	1.15	0.09

Example 7.8

This example is modified from the Motz problem (Motz 1946) which is often used as a test for a new elliptic solution method as it has a singularity on the boundary. We seek the solution to the problem

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t}$$

The geometry and boundary conditions are defined as follows, see Figure 7.20,

$$u(x, y, t) = 500 \text{ on } y = 0 \text{ for } 0 \leq x \leq 7,$$

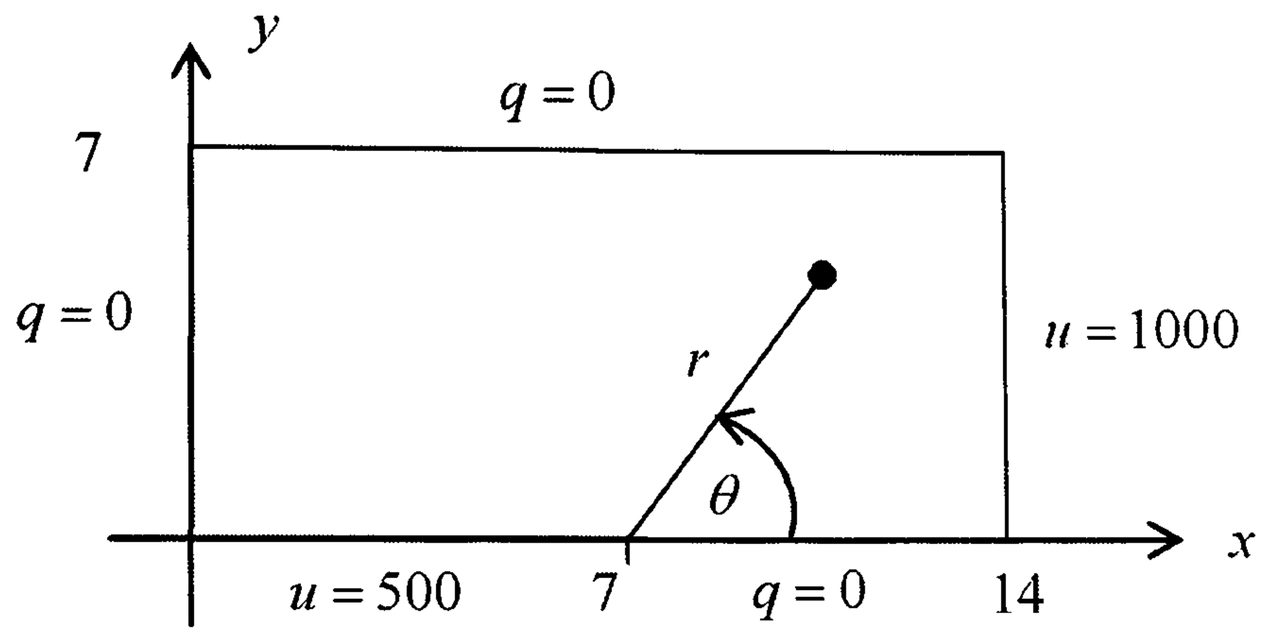

Figure 7.20: Boundary and initial conditions for Example 7.8

$$q(x, y, t) = 0 \text{ on } y = 0 \text{ for } 7 < x < 14$$

$$u(x, y, t) = 1000 \text{ on } x = 14,$$

$$q(x, y, t) = 0 \text{ on } x = 0 \text{ and } y = 7$$

and the initial condition is

$$u_0(x, y) = 0$$

Since $u_0 = 0$ we could use the approach of Chapter 6. However we wish to use this problem, which has a geometric singularity at $(7.0, 0.0)$ and a known analytic solution in the region of the singularity, as a test of our use of the dual reciprocity method with the Laplace transform method.

In Laplace space the boundary conditions are

$$\bar{u} = 500/\lambda \text{ on } y = 0 \text{ for } 0 \leq x \leq 7$$

$$\bar{q} = 0 \text{ on } y = 0 \text{ for } 7 < x < 14$$

$$\bar{u} = 1000/\lambda \text{ on } x = 14$$

$$\bar{q} = 0 \text{ on } x = 0 \text{ and } y = 7$$

We discretise the boundary into 56 elements, including the singular point O with 9 internal nodes, see Figure 7.21.

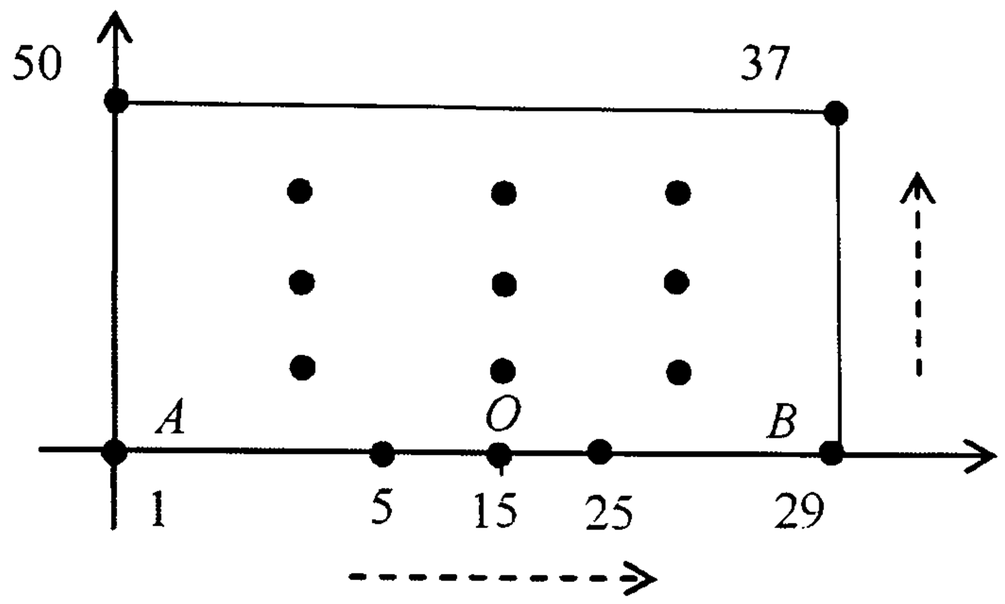

Figure 7.21: Boundary and internal nodes for Example 7.8

We solve the problem using the augmented thin plate spline for the interpolating function in the dual reciprocity and $\alpha = 1$. However following our investigation earlier when we considered a problem with a larger geometry, we scale the problem by a factor of 14 so that the problem domain becomes $0 \leq x \leq 1, 0 \leq y \leq 0.5$.

We show the time development of our solution in Figure 7.22 for time values between 0 and 1. We see that the solutions for the internal nodes and the one boundary node follow smooth curves obtaining the steady-state values by the time $t = 0.4$.

In Figure 7.23 we show the approximate solutions along the boundary $7 \leq x \leq 14$, $y = 0$ at various time values and see once again that the steady state is reached in a small time frame.

In Table 7.17 we compare our steady-state solution with those reported by Toutip, with the BETIS programme (París and Cañas 1997) and Symm (1973). The symbol *** in the table means that there is no solution from that reference. Our results compare very favourably with those using other methods of solution.

In the steady state the analytic solution in the neighbourhood of the singular point $(7, 0)$ in polar form is

$$u(r,\theta) = a_0 + a_1 r^{\frac{1}{2}} \cos\left(\frac{\theta}{2}\right) + a_2 r^{\frac{3}{2}} \cos\left(\frac{3\theta}{2}\right) + a_3 r^{\frac{5}{2}} \cos\left(\frac{5\theta}{2}\right) + \ldots \quad (7.33)$$

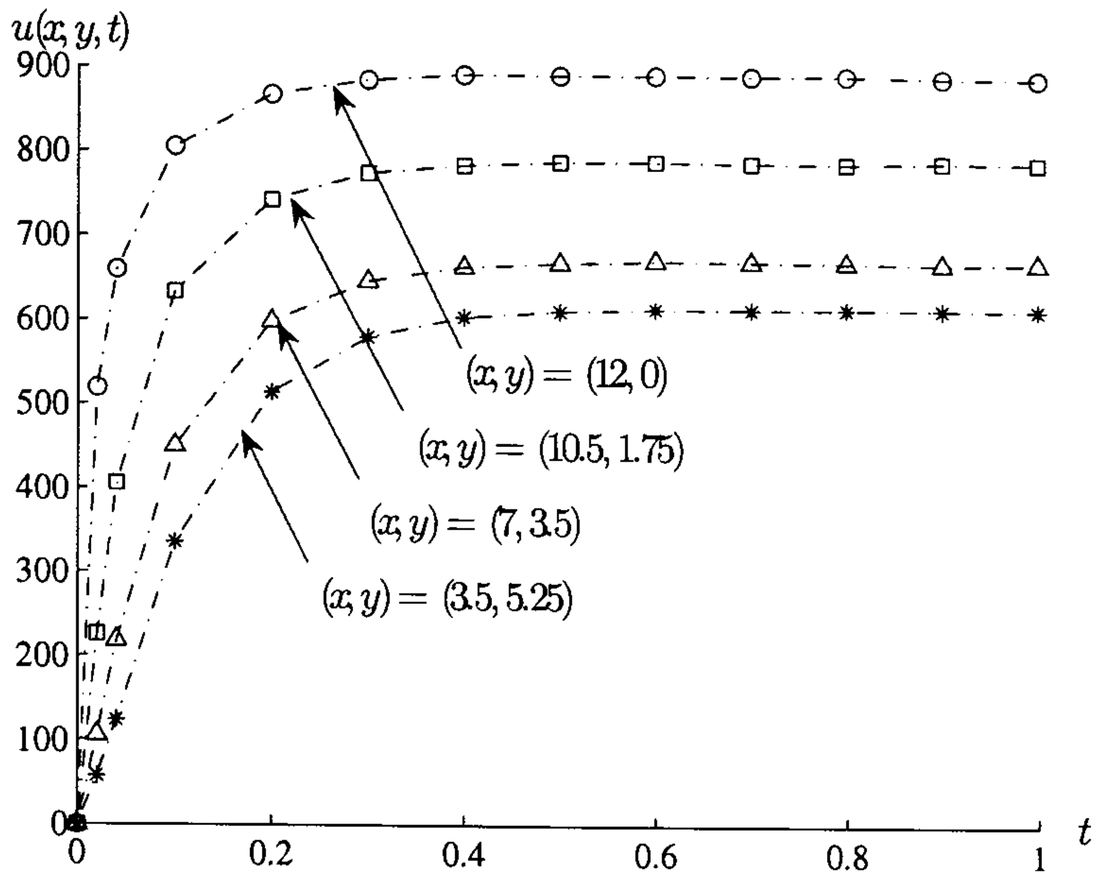

Figure 7.22: Time development of the solution for Example 7.8

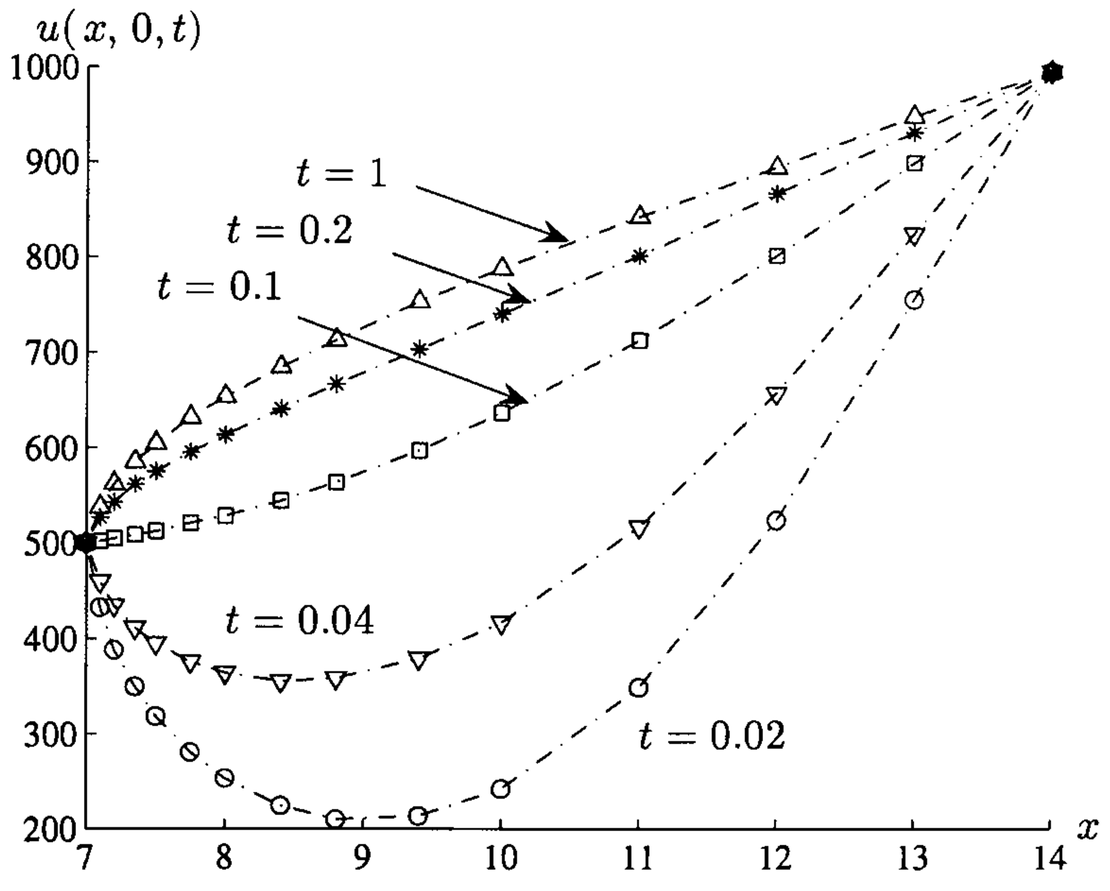

Figure 7.23: Space development of the solution for Example 7.8

where r is the distance from $(7,0)$ and θ is measured counter-clockwise from the line $y = 0, x > 7$. Whiteman and Papamichael (1972) showed that the first two a_i are given by

$$a_0 = 500, \ a_1 = 151.625$$

We test the accuracy of our results near the point $(7,0)$ to approximate the

Table 7.17: Steady state solution for Example 7.8

position from 'O' +7.0	LT	BETIS	Symm	Toutip
0.1	545.5	544.0	***	537.1
0.2	565.3	565.3	***	561.5
0.35	589.1	588.4	***	585.9
0.5	607.4	607.0	608.9	605.1
0.75	632.8	632.9	634.4	631.5
1.0	655.2	655.2	656.5	654.1
1.4	685.4	686.4	***	685.7
1.8	715.2	714.5	***	713.9
2.4	755.0	752.8	***	752.6
3.0	788.9	788.3	788.9	788.3
4.0	844.7	844.0	844.4	844.4
5.0	897.6	897.1	897.3	898.0
6.0	950.5	948.8	948.9	951.3

coefficients a_0 and a_1 in equation (7.33) with those from the references. We refine the boundary mesh to take into account the additional values $x = 7.01, 7.02, 7.03, 7.04, 7.05, 7.06$ and 7.08 and obtain the results in Table 7.18.

Table 7.18: Solutions for Example 7.8 for small values of r

x-value 7.0+	approx
0	499.949
0.01	509.8669
0.02	516.7933
0.03	522.5395
0.04	526.3172
0.05	530.8447
0.06	533.1944
0.08	539.8089
0.1	545.5303

For small r we should have, along OB, $u \approx a_0 + a_1 r^{\frac{1}{2}}$. Applying the least squares method to fit the function to the data in Table 7.17 we obtain the coefficients

$$a_0 = 497 \text{ and } a_1 = 148$$

and these are in good agreement with the coefficients given previously.

7.4 Summary of Chapter 7

In this chapter we have looked at a variety of problems to test the Laplace transform boundary element method using dual reciprocity for the non-homogeneity. We have shown that it behaves in a robust fashion and our results have been very satisfactory for boundary conditions which are monotonic in time. We have also shown that the method does not always work well for very small values of time and that for large geometries a suitable scaling is necessary. Contrary to some authors' suggestions, we have not found it necessary to have the number of internal nodes greater than half the number of boundary nodes.

We have used $f = 1 + R$ and augmented thin plate splines for the radial basis functions in the dual reciprocity method and found that, when both bases are available for use, *i.e.* one without a $\partial u/\partial x$ or $\partial u/\partial y$ term, the augmented thin plate spline gives the better approximation. Future work will include modifying the present code to handle terms in ∇u for augmented thin plate splines and also to consider other radial basis functions, particularly those which will accommodate the second derivative.

Chapter 8

Problems with non-monotonic time-dependent boundary conditions

8.1 Introduction

In Chapter 7 we augmented the Laplace transform boundary element method with the additional scheme of the dual reciprocity method for the solution of linear Poisson-type problems and saw that when the boundary conditions are monotonic in time the process recovers good solutions. However, difficulties can occur if the boundary conditions are not monotonic in time. Stehfest says "In the following, the term 'smooth' is used to express that the rate of convergence is sufficiently great. An oscillating $F(t)$ certainly is not smooth enough unless the wavelength of the oscillations is large".

He also says that "No accurate results are expected, too, if $F(t)$ has discontinuities near T."

In Section 8.2 we shall consider problems with discontinuous boundary conditions and in Section 8.3 we shall consider problems with periodic boundary conditions.

Consider again the initial boundary-value problem defined in the two-dimensional region, D, bounded by the closed curve $C = C_1 + C_2$ from Section 7.2:

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \quad \text{in } D \tag{8.1}$$

subject to the boundary conditions

$$u = u_1(x, y, t) \quad \text{on } C_1 \tag{8.2}$$

$$q \equiv \frac{\partial u}{\partial n} = q_2(x, y, t) \quad \text{on } C_2 \tag{8.3}$$

and the initial condition

$$u(x, y, 0) = u_0(x, y) \tag{8.4}$$

We define the Laplace transform in the usual way so that the initial boundary-value problem becomes

$$\nabla^2 \bar{u} = \frac{1}{\alpha}(\lambda \bar{u} - u_0) \quad \text{in } D \tag{8.5}$$

subject to

$$\bar{u} = \bar{u}_1 \quad \text{on } C_1 \tag{8.6}$$

$$\bar{q} = \bar{q}_2 \quad \text{on } C_2 \tag{8.7}$$

8.2 Problems with discontinuous boundary conditions

Suppose that the time-dependent boundary conditions, equations (8.2) and (8.3) are discontinuous at $t = T$ *e.g.* (Crann and Davies 2004a)

$$u_1(x, y, t) = \begin{cases} u_{1,1}(x, y, t) & 0 < t < T \\ u_{1,2}(x, y, t) & t > T \end{cases} \tag{8.8}$$

$$q_2(x,y,t) = \begin{cases} q_{2,1}(x,y,t) & 0 < t < T \\ q_{2,2}(x,y,t) & t > T \end{cases} \tag{8.9}$$

We apply the Laplace transform piecewise in time and seek solutions

$$\begin{matrix} u^{(1)}(x,y,t) & 0 < t < T \\ u^{(2)}(x,y,t) & t > T \end{matrix} \tag{8.10}$$

by solving as follows:

$$\nabla^2 u^{(1)} = \frac{1}{\alpha}\frac{\partial u^{(1)}}{\partial t} \quad \text{in } D, \quad 0 < t \leq T$$

subject to the boundary conditions

$$u^{(1)} = u_{1,1}(x,y,t) \text{ on } C_1 \text{ and } q_2^{(1)} = q_{2,1}(x,y,t) \text{ on } C_2$$

and the initial condition

$$u^{(1)}(x,y,0) = u_0(x,y)$$

We now use $u^{(1)}(x,y,T)$ as the initial value to find the solution for $t > T$:
We transform the time variable, $t \to t - T$

$$\nabla^2 u^{(2)} = \frac{1}{\alpha}\frac{\partial u^{(2)}}{\partial t} \text{ in } D, \quad t > 0$$

subject to the boundary conditions

$$u_1^{(2)} = u_{1,2}(x,y,t) \text{ on } C_1 \text{ and } q_2^{(2)} = q_{2,2}(x,y,t) \text{ on } C_2$$

and the initial condition

$$u^{(2)}(x,y,0) = u^{(1)}(x,y,T)$$

Example 8.1

The problem in this example is the partial differential equation (8.1) defined in the unit square $\{(x,y) : 0 < x < 1,\, 0 < y < 1\}$.

The boundary conditions are defined as follows, see Figure 8.1:

$$u(0, y, t) = 10$$

$$q(x, 0, t) = q(x, 1, t) = 0$$

$$u(1, y, t) = 20 + 10H(t - 1)$$

$$= \begin{cases} 20 & 0 \leq t \leq 1 \\ 30 & t > 1 \end{cases}$$

where $H(t-1)$ is the Heaviside unit step function and the initial condition is

$$u_0(x, y) = 20$$

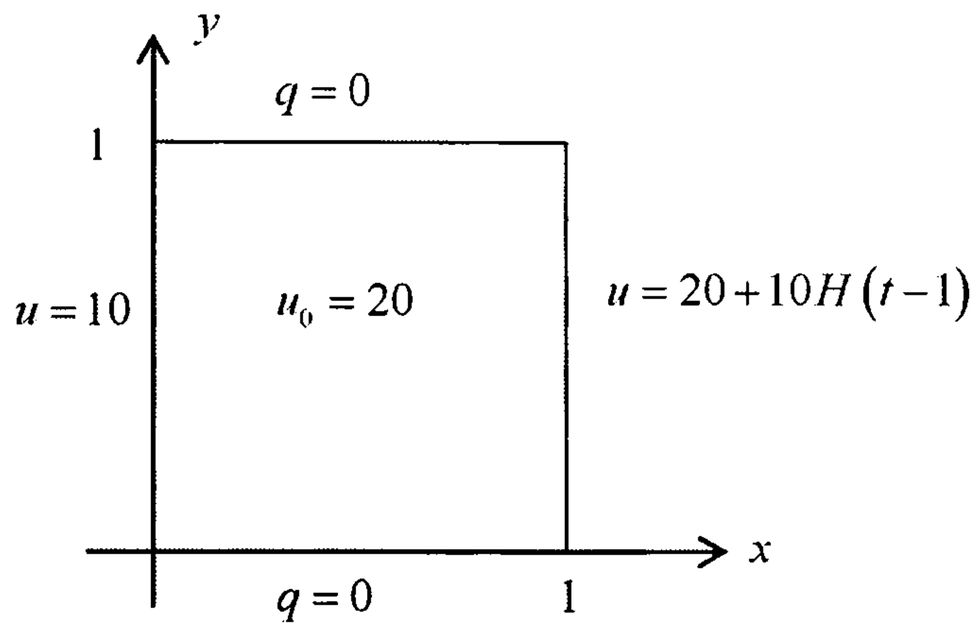

Figure 8.1: Boundary and initial conditions for Example 8.1

We solve the problem with $\alpha = 1.0$ and use $N = 32$ boundary points and $L = 9$ internal points. Also, in the dual reciprocity method, we use augmented thin plate splines for the basis functions in equation (7.11). For the numerical Laplace transform we use the Stehfest parameter value $M = 8$.

We first solve the problem using a single application of the Laplace transform. We refer to this solution as the Full LT solution. The boundary conditions transform to

$$u(0, y; \lambda) = \frac{10}{\lambda}$$

$$q(x, 0; \lambda) = q(x, 1; \lambda) = 0$$

$$u(1, y; \lambda) = \frac{20}{\lambda} + \frac{10}{\lambda} e^{-\lambda}$$

The solution is shown in Figure 8.2.

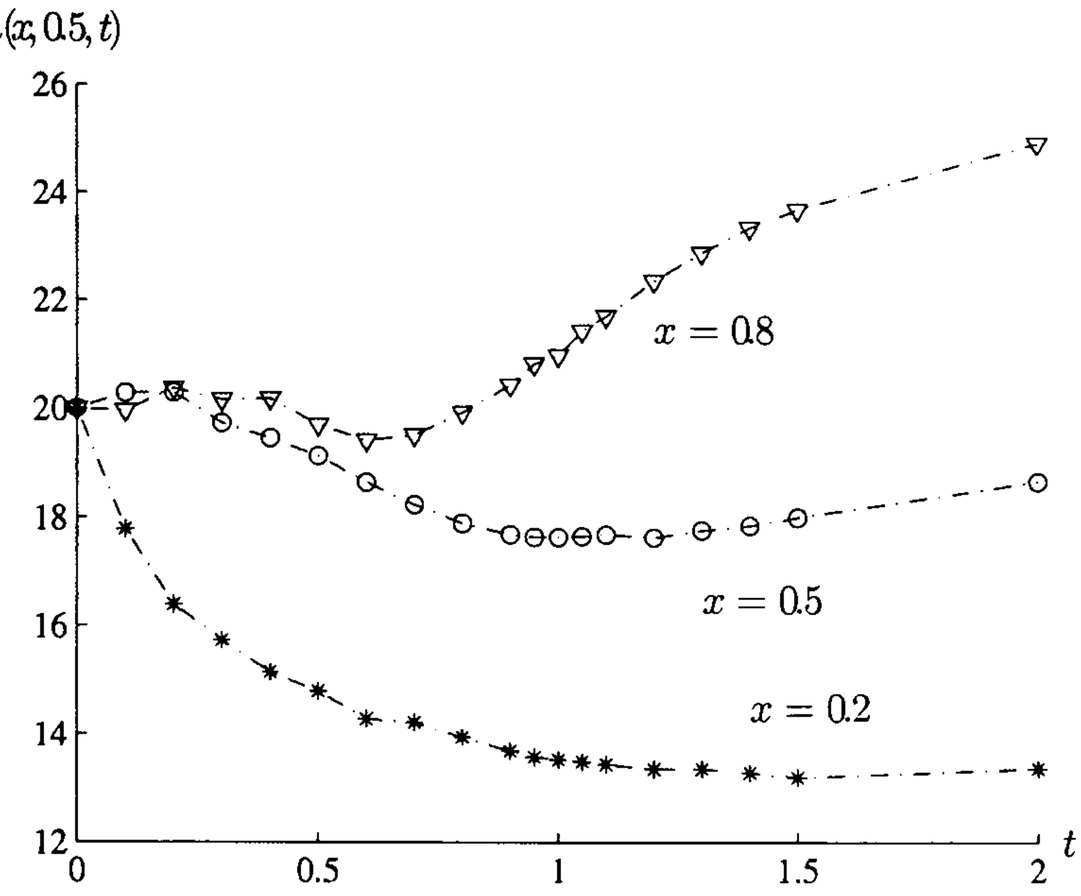

Figure 8.2: Full Laplace transform solution for $0 \leq t \leq 2$ in Example 8.1

We see that the approximate solutions at the three $x-$values are smooth and the effect of the discontinuity has been lost. This is as predicted by Stehfest.

We now solve the problem using the piecewise application of the Laplace transform and use our experience from the ordinary differential problem in Section 5.4. We refer to this as the Step LT solution. The boundary conditions transform to

$$\bar{u}(0, y; \lambda) = 10/\lambda$$

$$\bar{q}(x, 0; \lambda) = \bar{q}(x, 1; \lambda) = 0$$

$$\bar{u}(1, y; \lambda) = \begin{cases} \dfrac{20}{\lambda} & 0 \leq \quad t \leq 1 \\ \dfrac{30}{\lambda} & \qquad t > 1 \end{cases}$$

The solution is shown in Figure 8.3.

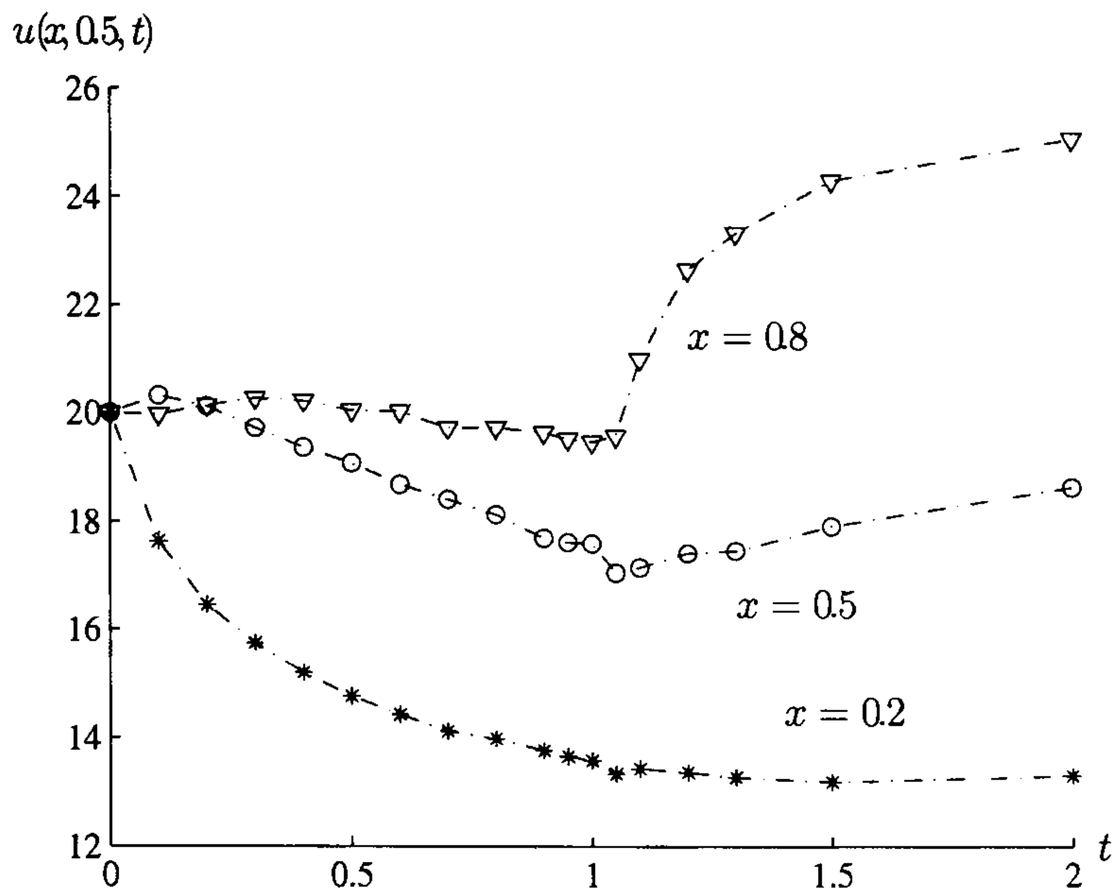

Figure 8.3: Step Laplace transform solution for $0 \leq t \leq 2$ in Example 8.1

We see now the effect of the discontinuity at $t = 1.0$ for all values of x. In both cases, Full and Step, the Laplace transform approach yields a relatively poor solution for small values of the time variable as we have already seen. This is a common problem associated with the numerical Laplace transform approach. However for values of t away from $t = 0$ the solution is in general very accurate. We notice from Figures 8.2 and 8.3 that the two approaches differ significantly in the region of $t = 1$, *i.e.* in the neighbourhood of the discontinuity. In order to investigate this behaviour we compare the results with an accurate explicit finite difference solution with $\Delta x = 0.1$ and $\Delta t = 0.01$. In Figure 8.4 we compare the two Laplace transform approaches with the finite difference solution for the two cases $x = 0.2$ and $x = 0.8$.

We now see that the Step LT solution tracks the FDM solution very well. It is obvious that the Full LT solution has been smoothed out and the approximation is not reasonable until t is approximately 2.

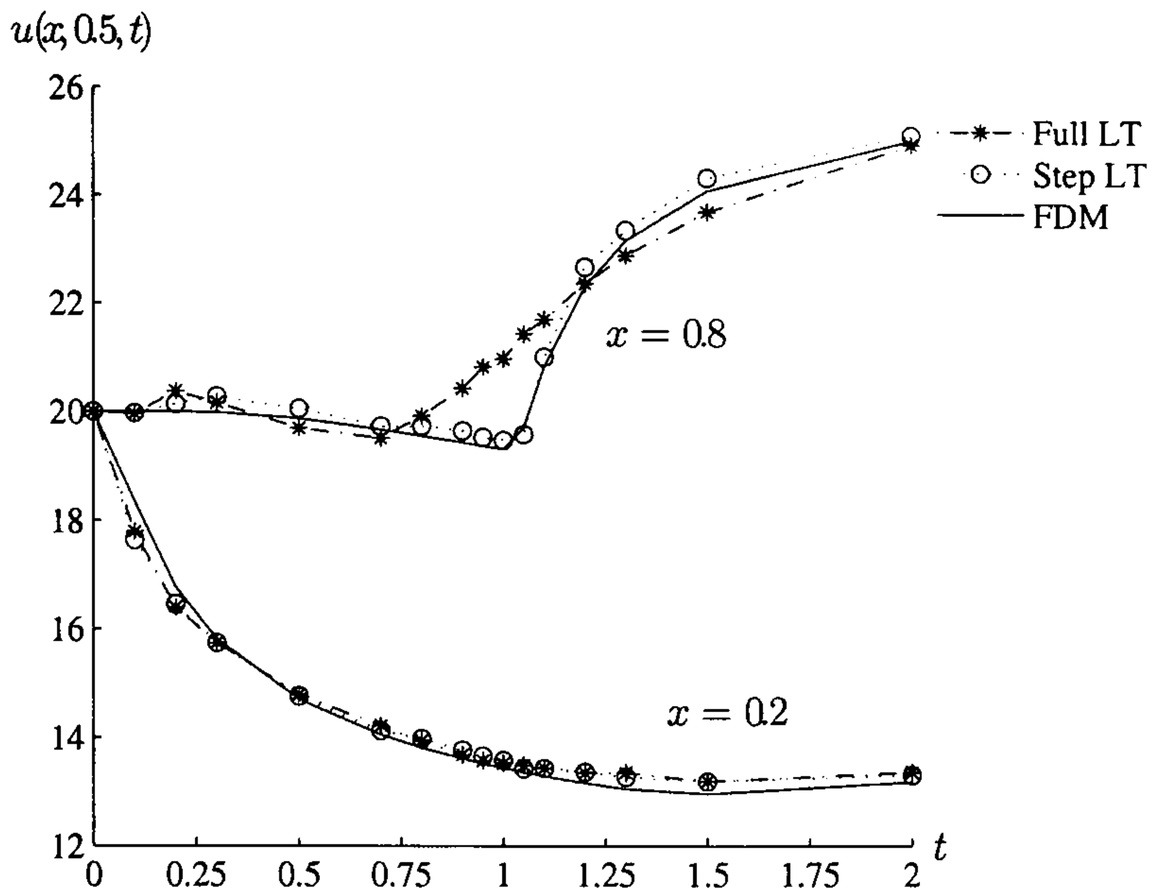

Figure 8.4: Comparison of the two Laplace transform solutions with the finite difference solution in Example 8.1

Example 8.2

This example is a similar problem with a discontinuous boundary condition but in a different geometry.

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} \quad \text{in } D \tag{8.11}$$

$$u(x, y, t) = 0 \text{ on the negative-}x \text{ quadrant}$$

and

$$\begin{aligned} u(x, y, t) &= 1 + H(t-1) \\ &= \begin{cases} 1 & 0 \leq t \leq 1 \\ 2 & t > 1 \end{cases} \end{aligned}$$

where $H(t-1)$ is the Heaviside unit step function and the initial condition is

$$u_0(x, y) = 0$$

The partial differential equation (8.1) is defined in a circle, radius 1.0, although due to the symmetry of the problem we can work in the upper half-plane, see Figure 8.5.

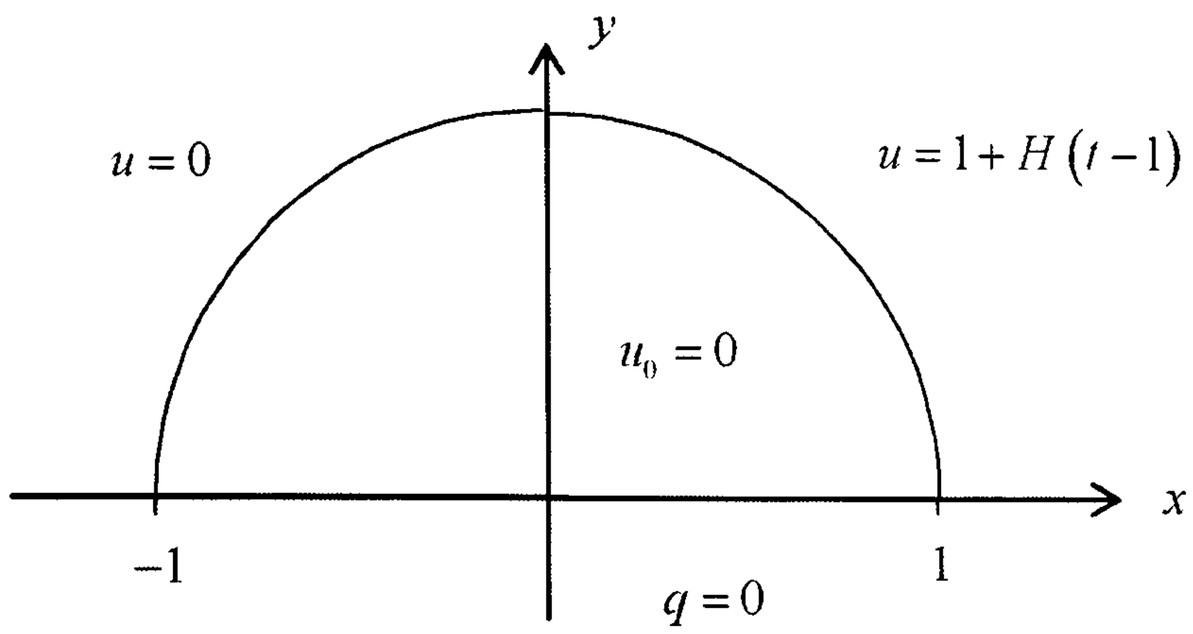

Figure 8.5: Boundary and initial conditions for Example 8.2

We solve the problem using the Step, piecewise, Laplace transform. The boundary conditions transform to

$$\bar{q}(x, 0; \lambda) = 0$$

$$\bar{u}(x, y; \lambda) = 0 \text{ on the negative-}x \text{ quadrant}$$

$$\bar{u}(1, y; \lambda) = \begin{cases} \frac{1}{\lambda} & 0 \leq t \leq 1 \\ \frac{2}{\lambda} & t > 1 \end{cases} \text{ on the positive-}x \text{ quadrant}$$

The solution for five internal nodes over time is shown in Figure 8.6. We can see that the solution is as we would expect with the discontinuity at $t = 1.0$ being very obvious. The solutions at the five nodes reach their first, local, steady-state values by about $t = 0.5$ and then at $t = 1$ take the step and are near to their full steady-state values after a further time of 0.5. For the solution at $(-0.65, 0.65)$, being so near to the left-hand quadrant, the approximation is dominated by the boundary values on the left-hand quadrant and the step at $t = 1$ is barely noticeable, while at the internal node $(0.65, 0.65)$ the solution quickly approaches its steady state, a value close to 1. It then jumps dramatically, approaching its steady state solution very quickly. The other three values show an intermediate behaviour as we would expect.

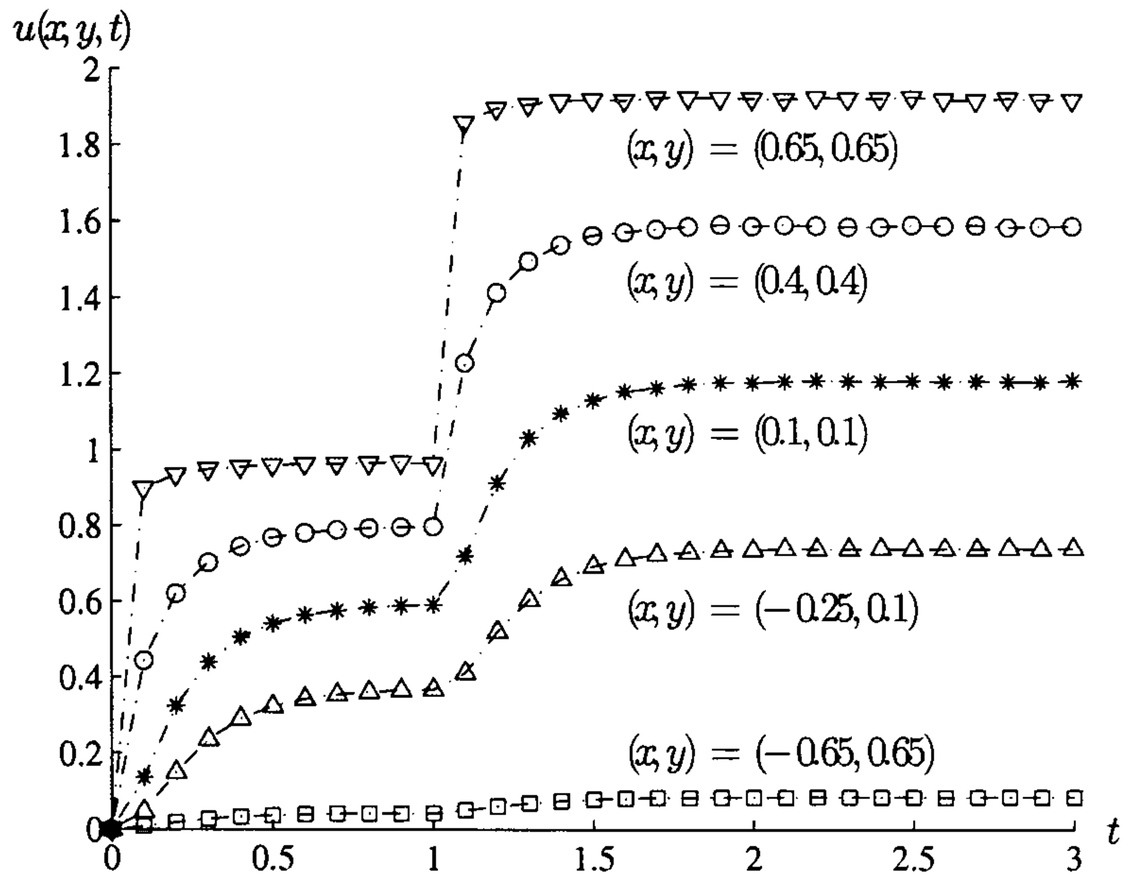

Figure 8.6: Time development of the solution for Example 8.2 for five points in the time period $0.1, \ldots, 3.0$

In Figure 8.7 we see the space discretisation of the solution. It clearly shows how the values along the radius r at $\theta = \pi/4$ approach the local then global steady-state values.

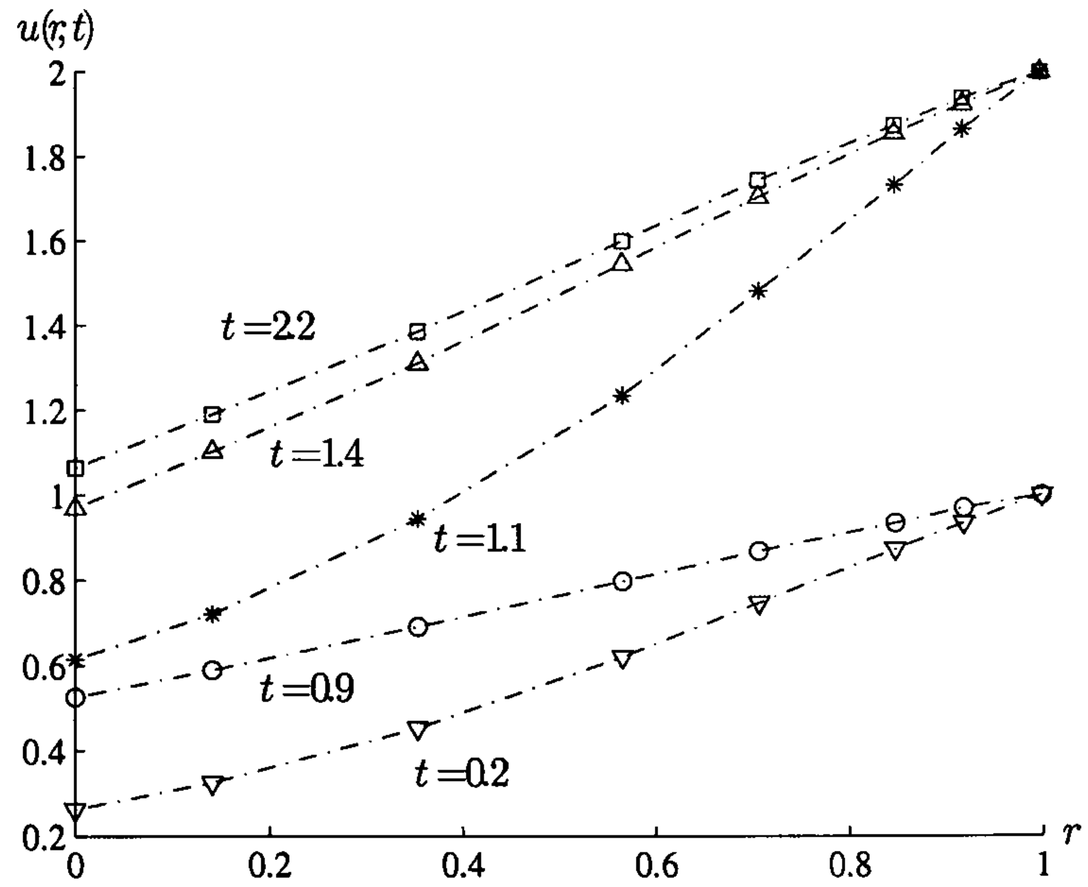

Figure 8.7: Space distribution of the solution for Example 8.2 for five time values for r at $\theta = \pi/4$

Example 8.3

This example is again a Step problem but with a discontinuity in the problem, not the boundary condition. We seek the solution to the problem

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} - 100H(t-1)$$

where $H(t-1)$ is the Heaviside unit step function. For this problem $\alpha = 0.1$. The geometry and boundary conditions are defined as follows, see Figure 8.8,

$$u(x,y,t) = 0 \text{ on } x = 0$$

$$u(x,y,t) = 2 \text{ on } x = 1$$

$$q(x,y,t) = 0 \text{ on } y = 0 \text{ and } y = 1$$

and the initial condition is

$$u_0(x,y) = 0$$

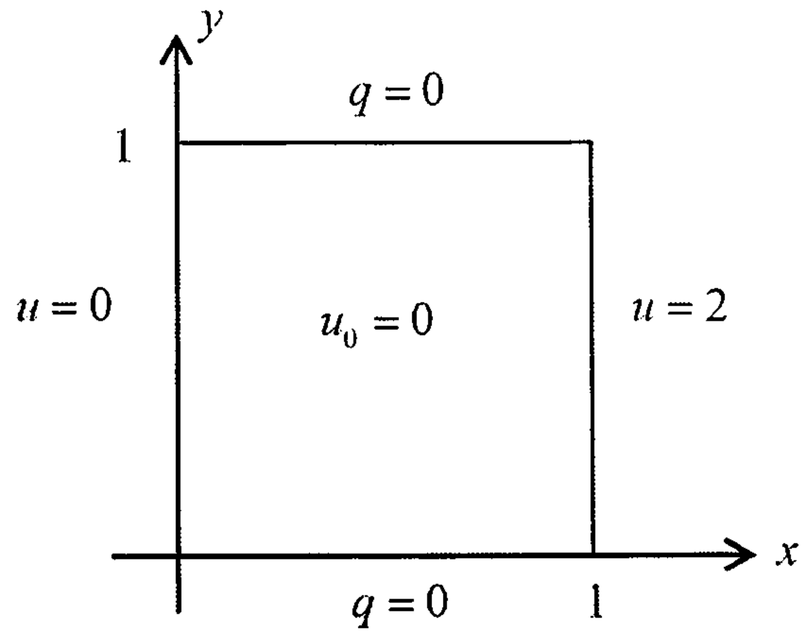

Figure 8.8: Boundary and initial conditions for Example 8.3

For the Step solution process the problem in Laplace space is

$$\nabla^2 \bar{u} = \frac{1}{\alpha}(\lambda\bar{u} - u_0) - \begin{cases} 0 & t < 1 \\ \frac{100}{\lambda} & t \geq 1 \end{cases}$$

where the initial condition, u_0 for $t > 1$ are the values of $\bar{u}$ at $t = 1$.

The boundary conditions are

$$\bar{u} = 0 \text{ on } x = 0$$

$$\bar{u} = \frac{2}{\lambda} \text{ on } x = 1$$

$$\bar{q} = 0 \text{ on } y = 0 \text{ and } y = 1$$

We solve the problem using the Step Laplace transform method, using the augmented thin plate spline for the interpolating function in the dual reciprocity. We can see the solution in Figure 8.9 for time values between 0 and 2. The discontinuity is very clear to see between $t = 0.9$ and $t = 1$.

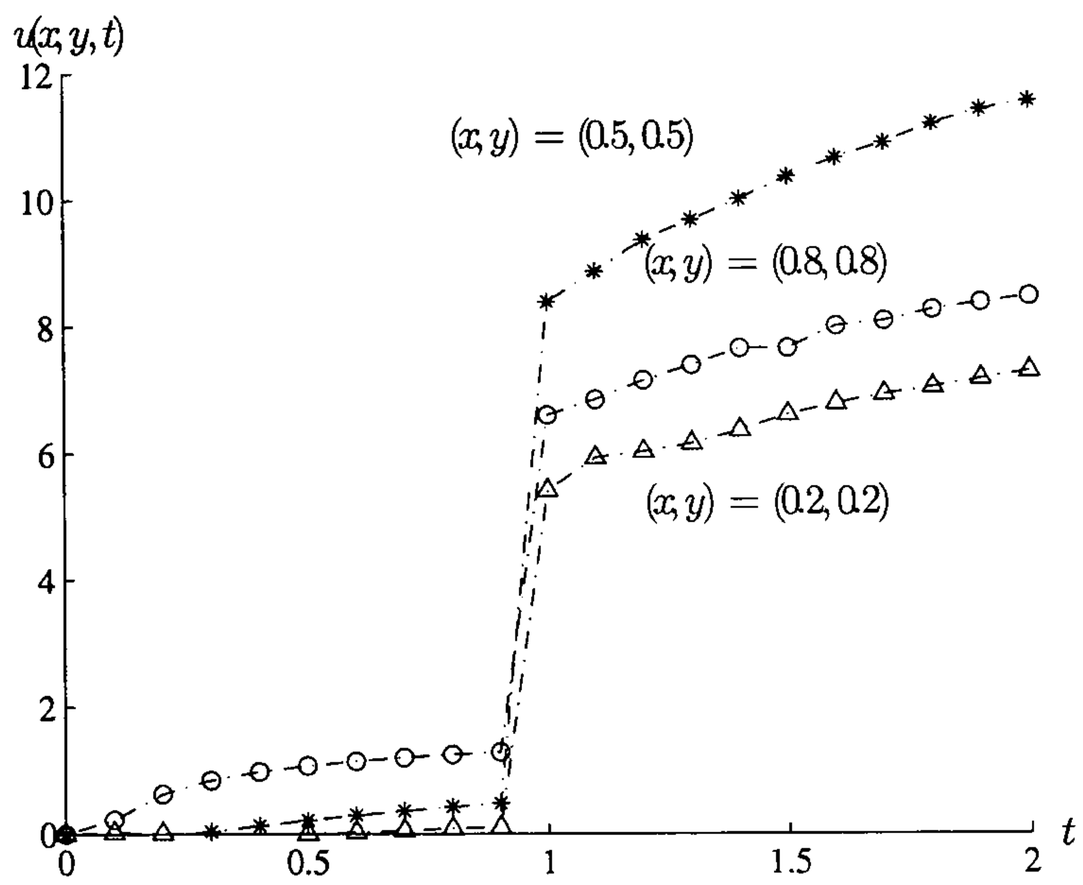

Figure 8.9: Time development of the Step LT solution for Example 8.3

In Figure 8.10 we see the solution of the x-values along the line $y = 0.5$ for $t = 0.1, 0.9, 1.1$ and 2. We can see the sudden jump after $t = 0.9$ and the curve is near to its maximum value of 13.3 for $x = 0.5$, the steady-state value, by $t = 2$.

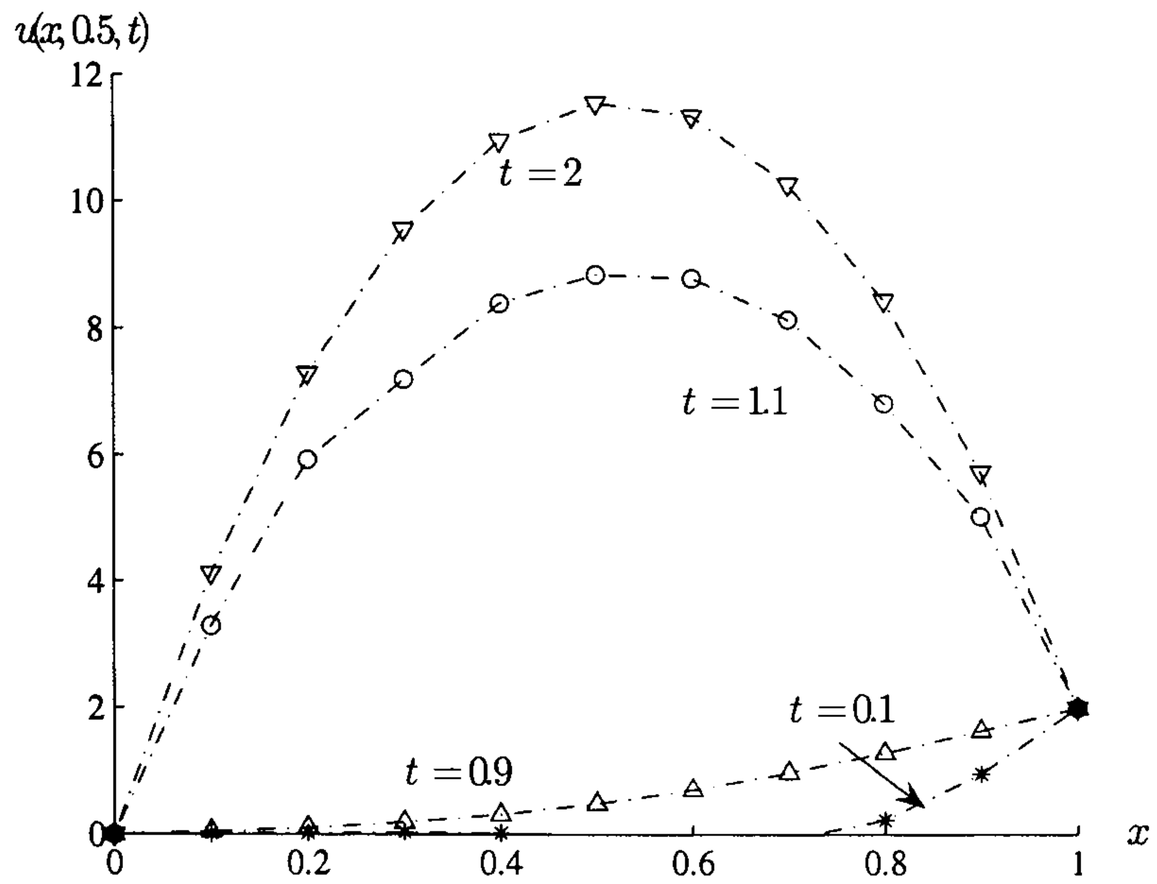

Figure 8.10: Space development of the Step LT solution for Example 8.3 for the internal nodes along the line $y = 0.5$

8.3 Problems with periodic boundary conditions

Parabolic problems in the time variable cannot produce time periodic solutions unless the data is itself periodic in time. The solution in such cases has the same period as the data. Consequently we know in advance the regions in which the solution is monotonic and we can apply the Laplace transform in a piecewise manner.

Suppose we have time-dependent boundary conditions, equations (8.2) and (8.3) which are periodic (Crann and Davies 2004b)

$$u_1(x, y, t + T) = u_1(x, y, t)$$

and

$$q_2(x, y, t + T) = q_2(x, y, t)$$

We apply the Laplace transform piecewise in time and seek solutions, $u^{(i)}(x, y)$, in the intervals $t_i \leq t \leq t_i + \frac{1}{4}T$, $i = 0, 1, 2, \ldots$ with $t_0 = 0$ as follows:

We solve

$$\nabla^2 u^{(i)} = \frac{1}{\alpha}\frac{\partial u^{(i)}}{\partial t} + h(x, y, t) \text{ in } D, \quad t_i \leq t \leq \tfrac{1}{4}T$$

subject to the boundary conditions

$$u^{(i)} = u_1(x, y, t) \text{ on } C_1$$

$$q^{(i)} = q_2(x, y, t) \text{ on } C_2$$

and the initial condition

$$u^{(i)}(x, y, 0) = u^{(i-1)}(x, y, t_{i-1} + \tfrac{1}{4}T)$$

We effect the Laplace transform solution by making the change of variable $t \to t_i + t'$ and so the problem is now defined on $0 \leq t' \leq \frac{1}{4}T$.

The following three examples are again defined in the unit square $\{(x, y) : 0 < x < 1, 0 < y < 1\}$ using $N = 32$ boundary points and $L = 9$ internal points. Also, as in the previous section, we use augmented thin plate splines in the dual reciprocity method. For the numerical Laplace transform we again use the Stehfest parameter value $M = 8$.

Example 8.4

This Dirichlet problem is defined as follows, see Figure 8.11,

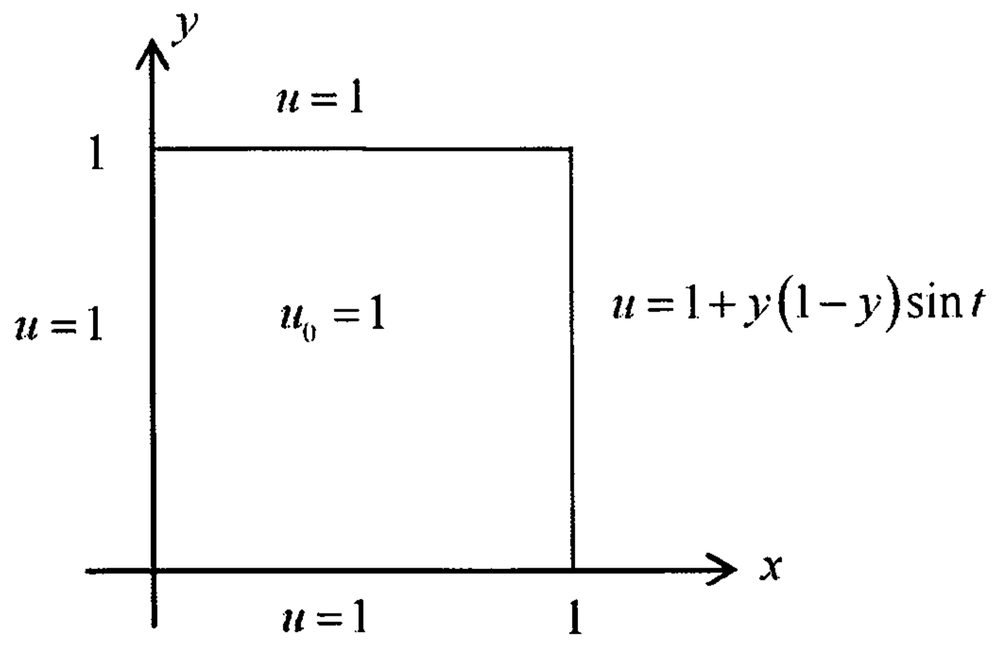

Figure 8.11: Boundary and initial conditions for Example 8.4

$$\nabla^2 u^{(i)} = \frac{1}{\alpha}\frac{\partial u^{(i)}}{\partial t} + h(x, y, t) \text{ in } D, \quad t_i \leq t \leq \tfrac{1}{4}T$$

$\alpha = 1.0$ and with the non-homogeneous term given by

$$h(x, y, t) = -2x \sin t - xy(1 - y) \cos t$$

subject to the boundary conditions

$$u(0, y, t) = u(x, 0, t) = u(x, 1, t) = 1$$

$$u(1, y, t) = 1 + y(1 - y) \sin t$$

and the initial condition

$$u(x, y, 0) = 1$$

We see that the boundary conditions have period 2π.

The analytic solution is

$$u(x, y, 0) = 1 + xy(1 - y) \sin t$$

In Figure 8.12 we show the solution over the period $0 \leq t \leq \frac{5}{2}\pi$.

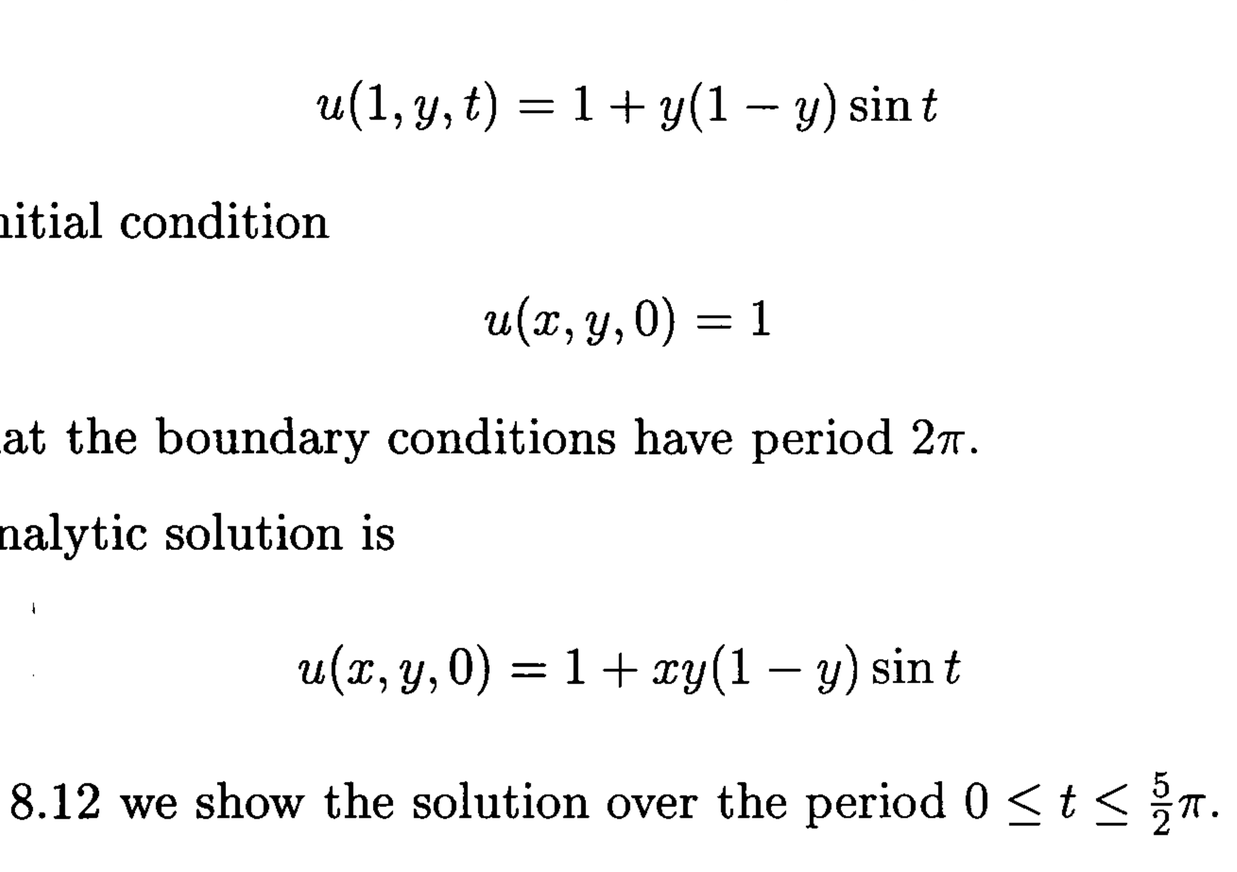

Figure 8.12: Time development at (0.25,0.25) using the Full LT solution for Example 8.4

We notice that, in the first quarter period, the approximation tracks the analytic solution very well but very quickly fails to pick up the oscillatory effect of the solution, as suggested by Stehfest.

In Figure 8.13 we show the Step LT time development of the approximate solution and the analytic solution at the point $(0.25, 0.25)$ plotted over the interval $0 \leq t \leq \frac{5}{2}\pi$ with time steps as described in Section 8.3.

We notice that the approximate solution tracks the analytic solution very well. The largest errors are found for values of t close to $t = \frac{1}{2}\pi$ and $t = \frac{3}{2}\pi$ and these errors are less that one percent. We also notice that the approximate solution is clearly exhibiting the correct periodic behaviour, tracking the analytic solution very well in the second period. Clearly, we can now predict approximate future values using the periodicity relationship $U_r(t) = U_r(t - 2n\pi)$ when $2n\pi \leq t \leq (2n+1)\pi$.

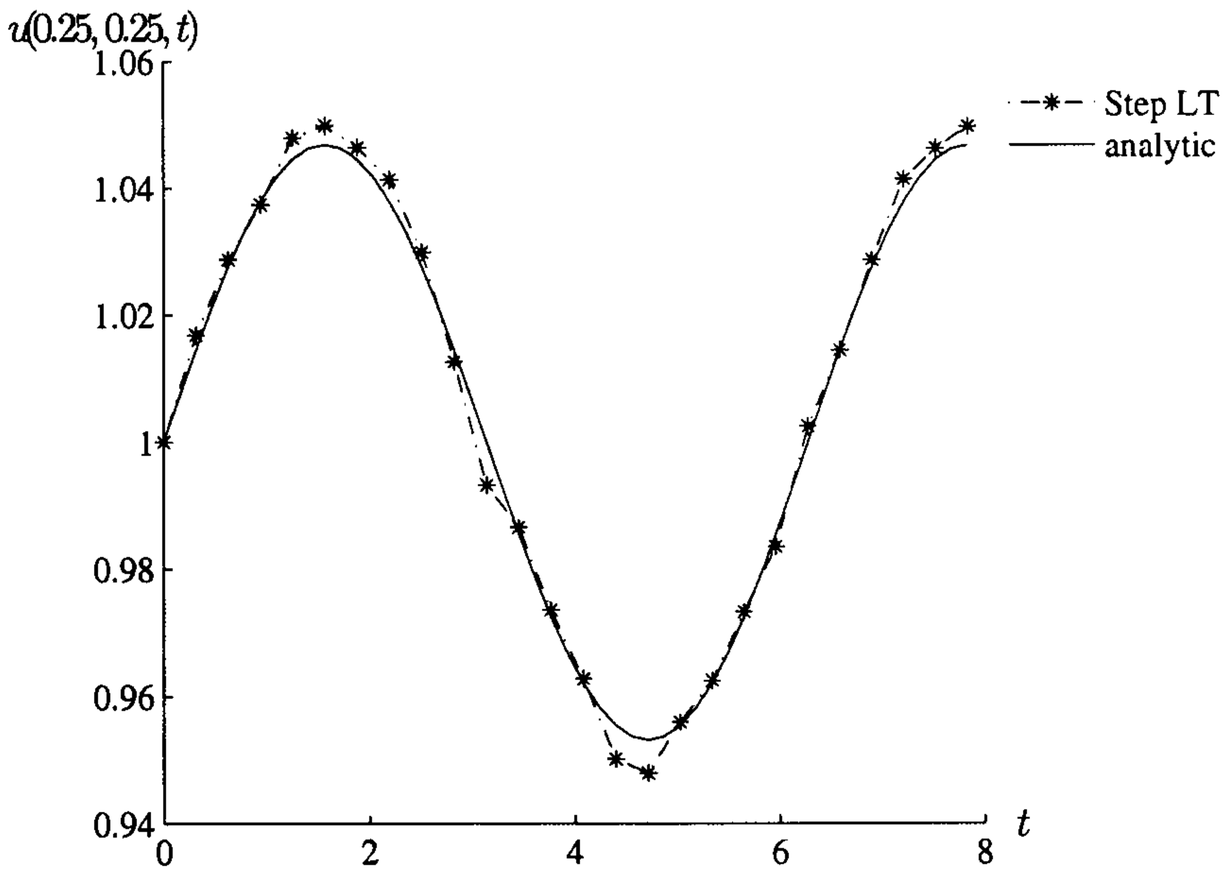

Figure 8.13: Time development at (0.25,0.25) using the Step LT solution for Example 8.4

In the next example we consider a similar problem but with periodic conditions on two boundaries.

Example 8.5

The problem is defined as follows, see Figure 8.14:

$$\nabla^2 u^{(i)} = \frac{1}{\alpha}\frac{\partial u^{(i)}}{\partial t} + h(x, y, t) \text{ in } D, \quad t_i \leq t \leq \tfrac{1}{4}T$$

$\alpha = 1.0$ and with the non-homogeneous term given by

$$h(x, y, t) = -2y \cos \pi t - 2x \sin \pi t$$

subject to the boundary conditions

$$u(0, y, t) = u(x, 0, t) = 0$$

$$u(1, y, t) = y(1 - y) \sin \pi t$$

$$u(x, 1, t) = x(1 - x) \cos \pi t$$

and the initial condition

$$u(x, y, 0) = xy(1 - x)$$

We see that this time the boundary conditions have period 2.

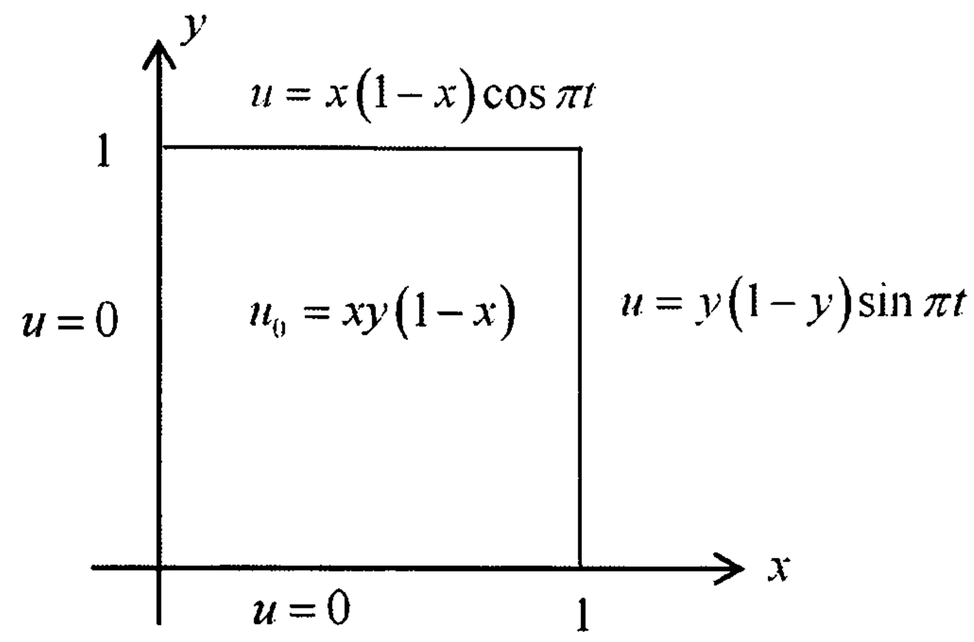

Figure 8.14: Boundary and initial conditions for Example 8.5

The analytic solution is

$$u(x, y, 0) = xy(1 - y) \sin \pi t + xy(1 - x) \cos \pi t$$

In Figure 8.15 we show the time development of the approximate solution and the analytic solution at the point $(0.25, 0.25)$ over the interval $0 \leq t \leq 4$.

Once again the approximate solution tracks the analytic solution very well and we can predict approximate future values for t in periods $[2n, 2(n + 1)]$.

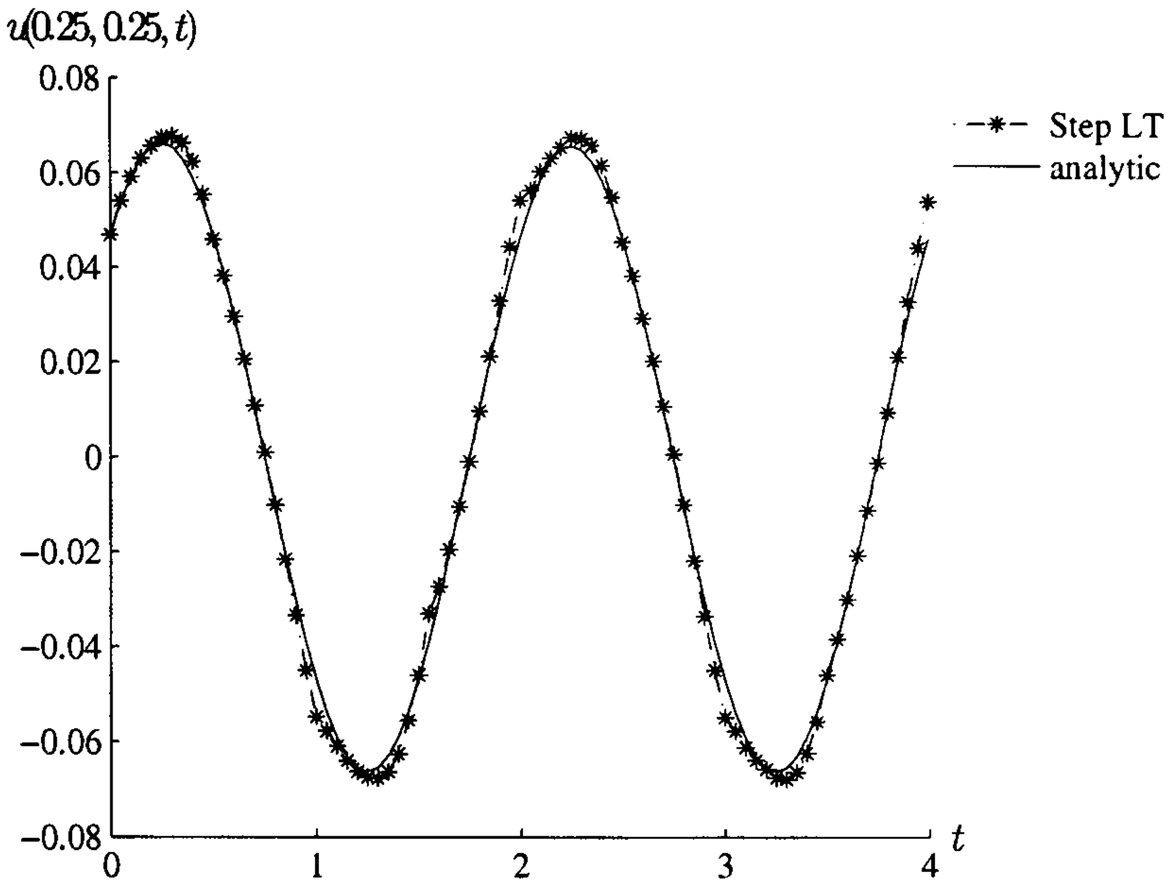

Figure 8.15: Time development of the solution at (0.25,0.25) for Example 8.5

Example 8.6

The previous two examples do not exhibit a transient term, the initial and boundary conditions are such that the systems are configured in the steady state at time $t = 0$. In the following example we consider a problem whose solution exhibits a transient term.

In this problem $\alpha = 0.2$ and the non-homogeneous term is given by

$$h(x, y, t) = -\frac{\pi}{\alpha} x \cos(\pi t)$$

The boundary conditions are given by, see Figure 8.16,

$$u(0, y, t) = 0$$

$$q(x, 0, t) = q(x, 1, t) = 0$$

$$u(1, y, t) = \sin \pi t$$

and the initial condition is

$$u(x, y, 0) = \sin \pi x$$

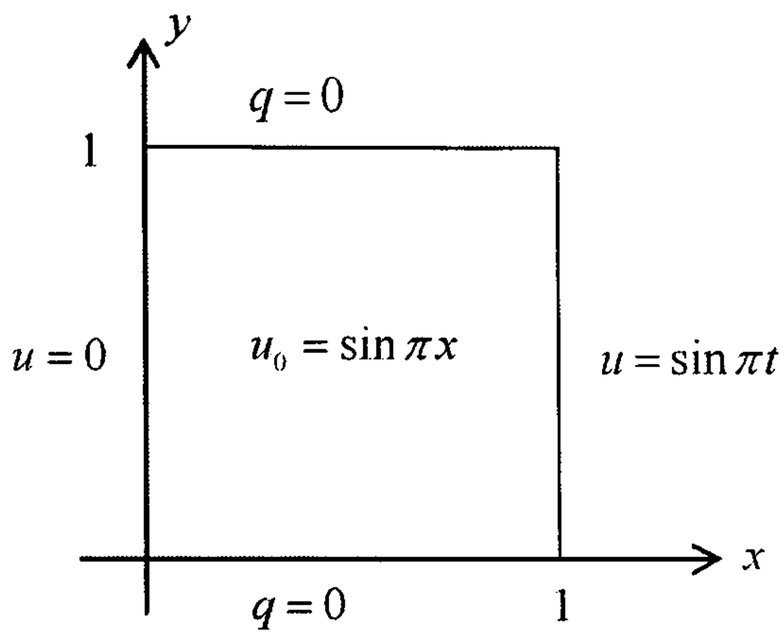

Figure 8.16: Boundary and initial conditions for Example 8.6

We see that the boundary conditions have period 2.

The analytic solution is given by

$$u(x, y, t) = \exp(-\alpha\pi^2 t) \sin \pi x + x \sin \pi t$$

In Figure 8.17 we show the time development of the approximate solution and the analytic solution at the point $(0.25, 0.25)$ plotted over $3\frac{1}{2}$ periods, *i.e.* over the interval $0 \leq t \leq 7$.

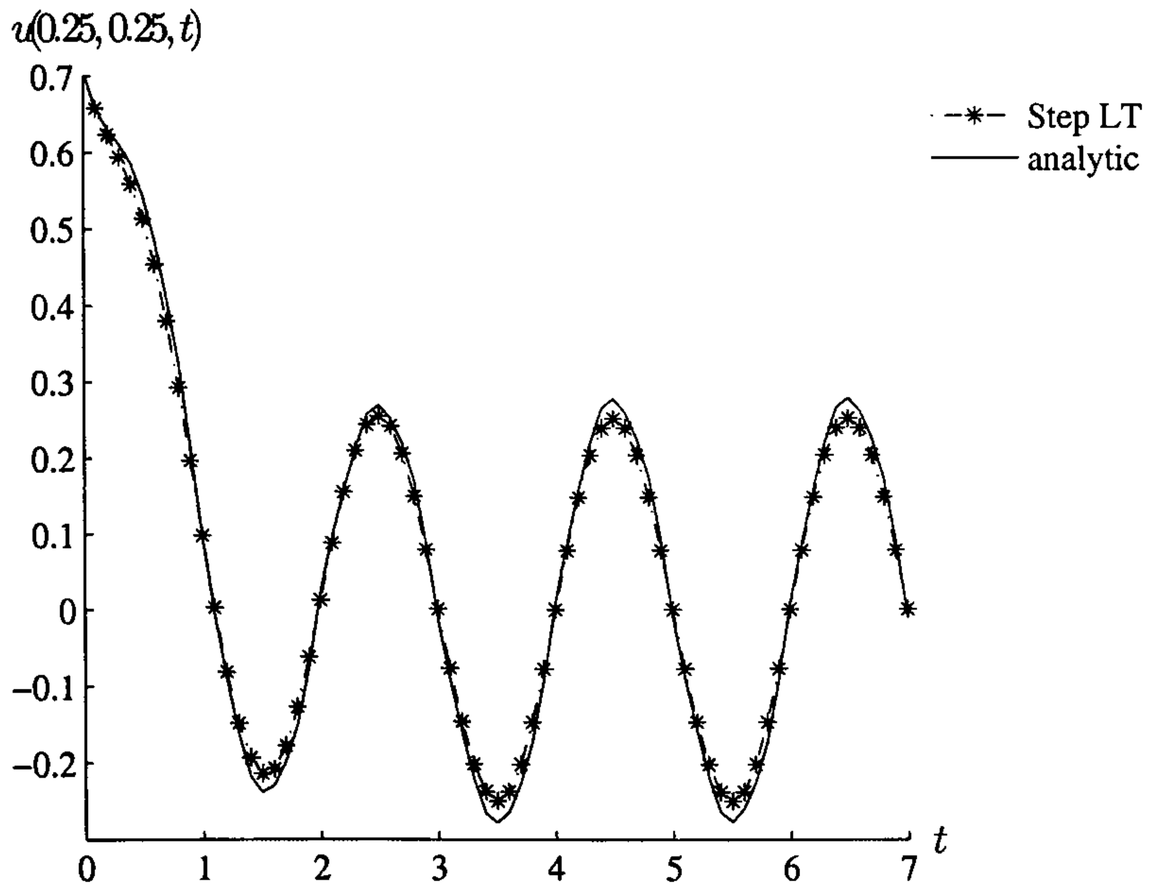

Figure 8.17: Time development of the solution at (0.25,0.25) for Example 8.6

We notice that the solution tracks the transient part very well and is in

good general agreement with the steady-state term. The numerical solution, see Table 8.1, suggests that the transient term has disappeared by $t = 3$. In fact, in the analytic solution, the transient term has a magnitude of the order of 0.002 at $t = 3$, *i.e.* smaller than the amplitude of the steady-state term, by a factor of about 100. The largest errors are at the points corresponding to maximum values of $|u|$ and these predict the steady-state amplitude to have an error of the order of approximately ten percent.

Table 8.1: Numerical solution of Example 8.6 for the internal node $(0.25, 0.25)$

time	steady-state term	transient term	analytic solution	approximate solution
0.5	0.250000	0.263544	0.513544	0.540789
1.0	0.000000	0.098225	0.098225	0.092984
1.5	-0.250000	0.036609	-0.213391	-0.236448
2.0	0.000000	0.013645	0.013645	0.025528
2.5	0.250000	0.005085	0.255085	0.282563
3.0	0.000000	0.001895	0.001895	-0.007991
3.5	-0.250000	0.000706	-0.249294	-0.275734
4.0	0.000000	0.000263	0.000263	0.011239
4.5	0.250000	0.000098	0.250098	0.276545
5.0	0.000000	0.000037	0.000037	-0.009184
5.5	-0.250000	0.000014	-0.249986	-0.276623
6.0	0.000000	0.000005	0.000005	0.010428
6.5	0.250000	0.000002	0.250002	0.276037
7.0	0.000000	0.000001	0.000001	-0.009111

8.4 Summary of Chapter 8

In this chapter we have shown that the Laplace transform boundary element method offers an excellent approach to the solution process for diffusion-type problems with discontinuous or periodic boundary conditions.

In the former case the Laplace transform cannot be applied directly since the approach smooths the condition in the neighbourhood of the discontinuity. We can overcome this problem by applying the Laplace transform in a

piecewise manner, developing the solution up to and including the discontinuity then using this solution as the initial value for a Laplace solution after the discontinuity.

If the solution is oscillatory in time we can apply the process in a piecewise manner in regions of width $\frac{1}{4}T$, where the period is T. An interesting observation is that we must use the process in a piecewise manner of intervals of width one-quarter period. We might expect that we should need only consider intervals of width one-half period. However our numerical experiments show that this is not the case and future work will be undertaken to explain this phenomenon.

Chapter 9

The solution of non-linear initial boundary-value problems

9.1 Introduction

In Chapter 6 we introduced the Laplace transform boundary element method for the solution of parabolic problems, showing that they can be solved easily and accurately by a variety of methods when using the Laplace transform for the time variable. In Chapter 7 we showed that this method when combined with the dual reciprocity method is an excellent method for the solution of time-dependent linear Poisson-type problems. In this section we consider non-linear Poisson-type problems.

The Laplace transform is a linear operator so we are not able to apply it directly to a non-linear equation. We develop an iterative process in which the equation is linearised in such a way that the Laplace transform can be used at each iterative step. The iteration proceeds until the change in the solution is within some predetermined tolerance. We illustrate the process

in the examples that follow.

The non-linear problems we shall consider in this chapter are of the form

$$\nabla^2 u = \frac{1}{\alpha(u)}\frac{\partial u}{\partial t} + a(u) + b(u)\frac{\partial u}{\partial x} + c(u)\frac{\partial u}{\partial y} + h(x, y, t)$$

9.2 Non-linear Poisson-type problems

Example 9.1

We consider the following problem defined in the unit square $\{(x, y) : 0 < x < 1,\, 0 < y < 1\}$, see Figure 9.1:

$$\nabla^2 u = \frac{1}{\alpha}\frac{\partial u}{\partial t} + u^2 + h \tag{9.1}$$

with

$$h = (2 + x^2)e^{-t} - x^4 e^{-2t}$$

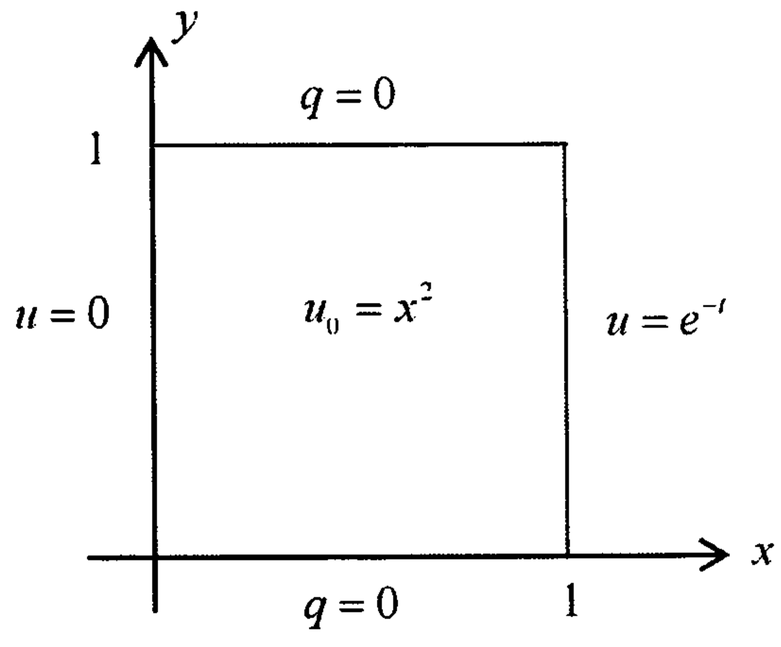

Figure 9.1: Boundary and initial conditions for Example 9.1

The boundary conditions are defined as follows:

$$u(1, y, t) = e^{-t}$$

$$q(x, 0, t) = q(x, 1, t) = 0$$

$$u(0, y, t) = 0$$

with initial condition

$$u(x, y, 0) = u_0 = x^2$$

The analytic solution is

$$u = x^2 e^{-t}$$

We attempt to transform the problem to Laplace space as before:

$$\nabla^2 \bar{u} = \frac{1}{\alpha}(\lambda \bar{u} - u_0) + \overline{(u^2)} + \frac{2 + x^2}{1 + \lambda} - \frac{x^4}{2 + \lambda}$$

However, the non-linear term cannot be transformed as it stands. We need to linearise the u^2 term and we do this with each of the three following methods:

1. **Direct iteration** We use direct iteration by putting the previous numerical results for u into the next iteration, so that equation (9.1) becomes

   $$\nabla^2 u_m = \frac{1}{\alpha}\frac{\partial u_m}{\partial t} + u_{m-1}^2 + (2 + x^2)e^{-t} - x^4 e^{-2t} \quad m = 1, 2, \ldots$$

   so that in Laplace space the equation is transformed to

   $$\nabla^2 \bar{u}_m = \frac{1}{\alpha}(\lambda \bar{u}_m - u_0) + \frac{u_{m-1}^2}{\lambda} + \frac{(2 + x^2)}{1 + \lambda} - \frac{x^4}{2 + \lambda}$$

   We start the process with the first approximation equal to the initial condition then solve the equation in Laplace space until we reach the required convergence and invert as usual.

2. **Semi-direct iteration** We follow Zhu (1999) and use a semi-direct iteration method by linearising any u^n term to a $(u_{m-1})^{n-1} u_m$ so that equation (9.1) becomes

   $$\nabla^2 u_m = \frac{1}{\alpha}\frac{\partial u_m}{\partial t} + u_{m-1} u_m + (2 + x^2)e^{-t} - x^4 e^{-2t} \quad m = 1, 2, \ldots$$

   and in Laplace space this is transformed to

   $$\nabla^2 \bar{u}_m = \frac{1}{\alpha}(\lambda \bar{u}_m - u_0) + u_{m-1}\bar{u}_m + \frac{(2 + x^2)}{1 + \lambda} - \frac{x^4}{2 + \lambda}$$

   and we can solve the problem as before.

3. **Taylor expansion iteration** Zhu (1999) also suggests using a first order Taylor expansion in the form

$$f(u_m) = f(u_{m-1}) + f'(u_{m-1})(u_m - u_{m-1}) \quad m = 1, 2, \ldots$$

which gives the following linearisation for equation (9.1)

$$\begin{aligned}\nabla^2 u_m &= \frac{1}{\alpha}\frac{\partial u_m}{\partial t} + u_{m-1}^2 + 2u_{m-1}(u_m - u_{m-1}) + (2 + x^2)e^{-t} - x^4 e^{-2t} \\ &= \frac{1}{\alpha}\frac{\partial u_m}{\partial t} + 2u_{m-1}u_m - u_{m-1}^2 + (2 + x^2)e^{-t} - x^4 e^{-2t}\end{aligned}$$

and the transformation in Laplace space becomes

$$\nabla^2 \bar{u}_m = \frac{1}{\alpha}(\lambda \bar{u}_m - u_0) + 2u_{m-1}\bar{u}_m - \frac{u_{m-1}^2}{\lambda} + \frac{(2 + x^2)}{1 + \lambda} - \frac{x^4}{2 + \lambda}$$

In all three cases we stop the iteration when the predetermined tolerance, ϵ, is satisfied

$$\left| \frac{\max(\text{abs}(u_{m-1} - u_m))}{\max(\text{abs}(u_{m-1} + u_m))} \right| < \epsilon$$

We use 32 boundary and 9 internal nodes, $M = 8$ for the Stehfest inversion parameter and the augmented thin plate spline for the radial basis function in the dual reciprocity method. We choose $\epsilon = 0.001$ for the linearisation and consider times $t = 0.1, 0.2, \ldots, 2.0$. In Figure 9.2 we show the three approximations together with the analytic solution at the internal node $(0.2, 0.2)$ for the problem in Example 9.1.

We see that all three iteration methods are in good agreement with the analytic solution.

In Table 9.1 we show the percentage errors for the three methods from the analytic solution. We see that the Taylor iteration method gives the best results and where appropriate we would use this method. However the results for all three methods are sufficiently good for practical purposes, so we can use any method with confidence.

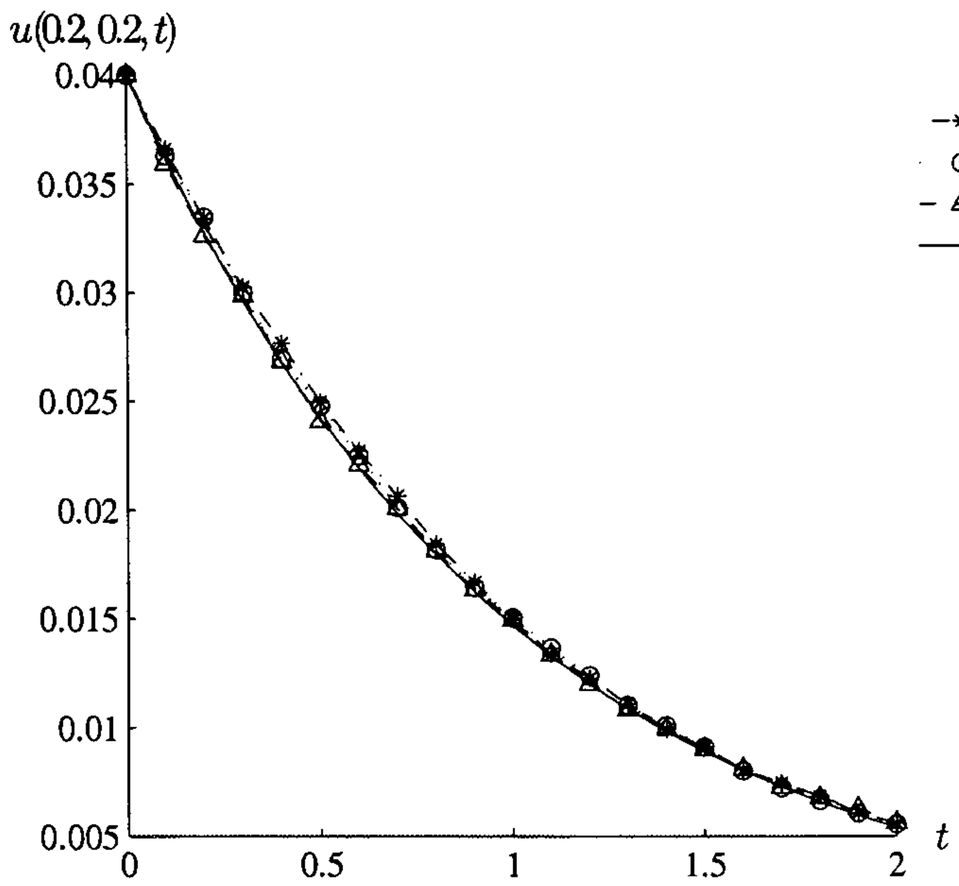

Figure 9.2: Time development of the solution for Example 9.1

Table 9.1: Percentage errors for the three methods for Example 9.1

Time	Direct	Semi-direct	Taylor
0.1	0.09	1.21	0.14
0.2	0.64	3.73	0.13
0.3	4.44	2.53	0.21
0.4	4.23	2.94	0.29
0.5	3.47	3.88	0.07
0.6	3.89	3.41	0.17
0.7	4.93	1.67	0.36
0.8	4.20	2.81	0.10
0.9	2.76	3.12	0.10
1.0	2.39	1.26	0.22
1.1	1.78	0.78	0.20
1.2	2.91	1.16	0.04
1.3	1.57	2.13	0.11
1.4	2.55	2.25	0.04
1.5	4.64	4.73	0.17
1.6	1.11	0.62	0.11
1.7	0.46	0.07	0.04
1.8	3.49	1.60	0.29
1.9	3.03	2.90	0.42
2.0	0.32	2.71	0.18

Example 9.2

This example has the same geometry, see Figure 9.1 and analytic solution $u = x^2 e^{-t}$ as Example 9.1 but with a $\partial u/\partial x$ term as follows:

$$\nabla^2 u = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + h \tag{9.2}$$

with

$$h = (2 + x^2)e^{-t} - 2x^3 e^{-2t}$$

We can do a direct linearisation on equation (9.2) in two different ways. We can linearise the u term as follows:

$$\nabla^2 u_m = \frac{\partial u_m}{\partial t} + u_{m-1}\frac{\partial u_m}{\partial x} + (2 + x^2)e^{-t} - 2x^3 e^{-2t}$$

so that in Laplace space we have

$$\nabla^2 \bar{u}_m = (\lambda \bar{u}_m - u_0) + u_{m-1}\frac{\partial \bar{u}_m}{\partial x} + \frac{(2 + x^2)}{1 + \lambda} - \frac{2x^3}{2 + \lambda}$$

and use the radial basis function $f = 1 + R$.

Alternatively we can linearise the $\partial u/\partial x$ term

$$\nabla^2 u_m = \frac{\partial u_m}{\partial t} + u_m\frac{\partial u_{m-1}}{\partial x} + (2 + x^2)e^{-t} - 2x^3 e^{-2t}$$

so that in Laplace space we have

$$\nabla^2 \bar{u}_m = (\lambda \bar{u}_m - u_0) + \bar{u}_m\frac{\partial u_{m-1}}{\partial x} + \frac{(2 + x^2)}{1 + \lambda} - \frac{2x^3}{2 + \lambda}$$

and use the augmented thin plate spline in the dual reciprocity approach.

We solve the problem as before and consider the solution at the three internal points $(0.2, 0.2)$, $(0.5, 0.5)$, $(0.8, 0.8)$, see Figure 9.3 for both iteration approaches. We see that both approaches show very good agreement to the analytic solution and approach the steady-state solution correctly.

We show in Tables 9.2, 9.3 and 9.4 the numerical solutions for the two approaches in Example 9.2. For both approaches the smaller time values show

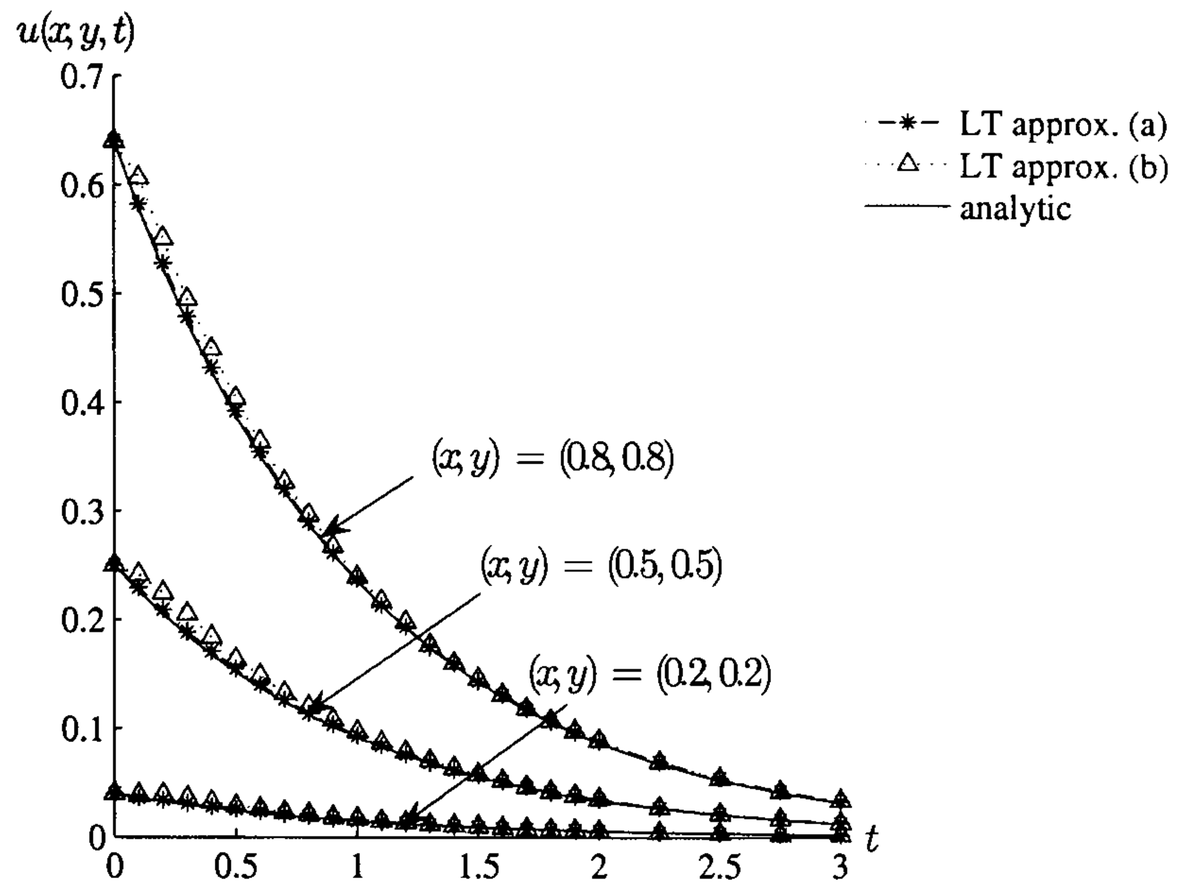

Figure 9.3: Time development of the solution for Example 9.2 (a) u linear, (b) $\partial u/\partial x$ linear

the largest errors and the first approach, linearising the u term has maximum error of eight percent. The second approach, linearising the $\partial u/\partial x$ term, is slightly less accurate even though it uses the augmented thin plate spline.

Table 9.2: Numerical solution and percentage errors for the two iterative approaches for Example 9.2 for the node $(0.2, 0.2)$

time	solution analytic	solution u linear	solution $\partial u/\partial x$ linear	errors u linear	errors $\partial u/\partial x$ linear
0.2	0.032749	0.034490	0.039071	5.32	19.30
0.4	0.026813	0.028335	0.032381	5.68	20.77
0.6	0.021952	0.023637	0.026007	7.67	18.47
0.8	0.017973	0.019317	0.020729	7.48	15.33
1.0	0.014715	0.015429	0.017098	4.85	16.19
1.2	0.012048	0.012742	0.013395	5.76	11.18
1.4	0.009864	0.010337	0.010805	4.80	9.54
1.6	0.008076	0.008301	0.008593	2.79	6.40
1.8	0.006612	0.006940	0.007009	4.96	6.00
2.0	0.005413	0.005683	0.005785	4.98	6.86

Table 9.3: Numerical solution and percentage errors for the two iterative approaches for Example 9.2 for the node $(0.5, 0.5)$

time	solution analytic	solution u linear	solution $\partial u/\partial x$ linear	errors u linear	errors $\partial u/\partial x$ linear
0.2	0.204683	0.209050	0.224503	2.13	8.83
0.4	0.167580	0.170873	0.184066	1.97	8.96
0.6	0.137203	0.139974	0.148334	2.02	7.50
0.8	0.112332	0.113768	0.119867	1.28	6.29
1.0	0.091970	0.093526	0.097139	1.69	5.32
1.2	0.075299	0.076161	0.078781	1.15	4.42
1.4	0.061649	0.062231	0.064045	0.94	3.74
1.6	0.050474	0.051419	0.052223	1.87	3.35
1.8	0.041325	0.041857	0.042345	1.29	2.41
2.0	0.033834	0.034765	0.035577	2.75	4.90

Example 9.3

We now consider a problem with a different non-linear term, e^{-u}. The geometry and analytic solution are as in Examples 9.1 and 9.2. The problem is stated as follows:

$$\nabla^2 u = \frac{\partial u}{\partial t} + e^{-u} + h \tag{9.3}$$

with

$$h = (2 + x^2)e^{-t} - \exp(-x^2 e^{-t})$$

Table 9.4: Numerical solution and percentage errors for the two iterative approaches for Example 9.2 for the node $(0.8, 0.8)$

time	solution analytic	solution u linear	solution $\partial u/\partial x$linear	errors u linear	errors $\partial u/\partial x$linear
0.2	0.523988	0.528426	0.551200	0.85	4.94
0.4	0.429005	0.432554	0.449832	0.83	4.63
0.6	0.351239	0.355285	0.364780	1.15	3.71
0.8	0.287571	0.290422	0.296735	0.99	3.09
1.0	0.235443	0.237568	0.239403	0.90	1.65
1.2	0.192764	0.193683	0.198082	0.48	2.68
1.4	0.157822	0.159482	0.160290	1.05	1.54
1.6	0.129214	0.131251	0.131059	1.58	1.41
1.8	0.105791	0.106295	0.107511	0.48	1.60
2.0	0.086615	0.088230	0.089285	1.87	2.99

We use direct linearisation to give

$$\nabla^2 u_m = \frac{\partial u_m}{\partial t} + e^{-u_{m-1}} + (2 + x^2)e^{-t} - \exp(-x^2 e^{-t}) \tag{9.4}$$

However, when we take the Laplace transform we have a problem with the $\exp(-x^2 e^{-t})$ term so we use our experience from Section 5.3.3 and develop the Maclaurin series expansion for the exponential as follows:

$$\exp(-x^2 e^{-t}) = 1 - x^2 e^{-t} + \frac{x^4 e^{-2t}}{2!} - \frac{x^6 e^{-3t}}{3!} + \frac{x^8 e^{-4t}}{4!} - \dots$$

and we are able to take the Laplace transform of equation (9.4), stopping the exponential expansion after the fifth term.

$$\nabla^2 \bar{u}_m = \frac{\partial \bar{u}_m}{\partial t} + \frac{e^{-u_{m-1}}}{\lambda} + \frac{(2 + x^2)}{1 + \lambda} -$$

$$- \left(\frac{1}{\lambda} - \frac{x^2}{1 + \lambda} + \frac{x^4}{2!(2 + \lambda)} - \frac{x^6}{3!(3 + \lambda)} + \frac{x^8}{4!(4 + \lambda)} \right)$$

We solve the problem with $f = 1 + R$ for the radial basis function in the dual reciprocity formulation, a tolerance of 0.001 in the iteration process and compare our approximation with the analytic solution for $t = 0.1 \dots, 3.0$, see Figure 9.4.

Once again, the analytic and approximate solutions in the graph are indistinguishable and we look at the numerical results in Table 9.5 and we

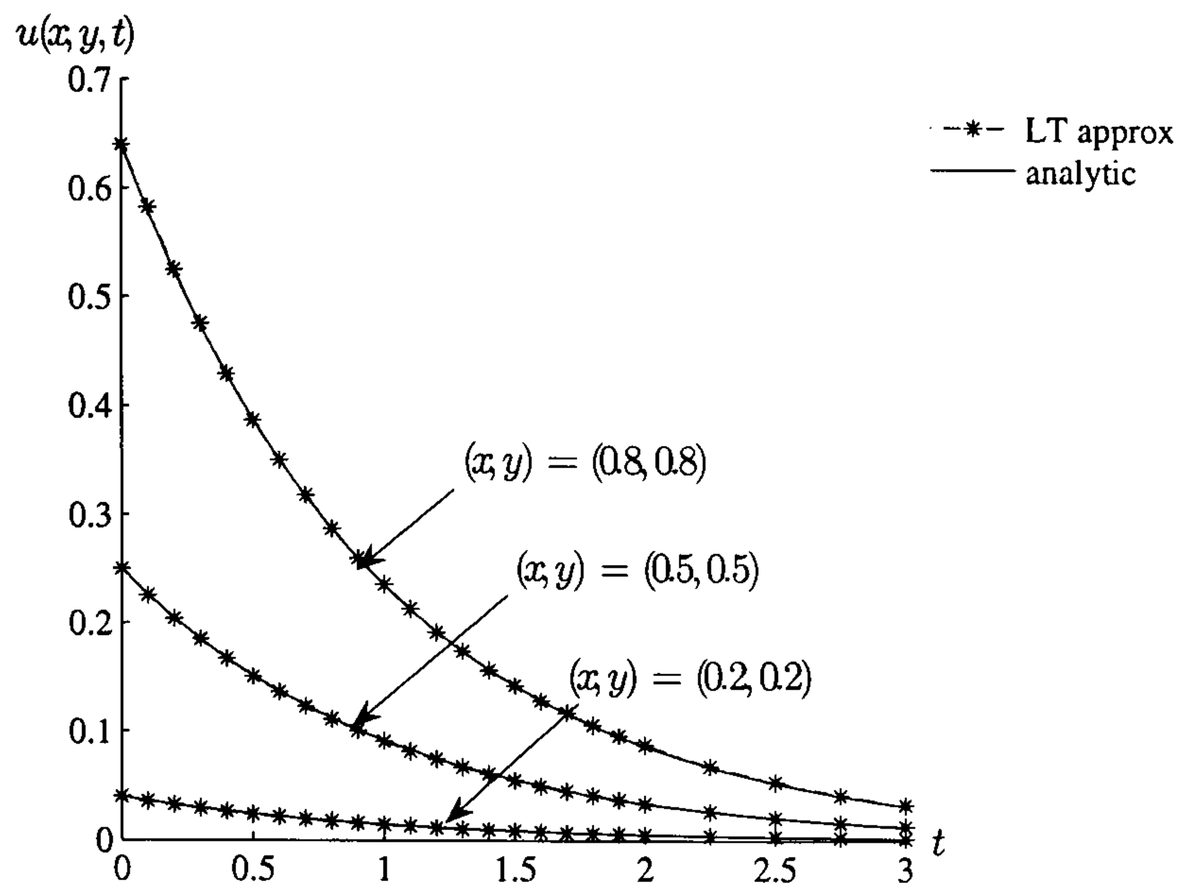

Figure 9.4: Time development of the solution for Example 9.3

see that the approximate solutions compare with the analytic values very well as before.

Table 9.5: Numerical solution for Example 9.3

Time	(0.2,0.2) analytic	(0.2,0.2) approx	(0.5,0.5) analytic	(0.5,0.5) approx	(0.8,0.8) analytic	(0.8,0.8) approx
0.20	0.0327	0.0326	0.2047	0.2041	0.5240	0.5252
0.40	0.0268	0.0263	0.1676	0.1673	0.4290	0.4299
0.60	0.0220	0.0214	0.1372	0.1365	0.3512	0.3511
0.80	0.0180	0.0176	0.1123	0.1115	0.2876	0.2878
1.00	0.0147	0.0143	0.0920	0.0913	0.2354	0.2363
1.20	0.0120	0.0115	0.0753	0.0748	0.1928	0.1921
1.40	0.0099	0.0094	0.0616	0.0610	0.1578	0.1571
1.60	0.0081	0.0078	0.0505	0.0498	0.1292	0.1286
1.80	0.0066	0.0063	0.0413	0.0416	0.1058	0.1065
2.00	0.0054	0.0053	0.0338	0.0339	0.0866	0.0881

However, of more interest, perhaps, are the percentage errors for the three internal nodes as shown in Table 9.6 together with the number of iterations needed for the iterative process of linearisation.

We see that the maximum percentage errors are five percent for the

Table 9.6: Percentage errors for Example 9.3 with number of iterations

time	(0.2,0.2)	(0.5,0.5)	(0.8,0.8)	iterations
0.2	0.54	0.29	0.24	5
0.4	2.06	0.17	0.21	4
0.6	2.50	0.48	0.04	4
0.8	2.18	0.75	0.08	3
1.0	2.97	0.77	0.35	5
1.2	4.52	0.61	0.32	7
1.4	4.92	1.04	0.46	6
1.6	3.12	1.41	0.44	4
1.8	5.22	0.62	0.67	5
2.0	2.58	0.17	1.73	5

internal node (0.2,0.2) and mostly less than one percent for the other two nodes. The average number of iterations needed for the linearisation process is 5 iterations. These are very promising results; we are using the basic direct iteration method, a simple radial basis function of $f = 1 + R$, an additional approximation for the exponential term and we get good results. There is plenty of scope for the further investigation of more complicated problems using more accurate methods.

Example 9.4

Consider the transient heat problem defined by the partial differential equation given by

$$\nabla.\,(k(u)\nabla u) = \frac{\partial}{\partial t}(\rho c u) \tag{9.5}$$

Writing

$$\nabla.(k(u)\nabla u) = k(u)\nabla^2 u + \nabla k.\nabla u$$

and taking ρ and c to be 1, we have

$$\nabla^2 u = \frac{1}{k(u)}\left(\frac{\partial u}{\partial t} - \nabla k.\nabla u\right)$$

Now

$$\nabla k = \frac{dk}{du}\nabla u$$

therefore

$$\nabla^2 u = \frac{1}{k(u)}\left(\frac{\partial u}{\partial t} - k'(u)|\nabla u|^2\right) \tag{9.6}$$

We linearise equation (9.6)

$$\nabla^2 u_m = \frac{1}{k(u_{m-1})}\left(\frac{\partial u_m}{\partial t} - k'(u_{m-1})|\nabla u_{m-1}|^2\right) \tag{9.7}$$

so that we can take the Laplace transform to obtain

$$\nabla^2 \bar{u}_m = \frac{1}{k(u_{m-1})}\left(\lambda \bar{u}_m - u_0 - \frac{k'(u_{m-1})|\nabla u_{m-1}|^2}{\lambda}\right) \tag{9.8}$$

Chen and Lin (1991) describe a transient heat conduction problem in a one-dimensional slab with

$$\nabla.\,(k(u)\nabla u) = \frac{\partial u}{\partial t} \tag{9.9}$$

and

$$k(u) = 1 + \beta u$$

subject to boundary conditions

$$u = 1 \text{ on } x = 1$$

$$q = 0 \text{ on } x = 0, y = 0, y = 1$$

and initial condition

$$u_0 = 0$$

We consider the same problem posed in two dimensions for which the solution is independent of y, see Figure 9.5.

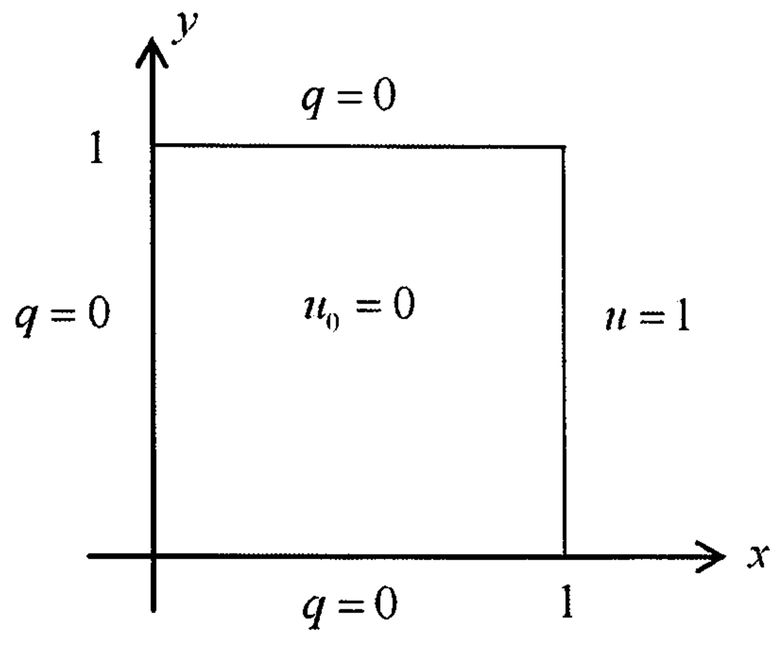

Figure 9.5: Boundary and initial conditions for Example 9.4

Chen and Lin choose $\beta = -0.3$ and use the Laplace transform with the FDM to solve the elliptic equation with eleven x-values, $x = 0, 0.1 \ldots, 1.0$ and invert back from Laplace space using a complex numerical inversion process. We use eleven nodes on each of the boundaries $y = 0$ and $y = 1$, $f = 1 + R$ for the dual reciprocity interpolating function, and find the solution at the nodes along $y = 0.5$.

Our solutions are shown in Figure 9.6 with those reported by Chen and Lin for $t = 0.2$ and $t = 1.0$. Our numerical values are shown in Tables 9.7.

We see that our solutions are comparable with the approximations reported by Chen and Lin and we conclude that our process is a suitable approach for non-linear heat conduction problems.

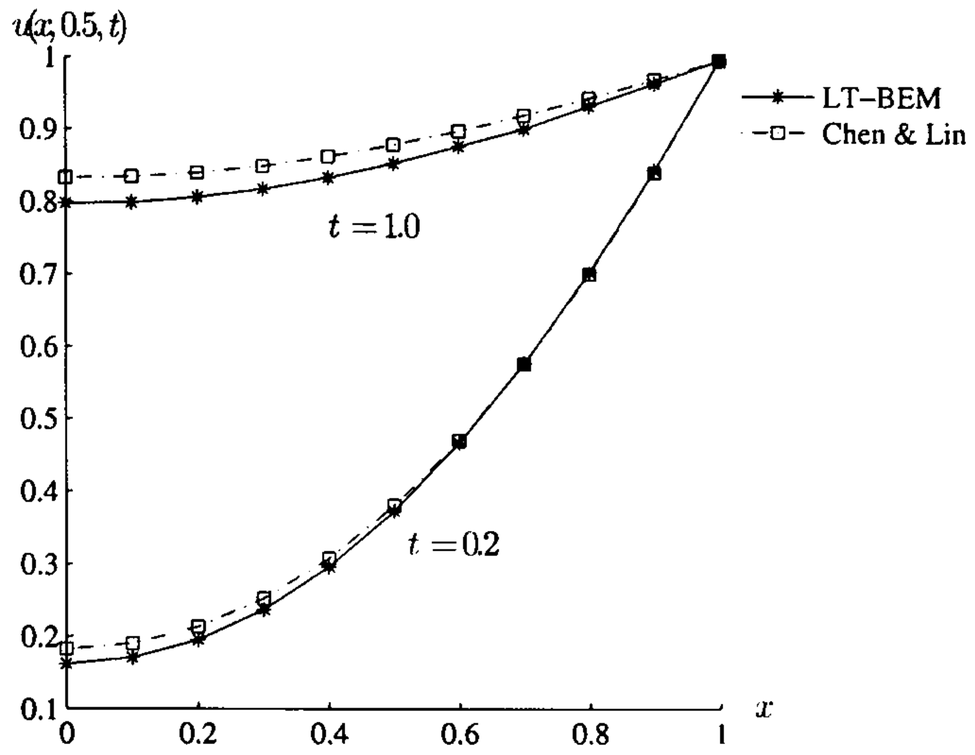

Figure 9.6: Space solution for Example 9.4 at $t = 0.2$ and $t = 1.0$

Table 9.7: Numerical solution for Example 9.4 at $t = 0.2$ and $t = 1.0$

	LTBEM approx	Chen & Lin approx	LTBEM approx	Chen & Lin approx
x	$t = 0.2$	$t = 0.2$	$t = 1.0$	$t = 1.0$
0.0	0.1618	0.1823	0.7978	0.8329
0.1	0.1706	0.1901	0.7994	0.8349
0.2	0.1954	0.2134	0.8070	0.8408
0.3	0.2370	0.2527	0.8191	0.8507
0.4	0.2966	0.3084	0.8350	0.8644
0.5	0.3736	0.3810	0.8556	0.8809
0.6	0.4678	0.4710	0.8796	0.9007
0.7	0.5787	0.5784	0.9044	0.9231
0.8	0.7061	0.7032	0.9363	0.9474
0.9	0.8473	0.8444	0.9676	0.9733
1.0	1.0000	1.0000	1.0000	1.0000

9.3 A coupled non-linear problem

There are many situations in applied science and engineering where materials are heated electrically via the ohmic heating, or Joule heating, process. In this process the heating occurs throughout the volume as compared with surface heating in conventional processes. The technique is frequently used as a method of food sterilisation in the food processing industry. It is important to know both that the food material itself is not degraded and that

the temperatures reached are sufficient to kill bacteria. These problems exhibit significant non-linearities since, for food materials, the electrical and thermal properties are dependent on the temperature. When this happens the resulting model of the ohmic heating process comprises a pair of coupled non-linear partial differential equations.

Problems of heat generation with coupled non-linear partial differential equations have been solved using a finite difference approach by Please *et al.* (1995) and a finite element solution is described by de Alwis and Fryer (1990) and Elliot and Larsson (1995). We shall use the Laplace transform boundary element with dual reciprocity and linearisation as described in the previous section (Crann *et al.* 2005).

We shall consider problems in a two-dimensional region, D, bounded by the closed curve $C = C_1 + C_2$. The underlying equations are described by Please *et al.* (1995):

1. The reactive convection-diffusion equation describing heat flow in D

$$\nabla.(k\nabla u) = \frac{\partial}{\partial t}(\rho c u) + \mathbf{v}.\nabla(\rho c u) - \sigma|\nabla\phi|^2 \tag{9.10}$$

2. The generalised Laplace equation describing the electric potential in D

$$\nabla.(\sigma\nabla\phi) = 0 \tag{9.11}$$

where $k = k(u)$ and $\sigma = \sigma(u)$, together with suitable boundary conditions on C

$$u = u_1(x, y, t) \text{ and } \phi = \phi_1(x, y, t) \text{ on } C_1 \tag{9.12}$$

$$q \equiv \frac{\partial u}{\partial n} = q_2(x, y, t) \text{ and } \psi \equiv \frac{\partial \phi}{\partial n} = \psi_2(x, y, t) \text{ on } C_2 \tag{9.13}$$

and initial conditions

$$u(x, y, 0) = u_0(x, y) \text{ and } \phi(x, y, 0) = \phi_0(x, y) \text{ in } D \tag{9.14}$$

At any point (x, y) and time t, the dependent variables are the temperature u and the electric potential ϕ. Once again the material parameters are the thermal conductivity k, electrical conductivity σ, the density ρ, the specific heat c and the velocity of convection $\mathbf{v}$.

We shall assume that ρ and c are constant and that k and σ depend on x, y and u. We re-write equations (9.21) and (9.22):

$$\nabla^2 u = \frac{1}{k}\left(-\nabla k.\nabla u + \rho c\mathbf{v}.\nabla u - \sigma|\nabla\phi|^2 + \rho c\frac{\partial u}{\partial t}\right) \tag{9.15}$$

$$\nabla^2\phi = \frac{1}{\sigma}\left(-\nabla\sigma.\nabla\phi\right) \tag{9.16}$$

which allows us to use the fundamental solution, $-\frac{1}{2\pi}\ln R$, for the Laplacian operator.

Before we can use the Laplace transform we must linearise equations (9.15) and (9.16) for an iterative approach. Since the examples in the previous section show that there is little to choose between the methods, we use the most simple method, the so-called direct iteration method. In order to simplify notation we use the symbols $\tilde{u}$ and $\tilde{\phi}$ to denote values from the previous iteration and re-write the equations as

$$\nabla^2 u = \frac{1}{k(\tilde{u})}\left(-\nabla k(\tilde{u}).\nabla u + \rho c\mathbf{v}.\nabla u - \sigma(\tilde{u})|\nabla\tilde{\phi}|^2 + \rho c\frac{\partial u}{\partial t}\right) \tag{9.17}$$

$$\nabla^2\phi = \frac{1}{\sigma(\tilde{u})}\left(-\nabla\sigma(\tilde{u}).\nabla\phi\right) \tag{9.18}$$

In Laplace space the initial boundary-value problem defined by equations (9.17), (9.18), (9.12), (9.13) and (9.14) becomes

$$\nabla^2\bar{u} = \frac{1}{k(\tilde{u})}\left(-\nabla k(\tilde{u}).\nabla\bar{u} + \rho c\mathbf{v}.\nabla\bar{u} - \frac{1}{\lambda}\sigma(\tilde{u})|\nabla\tilde{\phi}|^2 + \rho c(\lambda\bar{u} - u_0)\right) \tag{9.19}$$

$$\nabla^2\bar{\phi} = \frac{1}{\sigma(\tilde{u})}\left(-\nabla\sigma(\tilde{u}).\nabla\bar{\phi}\right) \tag{9.20}$$

Example 9.5

In problems in the food processing industry a good model for the thermophysical properties is that the heat capacity, ρc, is constant and both conductivities are linear with temperature.

Consequently we shall consider the following model problem (Crann *et al.* 2005), where we choose the functions $h_1(x, y, t)$ and $h_2(x, y, t)$ so that we have known analytic solutions $u = (x - \frac{1}{2}x^2)(2 - e^{-t})$ and $\phi = x + (x - x^2)e^{-t}$.

We seek the solution to the initial boundary-value problem

$$\nabla.(k\nabla u) = \frac{\partial}{\partial t}(\rho c u) + \mathbf{v}.\nabla(\rho c u) - \sigma|\nabla\phi|^2 \tag{9.21}$$

$$\nabla.(\sigma\nabla\phi) = 0 \tag{9.22}$$

$$\text{with } \rho c = 1, \ \mathbf{v} = \mathbf{i}, \ k(u) = 1 + u, \ \sigma(u) = 1 + u,$$

in the region $\{(x, y) : 0 < x < 1, \ 0 < y < 1\}$ subject to the boundary conditions, see Figure 9.7,

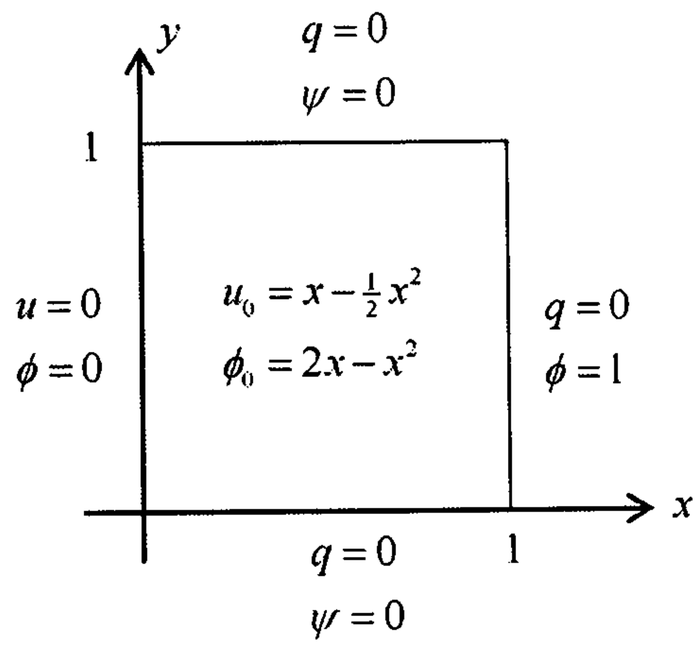

Figure 9.7: Boundary and initial conditions for Example 9.5

$$u = 0 \text{ on } x = 0, \ \ q = 0 \text{ on } x = 1, \ \ y = 0, \ \ y = 1,$$

$$\phi = 0 \text{ on } x = 0, \ \ \phi = 1 \text{ on } x = 1, \ \ \psi \equiv \frac{\partial\phi}{\partial n} = 0 \text{ on } y = 0, \ \ y = 1,$$

and the initial conditions

$$u(x,y,0) = x - \frac{1}{2}x^2 \text{ and } \phi(x,y,0) = 2x - x^2$$

$h_1(x,y,t)$ and $h_2(x,y,t)$ are given by

$$h_1(x,y,t) = (1 - 10x + 6x^2) + (6x - \frac{11}{2}x^2)e^{-t} + (2 - 7x + \frac{11}{2}x^2)e^{-2t}$$

$$h_2(x,y,t) = (2 - 2x) + (-1 - 9 + 6x^2)e^{-t} + (-1 + 5x - 3x^2)e^{-2t}$$

In the dual reciprocity form for equations (9.21) and (9.22) we use $f = 1 + R$. Details can be found in Crann *et al.* (2005). For the numerical solution we choose 32 boundary points and 9 internal points and $M = 8$ for the Stehfest inversion parameter. We use a tolerance $\epsilon = 0.001$ for the direct linearisation iteration method.

The space distributions for time values $t = 0.1$, 0.5, 1 and 5 are shown in Figures 9.8 and 9.9 and the time developments for values $x = 0.2$, 0.5 and 0.8 are shown in Figures 9.10 and 9.11. We note that the solution is independent of y.

We see that the approximate solution compares very well with the analytic values, typical errors being of the order of about three percent for ϕ and about four percent for u. Typically we need approximately four iterations to achieve convergence within tolerance for both iterative cycles.

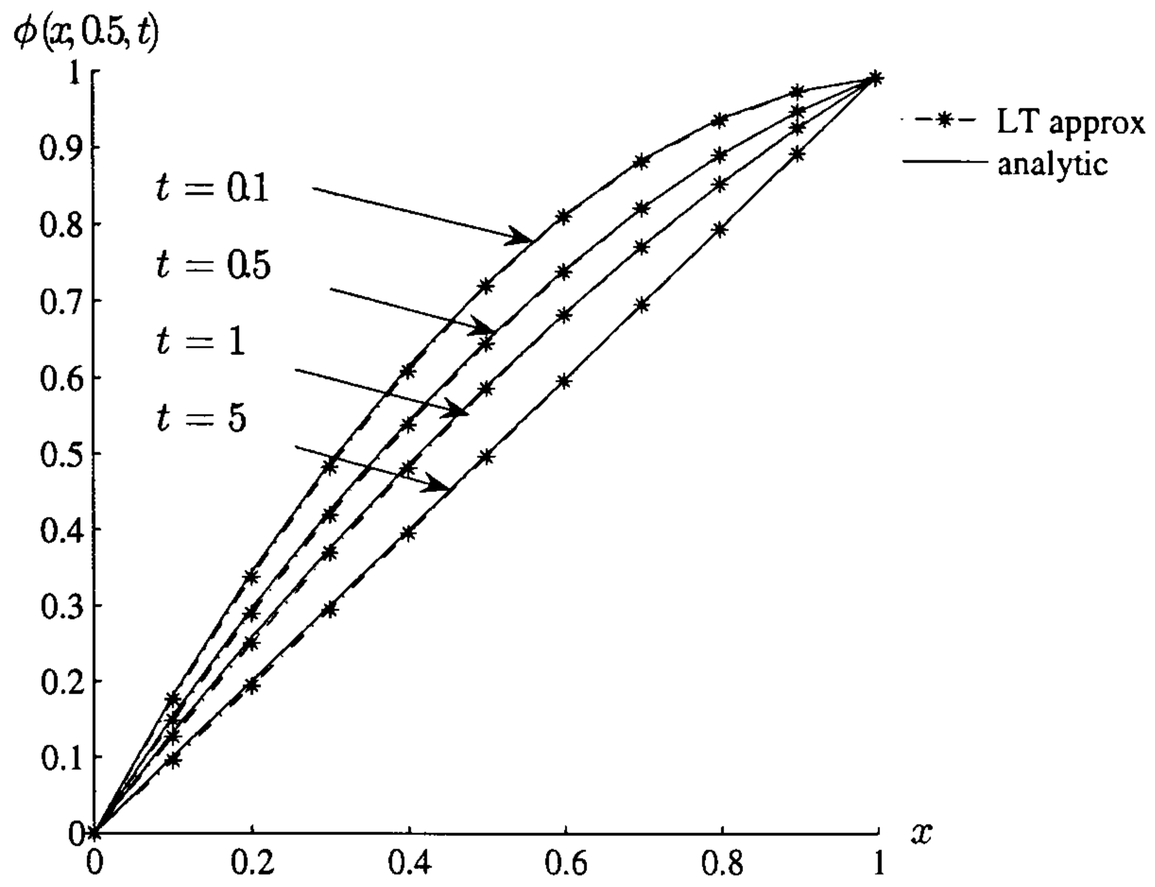

Figure 9.8: Space distribution of $\phi(x, y, t)$ for Example 9.5

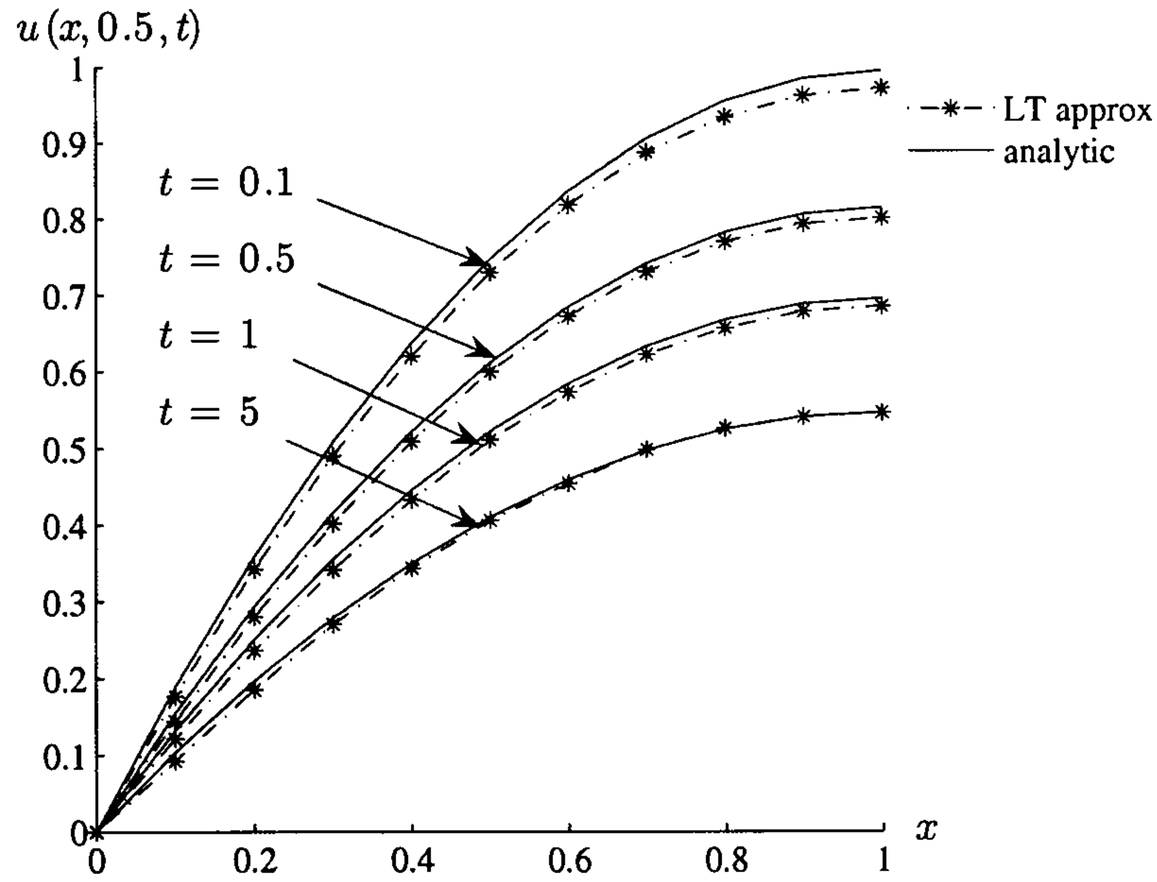

Figure 9.9: Space distribution of $u(x, y, t)$ for Example 9.5

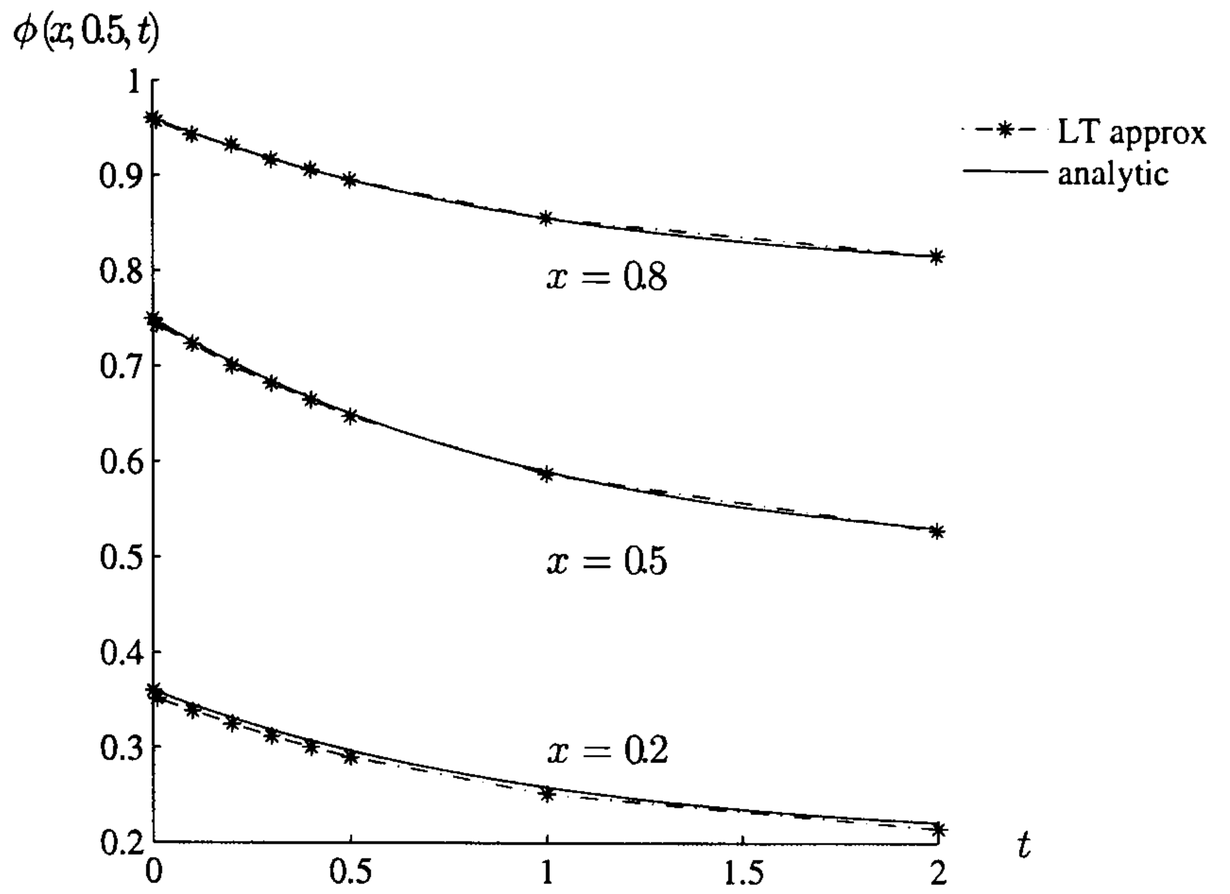

Figure 9.10: Time development of $\phi(x, y, t)$ for Example 9.5

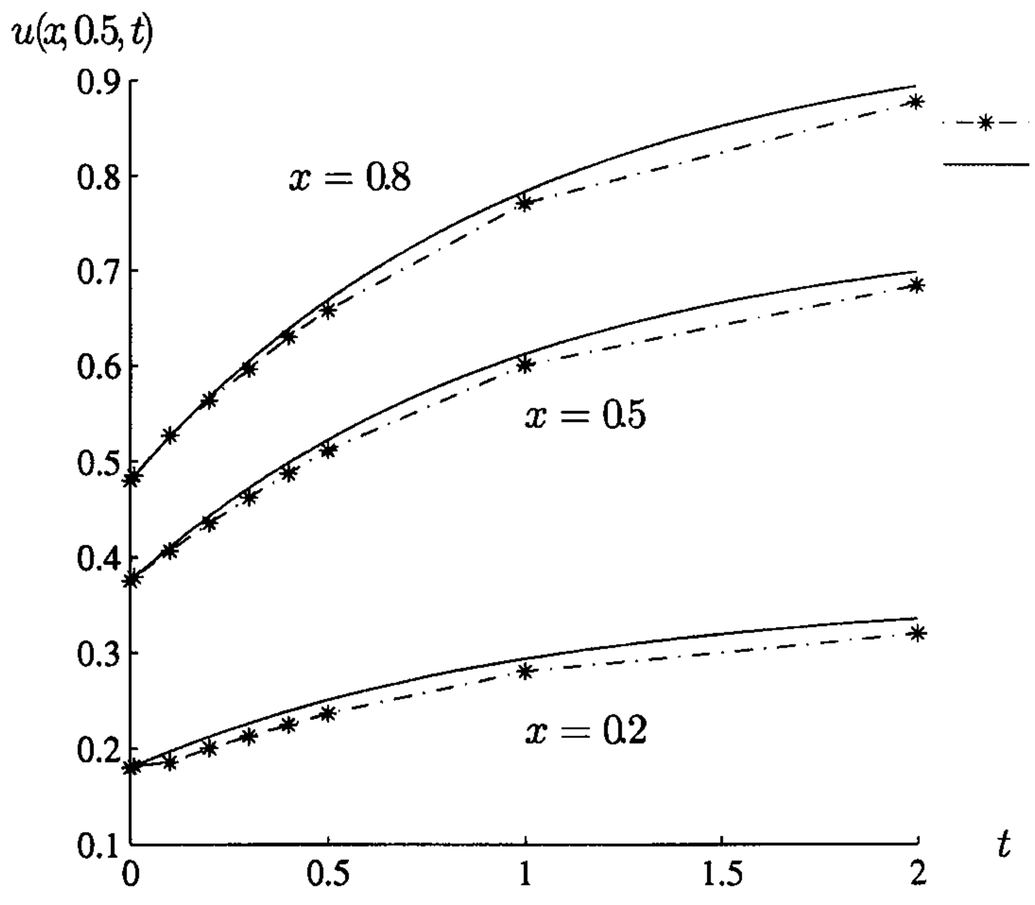

Figure 9.11: Time development of $u(x, y, t)$ for Example 9.5

Toutip (2001) considered this problem using an explicit finite difference method in time together with the dual reciprocity method. Our results are comparable with his. However, to ensure stability he used a time-step of $\Delta t = 0.01$ requiring a significant amount of computation time.

9.4 Summary of Chapter 9

In this chapter we have shown that the Laplace transform boundary element method with dual reciprocity for non-homogeneous terms provides a suitable technique for solving non-linear Poisson-type problems. However, there is the necessity to find a suitable linearisation which leads to a convergent solution in the transform domain. No such linearisation is needed with finite differences and finite elements but a solution of a non-linear system of equations is required at each stage. A feature for future work will be to consider a detailed comparison of the different solution schemes.

Problems in the food processing industry with coupled non-linear Poisson-type equations are of particular interest and have been shown to be suitable for a solution by our method. However real problems are likely to have significantly more complicated geometry and food products frequently contain multi-phase materials. The geometry should cause little difficulty because the boundary element method is ideally suited to handling complex geometry. Multi-phase problems offer a more significant challenge but domain decomposition approaches (Davies and Mushtaq 1997, Popov and Power 1999) offer a possible way forward.

Chapter 10

Conclusions and further work

10.1 Summary of thesis

This chapter outlines the main contributions of the research programme; what has been done, the difficulties encountered, decisions made and how results from examples have demonstrated these findings. This chapter also outlines the research objectives stated in Chapter 1 and shows how these objectives have been met and how they have led to further ideas and work.

The main feature of this work is the implementation of sequential and parallel code to use the Laplace transform boundary element method for the solution of initial boundary-value problems. The thesis begins in the early chapters with the classification of partial differential equations and describes ways in which they may be solved. The boundary element method (BEM) is chosen for the basis of this particular research work and its history and development is described with an explanation of the theory behind the method.

The Laplace transform method (LTM) is a valuable tool in the implementation of time-dependent problems and this is introduced with its early

background and applications. The LTM can transform a parabolic problem from a time and space domain into a space-only domain, thereby reducing the problem by one variable. The transformed problem can be solved using one of a number of solution processes and then inverted back into the time domain. There are various inversion processes and two real-variable methods are investigated for accuracy and efficiency. A number of problems are solved by the Laplace transform method using sequential and parallel implementations very successfully.

The LTM with the BEM (LTBEM) has been found to be accurate, efficient and useful for many parabolic problems with boundary and initial conditions where the initial condition is zero and thereby resulting in a solution of a homogeneous elliptic equation.

However when the elliptic equation is non-homogeneous a further refinement to the solution process needs to be made and the dual reciprocity method is used to handle the non-zero right-hand side. Thus the LTBEM with dual reciprocity has been thoroughly investigated on a variety of problems. Linear and non-linear problems have been solved. Problems with discontinuous or periodic boundary conditions have been considered. Finally a coupled non-linear system of equations has been solved successfully.

10.1.1 Difficulties encountered

One of the problems encountered in the BEM is the evaluation of singular integrals which occur when the integration and source points coincide. Chapter 4 concentrates on a number of methods of handling this non-singularity. A new idea using automatic differentiation was developed and thoroughly investigated. Accuracy was very good when compared with conventional methods and convergence criteria were introduced to aid use. However efficiency when using current LTBEM code, compared with some other methods, was not as good and it was decided not to use the new method at this

time. Telles method was considered the most suitable and this was used throughout the investigation of problems using the LTBEM. However when Toutip's sub-routine for the dual reciprocity was used the singular integrals were evaluated using Log-Gauss.

Problem The evaluation of singular integrals.

Decision The use of Telles method for the LTBEM or Log-Gauss for the LTBEM with dual reciprocity.

A problem with the Laplace transform method is the choice of an inversion process which is accurate, efficient and tracks the solution to the initial boundary-value problem. When the parabolic problem and its conditions are continuous and non-oscillatory in time two straight-forward, easy-to-use inversion methods using real variables, Stehfest's inversion method and a method based on shifted Legendre polynomials, were found to be suitable. Both methods were investigated and found to be robust and accurate for various parameters, but Stehfest's method was easier to implement.

Problem The choice of inversion method.

Decision Stehfest's inversion method with parameter $M = 8$.

However for other problems, either with discontinuities or oscillatory solutions, the inversion methods do not track the solution process. A new idea, the Step LT formulation, was considered and implemented and results were extremely good. This idea was used sequentially and in parallel to solve a variety of ordinary and partial differential equations.

Problem Poor solution of problems with non-monotonic boundary conditions.

Decision The Step LT formulation of the LTBEM.

The dual reciprocity method is a technique by which the domain integral is transferred to an equivalent boundary integral using a suitable interpolation function. Various interpolation functions can be used and often a

function from the series $f = 1 + R + R^2 + R^3 + \ldots + R^m$, where R is the distance function used in the definition of the fundamental solution, is considered. The simple function with $m = 1$, $f = 1 + R$, is often used. Alternatively the augmented thin plate spline, $f = R^2 \ln R + a + bx + cy$ has been found to be useful and, in general, the augmented thin plate spline gives the more accurate results. Since we use Toutip's sub-routine, both functions are available and we use them as stated in Chapter 7.

Problem The choice of interpolation function in the dual reciprocity method.

Decision Either $f = 1 + R$ or augmented thin plate spline according to the problem being solved.

There are two possible difficulties which occur when using the dual reciprocity method, the size of the geometry of the domain and the number of internal nodes within the domain.

In Chapter 7 we considered the size of the geometry of the problem domain and found that, particularly when using the augmented thin plate spline as interpolating function, the size of the domain was crucial to whether the solution was possible. If the geometry was suitably scaled, see Examples 7.1 and 7.8, the solution was very good.

Problem Poor results if the size of the geometry of the problem is large.

Decision Suitable scaling down to give accurate results.

When using the dual reciprocity method, various authors mention that the number of internal nodes should be greater than half the number of boundary nodes to obtain good results and have given experimental results supporting this. However, our results have not found this to be a problem. Most of our examples use 32 boundary nodes and 9 internal nodes with good results. When comparing our method with methods from other authors we have sometimes used more boundary nodes to compare our results, like for like. However we haven't found it necessary in any of our examples.

Problem The choice of the number of internal nodes compared with the number of boundary nodes.

Decision This hasn't been a problem.

The LTBEM is considered a suitable method for the solution of linear parabolic problems, since the Laplace transform is a linear operator. However, we develop an iterative process for use in non-linear problems in which the equation is linearised so that the Laplace transform can be used at each iterative step. We consider three simple iterative processes and report good results with each of them.

Problem The solution of non-linear problems.

Decision The development of three linear iterative processes.

10.2 Research objectives

Our objectives at the beginning of this research programme were, from Chapter 1:

1. To investigate the LTBEM for accuracy when considering numerical inversion methods,

2. To investigate the LTBEM for accuracy when considering non-monotonic boundary conditions,

3. To investigate the LTBEM on a distributed memory architecture for efficiency of computation.

We now consider each objective and demonstrate that they have been suitably addressed.

10.2.1 To investigate the LTBEM for accuracy when considering numerical inversion methods

There are many inversion processes for Laplace transforms. Davies and Martin (1979) give a very good account of a number of them, most containing complex variables, and they report that no one inversion method is suitable for all transforms in consideration of accuracy, efficiency and ease of implementation. They suggest that a method should be used according to the functional behaviour and if this is unknown then verification sought from a different method. However for this research we have considered known solution behaviour and sought to choose a straight-forward to use and implement inversion process. In Chapter 5 two inversion processes were considered and investigated. Both methods used real variables for the inversion and these were found to give accurate solutions under certain conditions. Test Laplace inversions were evaluated for accuracy and the results reported are very good.

10.2.2 To investigate the LTBEM for accuracy when considering non-monotonic boundary conditions

The conditions under which the chosen inversion process, Stehfest's method, gave accurate results was for problems requiring continuous boundary conditions and/or solutions and non-sinusoidal solutions, and these have been well documented by previous authors. However this research has developed methods to overcome these problems, using Step LT solutions, enabling the LTBEM to be used for problems not previously considered.

10.2.3 To investigate the LTBEM on a distributed memory architecture for efficiency of computation

In Chapter 6 we demonstrate the use of parallel computation. The Laplace transform method was used for the solution of a simple parabolic prob-

lem and the resulting elliptic problem solved using five different methods, then inverted using Stehfest's inversion method. Computation times on four processors of a transputer network were reported and speed-up, defined by the computing time of one processor divided by the total computing time , was found to be linear *i.e.* doubling the number of processors halves the computing time.

The same problem was solved using the LTBEM to investigate the speed-up using a second parallel network of eight processors on a SUN cluster but this time using different Stehfest M-parameters in the inversion process. Again the speed-up for the four processors was linear but for the SUN cluster the results showed some degradation in performance from two to four processors. The problem was assumed to be from the PVM message passing protocol rather than the machine.

The problem was again solved on a sixty-four processor *nCube* machine and there was once again almost perfect linear speed-up. This work has shown that the numerical Laplace transform using Stehfest's inversion process is ideally suited to implementation on a distributed memory architecture.

10.2.4 Further work also developed

Whilst in the development of this research other ideas have been proposed and followed up although not within our initial objectives. The work undertaken on singular integrals was a significant achievement and has produced ideas which can be taken further in a number of ways. The use of Taylor polynomials to programme complete code for various solution processes rather than only for small subroutines within a large programme might be more efficient. Certainly as far as accuracy is concerned the process is acceptable.

The use of the dual reciprocity method has enabled non-homogeneous

problems to be considered and new work has been completed with the solution of non-linear problems and coupled problems.

Although this doesn't seem to have been reported by other authors, we have sometimes found that our numerical Laplace transform inversion method yields poor results for small values of time. If small values of time are the only thing of interest then it would be best to use the FDM approach which would require only a small number of time steps. If, however, the solution was required for a larger time value then the Laplace transform approach offers a very attractive alternative to the FDM.

10.2.5 Published work

We list here the publications which have come from this research and briefly highlight the content referring to the relevant section.

1. Crann D (2005) Numerical studies using the Laplace transform, *University of Hertfordshire Department of Physics, Astronomy and Mathematics Technical Report*, **91**.
   Technical report reporting the examples and their numerical results from this thesis.
   Section 1.2

2. Davies A J and Crann D (2000) Alternative methods for the numerical solution of partial differential equations: the method of fundamental solutions and the multiquadric method, *University of Hertfordshire Mathematics Department Technical Report*, **57**.
   Report and results on the use of mesh-free methods for the solution of partial differential equations.
   Section 2.2.4

3. Davies A J and Crann D (1998) The boundary element method on a spreadsheet, *Int. J. Math. Educ. Sci. Technol.*, **29**, 851-865.
   Paper on the numerical implementation of the BEM.
   Section 3.3

4. Crann D, Christianson D B, Davies A J and Brown S A (1997) Automatic differentiation for the evaluation of singular integrals in two-dimensional boundary element computations, *Boundary Elements XIX*, eds. Marchetti M, Brebbia C A and Aliabadi M H, 677-686, Computational Mechanics Publications.

Paper on the AD Taylor polynomial method for the evaluation of singular integrals, for Laplace's equation.
Section 4.5, 4.8

5. Crann D, Christianson D B, Davies A J and Brown S A (1998) Automatic differentiation for the evaluation of singular integrals in two-dimensional boundary element computations, *University of Hertfordshire Mathematics Department Technical Report*, **41**.
Report on the AD Taylor polynomial method for the evaluation of singular integrals, for Laplace's equation and Helmholtz equation with results.
Section 4.6, 4.7, 4.8

6. Crann D, Davies A J and Christianson D B ((2003) Evaluation of logarithmic integrals in two-dimensional boundary element computation, *Advances in Boundary Element Techniques IV*, eds. Gallego R and Aliabadi M H, 321-326, Queen Mary, University of London.
Paper on the comparison of four methods of evaluating singular integrals for accuracy and efficiency.
Section 4.9

7. Crann D, Davies A J, Lai C-H and Leong S H (1998) Time domain decomposition for European options in financial modelling, *Domain Decomposition Methods 10*, eds. Mandel, Farhat and Cai, 486-491, John Wiley and Sons Ltd.
Paper using the Laplace transform in financial modelling.
Section 5.1

8. Davies A J and Crann D (2004) *A handbook of essential mathematical formulae*, University of Hertfordshire Press.
An extensive table of Laplace transforms.
Section 5.1, 5.4

9. Lai C-H, Crann D and Davies A J (2005) On a Parallel Time-domain Method for the non-linear Black-Scholes Equation, to appear in *Domain Decomposition Methods 16.*
Paper on the parallel investigation of Stehfest's Laplace transform inversion parameter during the solution process of the non-linear Black-Scholes equation.
Section 5.1

10. Crann D (1996) The Laplace transform: numerical inversion of computational methods, *University of Hertfordshire Mathematics Department Technical Report*, **21**.
Investigation into the optimal parameter in Stehfest's Laplace trans-

form inversion method.
Section 5.3.1, 5.3.3, 5.6

11. Crann D, Davies A J and Mushtaq J (1998) Parallel Laplace transform boundary element methods for diffusion problems, *Boundary Elements XX*, eds. Kassab A, Brebbia C A and Chopra M, 259-268, Computational Mechanics Publications.
Paper using LTBEM in parallel to compare the inversion methods by Stehfest and the SLP.
Section 5.3.3, 6.6

12. Davies A J and Crann D (1999) The solution of differential equations using numerical Laplace transforms, *Int. J. Math. Educ. Sci. Technol.*, **30**, 65-79.
Paper on the Laplace transform FDM for ordinary differential equations, including a discontinuous forcing term.
Section 5.4

13. Davies A J, Crann D and Mushtaq J (1996) A parallel implementation of the Laplace transform BEM, *Boundary Elements XVIII*, eds. Brebbia C A, Martins J B, Aliabadi M H and Haie N, 213-222, Computational Mechanics Publications.
Paper on a parallel implementation of the LTBEM using four transputers and eight SUN workstations.
Section 6.6

14. Davies A J, Mushtaq J, Radford L E and Crann D (1997) The numerical Laplace transform solution method on a distributed memory architecture, *Applications of High Performance Computing V*, 245-254.
Paper on the parallel implementation of the Laplace transform method with five different solvers.
Section 6.6

15. Davies A J, Crann D and Mushtaq J (2000) A parallel Laplace transform method for diffusion problems with discontinuous boundary conditions, *Applications of High Performance Computing in Engineering VI*, eds. Ingber M, Power H and Brebbia C A, 3-10, WIT press.
Paper using a parallel implementation of the Laplace transform and FDM for the solution of a diffusion problem with a discontinuous boundary condition.
Section 6.6

16. Davies A J and Crann D (2001) Parallel Laplace transform methods for boundary element solutions of diffusion-type problems, *Advances in Boundary Element Techniques II*, 183-190,Hoggar.

Paper on the parallel implementation of the LTBEM on a 64 processor *nCube* machine.
Section 6.6

17. Crann D and Davies A J (2004a) The Laplace transform boundary element method for diffusion problems with discontinuous boundary conditions, *Advances in Boundary Element Techniques V*, 249-254.
Paper on the LTBEM for discontinuous boundary conditions.
Section 8.2

18. Crann D and Davies A J (2004b) The Laplace transform boundary element method for diffusion problems with periodic boundary conditions, *Boundary Elements XXVI*, 393-402.
Paper on the LTBEM for problems with periodic boundary conditions.
Section 8.3

19. Crann D, Davies A J and Christianson D B (2005) The Laplace transform dual reciprocity boundary element method for electromagnetic heating problems - to appear in *Advances in Boundary Element Techniques VI.*
Paper on the LTBEM for a non-linear coupled problem.
Section 9.3

10.3 Future research work

Some features of this research have an obvious initial improvement and work is already being started to refine these features, such as updating the present code to enable the augmented thin plate spline to be used for the solution of the first derivative in the dual reciprocity method and to see if the use of Telles method for singular and non-singular integrals is computationally more efficient.

The research objectives have been completed and the following new ideas await to be addressed:

1. Can we use automatic differentiation for near-singular integrals and the whole solution processes?

2. What are the convergence criteria for Stehfest's method and what is the behaviour of the errors?

3. Can we explain why for problems with sinusoidal boundary conditions the time step needs to be one quarter of the time period?

4. Which interpolation functions can be used in the dual reciprocity method to enable us to solve problems containing a second derivative on the right-hand side?

5. Although the Laplace transform method doesn't always give accurate results for small time-steps, how does the Laplace transform with the BEM compare with the Laplace transform and other solution processes for accuracy and efficiency in general?

6. Can we use more efficient iterative schemes in the solution of non-linear problems?

7. Can we use our method yet to solve other real-life problems, in the financial sector or the food processing industry? Are there other practical uses for our solution process?

Chapter 11

References

Abell M L and Braselton J P (1994) *Maple® by example*, Academic Press.

Abramowitz M and Stegun A (1972) *Handbook of Mathematical Functions*, Dover.

Ademoyero O O (2003) *A parallel Galerkin boundary element method*, PhD Thesis, University of Hertfordshire.

Aliabadi M H (2002) *The Boundary Element Method Volume 2*, Wiley.

Aral M M and Gülçat U (1977) A finite element Laplace transform solution technique for the wave equation, *Int. J. Num. Meth. Engng.*, **11**, 1719-1732.

Bartholomew-Biggs M, Brown S, Christianson B and Dixon L (2000) Automatic differentiation of algorithms, *Jnl. Comp. App. Maths.*, **124**, 171-190.

Beale J T and Attwood M J (2002), Evaluating Singular and Nearly Singular Integrals Numerically, unpublished work, Duke University.

Beale J T and Lai M-C (2001) A method for computing nearly singular integrals, *SIAM Jnl. of Num. Anal.*, **38**,1902-1925.

Becker A A (1992) *The boundary element method in engineering*, McGraw-Hill.

Bekker A A (2003) *Why Do ... Boundary element Analysis*, NAFEMS Ltd.

Brebbia C A (1978) *The boundary element method for engineers*, Pentech Press.

Brebbia C A and Dominguez J (1977) Boundary element method for potential problems, *Appl. Math. Modelling*, **1**, 372-278.

Brebbia C A and Dominguez J (1989) *Boundary elements, an introductory course*, (2nd ed. 1992), McGraw-Hill/Computational Mechanics Publications.

Brebbia C A, Telles J C F and Wrobel L C (1984) *Boundary Element Techniques*, Springer-Verlag, Berlin and New York.

Broyden C G and Vespucci M T (2004) *Krylov solvers for linear algebraic systems*, Elsevier.

Bucher H F and Wrobel L C (2001) A novel approach to applying fast wavelet transforms in the boundary element method, *Advances in Boundary Element Techniques II*, Hoggar Press, Geneva.

Carslaw H S and Jaeger J C (1959) *Conduction of Heat in Solids*, 2nd ed., Oxford University Press.

Chang Y-P, Kang C S and Chen P J (1973) The use of fundamental Green's functions for the solution of problems of heat conduction in anisotropic media, *Int. J. Heat Mass Transfer*, **16**, 1905-1918.

Chantasiriwan S (2004) Cartesian grid methods using radial basis functions for solving Poisson, Helmholtz and diffusion-convention equations, *Engng. Anal. with Boundary Elements*, **28**,1417-1425.

Chen H-T and Lin J-Y (1991) Application of the Laplace transform to nonlinear transient problems *Appl. Math. Modelling*, **15**, 144-151.

Cheng A H-D and Cheng D T (2005) Heritage and early history of the boundary element method, *Engng. Anal. with Boundary Elements*, **29**, 268-302.

Crank J (1975) *The mathematics of diffusion*, Oxford University Press.

Crann D (1996) The Laplace transform: numerical inversion of computational methods, *University of Hertfordshire Mathematics Department Technical Report*, **21**.

Crann D (2005) Numerical studies using the Laplace transform, *University of Hertfordshire Department of Physics, Astronomy and Mathematics Technical Report*, **91**.

Crann D, Christianson D B, Davies A J and Brown S A (1997) Automatic differentiation for the evaluation of singular integrals in two-dimensional boundary element computations, *Boundary Elements XIX*, eds. Marchetti M, Brebbia C A and Aliabadi M H, 677-686, Computational Mechanics Publications.

Crann D, Christianson D B, Davies A J and Brown S A (1998) Automatic differentiation for the evaluation of singular integrals in two-dimensional boundary element computations, *University of Hertfordshire Mathematics Department Technical Report*, **41**.

Crann D and Davies A J (2004a) The Laplace transform boundary element method for diffusion problems with discontinuous boundary conditions, *Advances in Boundary Element Techniques V*, 249-254.

Crann D and Davies A J (2004b) The Laplace transform boundary element method for diffusion problems with periodic boundary conditions, *Boundary Elements XXVI*, 393-402.

Crann D, Davies A J and Christianson D B (2003) Evaluation of logarithmic integrals in two-dimensional boundary element computation, *Advances in Boundary Element Techniques IV*, eds. Gallego R and Aliabadi M H, 321-326, Queen Mary, University of London.

Crann D, Davies A J and Christianson D B (2005) The Laplace transform dual reciprocity boundary element method for electromagnetic heating problems - to appear in *Advances in Boundary Element Techniques VI.*

Crann D, Davies A J, Lai C-H and Leong S H (1998) Time domain decomposition for European options in financial modelling, *Domain Decomposition Methods 10*, eds. Mandel, Farhat and Cai, 486-491, John Wiley and Sons Ltd.

Crann D, Davies A J and Mushtaq J (1998) Parallel Laplace transform boundary element methods for diffusion problems, *Boundary Elements XX*, eds. Kassab A, Brebbia C A and Chopra M, 259-268, Computational Mechanics Publications.

Crow J A (1993) Quadratic integrands with a logarithmic singularity, *Math. Comp.*, **60**, 297-301.

Cruse T A (1969) Numerical solution in three dimensional elastostatics, *Int. Jnl. Solids and Structures*, **5**, 1259-1374.

Curran D A S, Cross M and Lewis B (1980) Solution of parabolic differential equations by the boundary element method using discretization in time, *Appl. Math. Modelling*, **4**, 398-400.

Davies A J (1985) *The finite element method: a first approach*, Oxford University Press.

Davies A J (1988a) The boundary element method on the ICL DAP, *Parallel Computing*, **8**, 335-343.

Davies A J (1988b) Quadratic isoparametric boundary elements: an implementation on the ICL DAP, *Boundary Elements X*, ed. Brebbia C A, **3**, 657-666, Springer-Verlag.

Davies A J (1988c) Mapping the boundary element to the ICL DAP, *CONPAR88*, eds. Hesshope C R and Reinartz K D, 230-237.

Davies A J (1989) *Aspects of the boundary integral equation method and its implementation on a distributed array processor*, PhD Thesis, University of London.

Davies A J and Crann D (1998) The boundary element method on a spreadsheet, *Int. J. Math. Educ. Sci. Technol.*, **29**, 851-865.

Davies A J and Crann D (1999) The solution of differential equations using numerical Laplace transforms, *Int. J. Math. Educ. Sci. Technol.*, **30**, 65-79.

Davies A J and Crann D (2000) Alternative methods for the numerical solution of partial differential equations: the method of fundamental solutions and the multiquadric method, *University of Hertfordshire Mathematics Department Technical Report*, **57**.

Davies A J and Crann D (2001) Parallel Laplace transform methods for boundary element solutions of diffusion-type problems, *Advances in Boundary Element Techniques II*, eds. Denda M, Aliabadi M H and Charafi A, 183-190,Hoggar.

Davies A J and Crann D (2004) *A handbook of essential mathematical formulae*, University of Hertfordshire Press.

Davies A J, Crann D and Mushtaq J (1996) A parallel implementation of the Laplace transform BEM, *Boundary Elements XVIII*, eds. Brebbia C A, Martins J B, Aliabadi M H and Haie N, 213-222, Computational Mechanics Publications.

Davies A J, Crann D and Mushtaq J (2000) A parallel Laplace transform method for diffusion problems with discontinuous boundary conditions, *Applications of High Performance Computing in Engineering VI*, eds. Ingber M, Power H and Brebbia C A, 3-10, WIT Press.

Davies A J, Mushtaq J, Radford L E and Crann D (1997) The numerical Laplace transform solution method on a distributed memory architecture, *Applications of High Performance Computing V*, eds. Power H and Long J J, 245-254, Computational Mechanics Publications.

Davies B (2002) *Integral Transforms and Their Applications*, Third Edition, Springer.

Davies B and Martin B (1979) Numerical Inversion of the Laplace Transform: a Survey and Comparison of Methods, *Jnl. of Comp. Physics*, **33**, 1-32.

de Alwis A A P and Fryer P J (1990) A finite element analysis of heat generation in the food industry, *Chem. Eng. Sci.*, **45**, 1547-1549.

Dryden H L, Murnaghan F D and Bateman H (1956) *Hydrodynamics*, Dover.

Edelstein-Keshet L (1988) *Mathematical models in biology*, McGraw-Hill.

Elliot C M and Larsson S (1995) A finite element model for the time-dependent Joule heating problem, *Maths Com.*, **64**, 1433-1453.

Franke R (1982) Scattered data interpolation: tests of some methods, *Math. Comput.*, **38**, 181-200.

Fredholm I (1903) Sur une classe d'equations fonctionelles, *Acta Math.*, **27**, 365-390.

Gaver D P Jr (1966) Observing stochastic processes, and approximate transform inversion, *Oper. Res 14*, **3**,444-459.

Goldberg M A and Chen C S (1999) The method of fundamental solutions for potential, Helmholtz and diffusion problems, Chapter 4, *Boundary Integral Methods: Numerical and Mathematical Aspects*, Computational Mechanics Publications.

Gray L J (1993) Evaluation of singular and hypersingular Galerkin integrals: direct limits and symbolic computation, *Singular integrals in boundary element methods*, eds. Sladek V and Sladek J, Computational Mechanics Publications, 33-84.

Gray L J, Kaplan T, Richardson J D and Paulino G H (2005) Green's functions and boundary integral analysis for exponentially graded matherials: Heat conduction, *ASME Journal of Applied Mechanics* (in press).

Green G (1828) *An essay on the application of mathematical analysis to the theories of electricity and magnetism*, Longman.

Hadamard J (1923) *Lectures on Cauchy's problem in linear partial differential equations*, Dover.

Hanselman D and Littlefield B (2001) *Mastering MATLAB* ® *6*, Prentice Hall.

Harrington R F, Pontoppidan K, Abrahamsen P and Albertson N C (1969) Computation of Laplacian Potentials by an equivalent source method, *Proc. IEE*, **116**, 1715-1720.

Hess J L and Smith A M O (1964) Calculation of non-lifting potential flow about arbitrary three-dimensional bodies, *Jnl. Ship Res.*, **8**, 22-44.

Ingber M S and Davies A J (1997) High Performance Computing - Special Issue, *Engng. Anal. with Boundary Elements*, **19**.

Irons B M (1966) Engineering application of numerical integration in stiffness method, *AIAA Journal*, **14**, 2035-7.

Irons B M (1970) A frontal solution program, *Int. J. Num. Meth. Eng.*, **2**, 5-32.

Jakob M (1949) *Heat transfer*, John Wiley and Sons.

Jameson A and Mavriplis D (1986) Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh *AIAA Journal*, **24**, 611-618.

Jaswon M A (1963) Integral equations in potential theory I, *Proc. Roy. Soc. Lon.*, **A275**, 23-32.

Jaswon M A and Ponter A R (1963) An integral equation solution of the torsion problem, *Proc. Roy. Soc. Lon.*, **A275**, 237-246.

Jaswon M A and Symm G T (1977) *Integral equation methods in potential theory and elastostatics*, Academic Press.

Kellog O D (1929) *Foundations of potential theory*, Springer, Berlin.

Kupradze O D (1965) *Potential methods in the theory of elasticity*, Daniel Davy, New York.

Kythe P K (1996) *Fundamental solutions for differential operators and applications*, Birkhäuser Boston.

Lachat J C and Combescure A (1977) Laplace transform and boundary integral equation applications to transient heat conduction problems, *Proc. First Symp. on Innovative Numerical Analysis in Applied Engineering*, CETIM, Versailles, 1.57-1.62.

Lachat J C and Watson J O (1976) Effective numerical treatment of boundary integral equations: a formulation for three-dimensional elastostatics, *Int. J. Num. Meth. Engng.*, **10**, 991-1005.

Lai C-H, Crann D and Davies A J (2005) On a Parallel Time-domain Method for the non-linear Black-Scholes Equation, to appear in *Domain Decomposition Methods 16.*

Lai C-H and Liddell H M (1987) A review of parallel finite element methods on the DAP, *App. Math. Modelling*, **11**, 330-340.

Lamb H (1932) *Hydrodynamics*, Cambridge University Press.

Logan J D (1994) *An introduction to nonlinear partial differential equations*, John Wiley and Sons.

Love A E H (1927) *A treatise on the mathematical theory of elasticity*, Dover.

Liu G-R (2003) *Mesh free methods: moving beyond the finite element method*, CRC Press LLC.

Mammoli A A and Ingber M S (1999) Stokes flow around cylinders in a bounded two-dimensional domain using multipole-accelerated boundary element methods, *Int. J. Num. Meth. Engng.*, **44**, 897-917.

Mikhlin S G (1957) *Integral equations*, Pergamon, London.

Moridis G J and Reddell D L (1991a) The Laplace Transform Finite Difference (LTFD) Method for Simulation of Flow through Porous Media, *Water Resources Research*, **27**, 1873-1884.

Moridis G J and Reddell D L (1991b) The Laplace Transform Finite Element (LTFE) Numerical Method for the Solution of the Groundwater Equation, paper H22C-4, ASGU91 Spring Meeting, *EOS Trans. of the AGU*, **72** (17).

Moridis G J and Reddell D L (1991c) The Laplace Transform Boundary Element (LTBE) Numerical Method for the solution of diffusion-type problems, *Boundary Elements XIII*, eds. Brebbia C A and Gipson G S, 83-97, Elsevier.

Motz H (1946) The treatment of singularities of partial differential equations by relaxation methods, *Qtrly. App. Maths.*, **4**, 371-377.

Muskhelishvili N I (1953) *Singular integral equations*, Noordhoff, Gronongen.

Nardini D and Brebbia C A (1982) A new approach to free vibration analysis using boundary elements, *Boundary Element Methods in Engineering*, ed. Brebbia C A, Springer-Verlag, Berlin.

Nardini D and Brebbia C A (1985) Boundary integral formulation of mass matrices for dynamic analysis, *Topics in Boundary Element Research*, **2**, Springer-Verlag, Berlin.

Natalini B and Popov V (2005) On the optimal implementation of the boundary element dual reciprocity method - multi domain approach

for 3D potential problems - to appear in *Computer Methods in Applied Mechanics and Engineering*, eds. Hughes T J R, Oden J T and Papadrakakis M, Elsevier.

Ortega J M and Voigt R G (1985) Solution of partial differential equations on vector and parallel computers, *SIAM Review*, **27**, 1-96.

París F and Cañas J (1997) *Boundary element method: Fundamentals and applications*, Oxford University Press.

Partridge P W and Brebbia C A (1989) Computer implementation of the BEM dual reciprocity method for the solution of Poisson type equations, *Software for Engineering Workstations*, **5** (4), 199-206.

Partridge P W, Brebbia C A and Wrobel L C (1992) *The Dual Reciprocity Boundary Element Method* Computational Mechanics Publications and Elsevier Applied Science.

Partridge P W and Wrobel L C (1990) The dual reciprocity method for spontaneous ignition, *Int. J. Meth. Engng.*, **30**, 953-963.

Please C P, Schwendeman D W and Hagan P S (1995) Ohmic heating of foods during aseptic processing, *IMA J. Maths Appl. Bus. & Ind.*, **5**, 283-301.

Popov V and Power H (1999) The DRM-MD integral equation method: an efficient approach for the numerical solution of domain dominant problems, *Int. J. Num. Meth. Engng.*, **44**, 327-353.

Popov V and Power H (2001) An $O(N)$ Taylor series multipole boundary element method for three-dimensional elasticity problems, *Engng. Anal. with Boundary Elements*, **25**, 7-18.

Ramesh P S and Lean M H (1991) Accurate integration of singular kernels in boundary integral formulations for Helmholtz equations, *Int. J. Num. Meth. Engng.*, **31**, 1055-1068.

Renardy M and Rogers R C (1993) *An introduction to differential equations*, Springer-Verlag.

Rizzo F J (1967) An integral equation approach to boundary value problems, *Qtrly. App. Maths.*, **25**, 83-95.

Rizzo F J and Shippy D J (1970) A method of solution of certain problems of transient heat conduction, *A. I. A. A. Journal*, **8**, 2004-2009.

Rubinstein Z (1969) *A Course in Ordinary and Partial Differential Equations*, Academic Press Inc.

Schapery R A (1962) Approximate Methods of Transform Inversion for Viscoelastic Stress Analysis, *Proc. Fourth US National Congress on Applied Mechanics*, **2**, 1075-2085.

Smith G D (1978) *Numerical solution of partial differential equations: finite difference methods*, Second edition, Oxford University Press.

Smith R N L (1996) Simplifying integration for logarithmic singularities, *Boundary Elements XVIII*, eds. Brebbia C A, Martius J B, Aliabadi M H and Haie N, 233-240, Computational Mechanics Press.

Smith R N L and Mason J C (1982) A boundary element method for curved crack problems in two dimensions, *Boundary element methods in engineering*, ed. Brebbia C A, 472-484, Springer-Verlag.

Sokolnikoff I S (1956) *Mathematical theory of elasticity*, McGraw-Hill.

Stehfest H (1970) Numerical inversion of Laplace transforms, *Comm. ACM*, **13**, 47-49 and 624.

Stratton J A (1941) *Electromagnetic theory*, McGraw-Hill.

Stroud A H and Secrest D (1966) *Gaussian quadrature formulas*, Prentice-Hall.

Symm G T (1963) Integral equation methods in potential theory II, *Proc. Roy. Soc. Lon.*, **A275**, 33-46.

Symm G T (1984) Boundary elements on a distributed array processor, *Engng. Anal.*, **1**, 162-165.

Telles J C F (1987) A self-adaptive co-ordinate transformation for efficient numerical evaluation of general boundary element integrals, *Int. J. Num. Meth. Engng.*, **24**, 959-973.

Toutip W (2001) *The dual reciprocity boundary element method for linear and non-linear problems*, PhD Thesis, University of Hertfordshire.

Whiteman J R and Papamichael N (1972) Treatment of harmonic mixed boundary problems by conformal transformation methods, *Jnl. App. Maths. and Phys.*, **23**,655-664.

Williams W E (1980)*Partial differential equations*, Oxford University Press.

Wilmott P, Howison S and Dewynne J (1995) *The mathematics of Financial Derivatives*, Cambridge University Press.

Wing G M (1991) *A primer on integral equations of the first kind: the problem of deconvolution and unfolding*, with the assistance of Zahrt J D, SIAM.

Wrobel L C (2002) *the Boundary Element Method Volume 1*, Wiley.

Zakian V and Littlewood R K (1973) Numerical inversion of Laplace transforms by weighted least-squares approximation, *Comp. J.*, **16**, 66-68.

Zhu S-P (1999) Time-dependent reaction-diffusion problems and the LT-DRM approach, *Boundary Integral Methods, Numerical and Mathematical Aspects*, ed. Goldberg M, 1-35, Computational Mechanics Publications.

Zhu S, Satravaha P and Lu X (1994) Solving linear diffusion equations with the dual reciprocity method in Laplace space, *Engng. Anal. with Boundary Elements*, **13**, 1-10.

Zienkiewicz O C and Cheung Y K (1965) Finite elements in the solution of field problems, *The Engineer*, **220**, 507-510.

Zienkiewicz O C and Taylor R L (2000) *The Finite Element Method*, Vols. 1, 2, 3, Fifth edition, Butterworth-Heinemann.

Appendix A

Automatic Differentiation *fortran90* constructs

In this appendix we present the *fortran90* module for evaluating Taylor polynomials. The module shows how we develop the processes of addition, subtraction, multiplication, division, square root and log, together with procedures for performing differentiation, integration and evaluation of the Bessel function.

```
module taylormod
implicit double precision(a-h,o-z)

! For Taylorprog, taylor-degree is 6 or 20
! For Taylor-Bess, taylor-degree is 13 or 21

integer, private :: taylor.degree=20;
integer::numadd,nummult,numother

! put taylor-degree integer into type(taylor) as well as above
type taylor
real series(20)
end type taylor

type(taylor)::sumA

interface operator(+)
module procedure plus.tt
end interface

interface operator(-)
module procedure minus.tt
```

```
end interface

interface operator(*)
module procedure times.tt
end interface

interface mult
module procedure mult.tt
end interface

interface div
module procedure div.tt
end interface

interface recip
module procedure recip.t
end interface

interface tsqrt
module procedure tsqrt.t
end interface

interface tlog
module procedure tlog.t
end interface

interface shleft
module procedure shleft.t
end interface

interface shright
module procedure shright.t
end interface

interface deriv
module procedure deriv.t
end interface

interface tint
module procedure tint.t
end interface

interface J1integ
module procedure J1integ.t
```

```
end interface

interface J1loginteg
module procedure J1loginteg.t
end interface

interface J2integ
module procedure J2integ.t
end interface

interface J2loginteg
module procedure J2loginteg.t
end interface

interface J3integ
module procedure J3integ.t
end interface

interface J3loginteg
module procedure J3loginteg.t
end interface

interface bessk
module procedure bessk.t
end interface


CONTAINS

subroutine init.taylor(t1) ! initialises taylor series to zero
type(taylor), intent (inout)::t1

t1%series=0.0

end subroutine init.taylor

subroutine set.taylor(t1,value,n) ! initialises taylor series with
type(taylor), intent(inout)::t1 ! values in position n
real,intent(in)::value
integer,intent(in)::n
t1%series(n)=value
end subroutine set.taylor

function plus.tt(t1,t2) ! adds two taylor series together
```

```
type(taylor), intent(in)::t1,t2
type(taylor)::plus.tt
plus.tt%series=t1%series+t2%series
numadd=numadd+1
end function plus.tt

function minus.tt(t1,t2) ! finds the difference of two taylor
type(taylor),intent(in)::t1,t2 ! series, t1-t2
type(taylor)::minus.tt
minus.tt%series=t1%series-t2%series
numadd=numadd+1
end function minus.tt

function mult.tt(t1,t2) ! multiplies two taylor series
type(taylor),intent(inout)::t1,t2 ! together
type(taylor)::mult.tt,total
integer i,p
mult.tt%series=0.0
do p=1,taylor.degree
do i=1,p
total%series(i)=t1%series(i)*t2%series(p+1-i)
mult.tt%series(p)=mult.tt%series(p)+total%series(i)
numadd=numadd+3
nummult=nummult+1
end do
end do
end function mult.tt

function div.tt(t1,t2) ! divides two taylor series
type(taylor),intent(inout)::t1,t2 ! div(t1,t2)=t2/t1
type(taylor)::div.tt,total,newtotal
integer i,p
div.tt%series=0.0
total%series=0.0
newtotal%series=0.0
div.tt%series(1)=t2%series(1)/t1%series(1)
nummult=nummult+1
do p=2,taylor.degree
do i=1,p-1
total%series(i)=t1%series(p+1-i)*div.tt%series(i)
newtotal%series(p)=newtotal%series(p)+total%series(i)
numadd=numadd+3
nummult=nummult+1
end do
```

```
div.tt%series(p)=(t2%series(p)-newtotal%series(p))/t1%series(1)
numadd=numadd+1
nummult=nummult+1
end do
end function div.tt

function recip.t(t1) ! finds the reciprocal of
type(taylor),intent(inout)::t1 ! a taylor series
type(taylor)::recip.t,one
call init.taylor(one)
one%series(1)=1.0
recip.t=div.tt(t1,one)
end function recip.t

function times.tt(t1,n) ! multiplies a taylor series
type(taylor),intent(in)::t1 ! by a scalar
type(taylor)::times.tt
real,intent(in):: n
times.tt%series=t1%series*n
nummult=nummult+1
end function times.tt

function tsqrt.t(t1) ! finds square root of a taylor series
type(taylor),intent(in)::t1 ! constant not negative
type(taylor)::tsqrt.t,new1,new2
integer i,j
tsqrt.t%series(1)=sqrt(t1%series(1))
tsqrt.t%series(2)=t1%series(2)/(2.0*tsqrt.t%series(1))
new2%series=0.0
nummult=nummult+2
numother=numother+1
do j=3,taylor.degree
do i=2,j-1
new1%series(i)=tsqrt.t%series(i)*tsqrt.t%series(j+1-i)
new2%series(j)=new2%series(j)+new1%series(i)
tsqrt.t%series(j)=(t1%series(j)-new2%series(j))/&
&(2.0*tsqrt.t%series(1))
numadd=numadd+4
nummult=nummult+3
end do
end do
end function tsqrt.t

function tlog.t(t1) ! finds the log of a taylor series
```

```
type(taylor),intent(inout)::t1
type(taylor)::tlog.t,next1,next2,next3
next1=deriv.t(t1)
next2=recip.t(t1)
next3=mult(next1,next2)
tlog.t=tint(next3)
tlog.t%series(1)=log(t1%series(1))
numother=numother+1
end function tlog.t

function shleft.t(t1) ! shifts constants to the left
type(taylor),intent(in)::t1 ! within the taylor series
type(taylor)::shleft.t
integer i
do i=1,taylor.degree-1
shleft.t%series(i)=t1%series(i+1)
end do
end function shleft.t

function shright.t(t1) ! shifts constants to the right
type(taylor),intent(in)::t1 ! within the taylor series
type(taylor)::shright.t
integer i
do i=2,taylor.degree
shright.t%series(i)=t1%series(i-1)
end do
shright.t%series(1)=0.0
end function shright.t

function deriv.t(t1) ! finds the derivative of a
type(taylor),intent(in)::t1 ! taylor series
type(taylor)::deriv.t
integer i
do i=1,taylor.degree-1
deriv.t%series(i)=i*t1%series(i+1)
nummult=nummult+1
end do
end function deriv.t

function tint.t(t1) ! finds the integral of a
type(taylor),intent(in)::t1 ! taylor series
type(taylor)::tint.t !***the first term is set to 0.0
integer i !***set this separately when using
tint.t%series(1)=0.0
```

```
do i=2,taylor.degree
tint.t%series(i)=t1%series(i-1)/(i-1)
nummult=nummult+1
end do
end function tint.t

function J1integ.t(t1) ! finds the integral of a taylor
type(taylor),intent(in)::t1 ! series between -1 and +1 for J1
type(taylor)::J1integ.t
integer i
do i=1,taylor.degree
J1integ.t%series(i)=0.0
J1integ.t%series(1)=J1integ.t%series(1)+(2**i)*t1%series(i)/(i)
numadd=numadd+1
nummult=nummult+4
end do
end function J1integ.t

function J1loginteg.t(t1) ! finds the integral of a taylor
type(taylor),intent(in)::t1 ! series multiplied by the log
type(taylor)::J1loginteg.t ! between -1 and +1 for J1
integer i
do i=1,taylor.degree
J1loginteg.t%series(i)=0.0
J1loginteg.t%series(1)=J1loginteg.t%series(1)&
&+((2.0**i)*t1%series(i)/i)*(log(2.0)-(1.0/real(i)))
numadd=numadd+2
nummult=nummult+5
numother=numother+1
end do
end function J1loginteg.t

function J2integ.t(t1) ! finds the integral of a taylor
type(taylor),intent(in)::t1 ! series between -1 and +1 for J2
type(taylor)::J2integ.t
integer i
do i=1,taylor.degree
J2integ.t%series(i)=0.0
if (mod(i,2)==0) then
J2integ.t%series(i)=0.0
else
J2integ.t%series(1)=J2integ.t%series(1)+2*t1%series(i)/(i)
numadd=numadd+1
nummult=nummult+2
```

```
end if
end do

    end function J2integ.t

function J2loginteg.t(t1) ! finds the integral of a taylor
type(taylor),intent(in)::t1 ! series multiplied by the log
type(taylor)::J2loginteg.t ! between -1 and +1 for J2
integer i
do i=1,taylor.degree
J2loginteg.t%series(i)=0.0
if (mod(i,2)==0) then
J2loginteg.t%series(i)=0.0
else
J2loginteg.t%series(1)=J2loginteg.t%series(1)-2.0*t1%series(i)&
&/real(i)**2.0
numadd=numadd+1
nummult=nummult+4
end if
end do
end function J2loginteg.t

function J3integ.t(t1) ! finds the integral of a taylor
type(taylor),intent(in)::t1 ! series between -1 and +1 for J3
type(taylor)::J3integ.t
integer i
do i=1,taylor.degree
J3integ.t%series(i)=0.0
J3integ.t%series(1)=J3integ.t%series(1)-((-2)**i)*t1%series(i)/(i)
numadd=numadd+2
nummult=nummult+4
end do
end function J3integ.t

function J3loginteg.t(t1) ! finds the integral of a taylor
type(taylor),intent(in)::t1 ! series multiplied by the log
type(taylor)::J3loginteg.t ! between -1 and +1 for J3
integer i
do i=1,taylor.degree
J3loginteg.t%series(i)=0.0
J3loginteg.t%series(1)=J3loginteg.t%series(1)-&
&(((-2.0)**i)*t1%series(i)/i)*(log(2.0)-(1.0/i))
numadd=numadd+3
nummult=nummult+5
```

```
numother=numother+1
end do
end function J3loginteg.t

subroutine tread(t1) ! reads a taylor series from screen
type(taylor),intent(inout)::t1
real value
integer n,i
print*,'what is the degree of the taylor series?'
read*,n
print*,'type in the values'
do i=1,n
read*,value
t1%series(i)=value
end do
end subroutine tread

subroutine tprint(t1) ! prints a taylor series to screen
type(taylor), intent(in)::t1
print*,t1%series
end subroutine tprint

subroutine print(t1) ! prints a taylor series as a
type(taylor),intent(in)::t1 ! real to the screen
real a
a=t1%series(1)
print*,a
end subroutine print

function distance(a,b,c,d,e,f) ! finds the Jtest of 3 nodes
type(taylor),intent(in)::a,b,c,d,e,f
! real,intent(inout)::distance
real distance
real p,q,r,s,t,u,first,second
p=a%series(1)
q=b%series(1)
r=c%series(1)
s=d%series(1)
t=e%series(1)
u=f%series(1)
first=sqrt((q-0.5*(r+p))**2+(t-0.5*(u+s))**2)
second=0.5*(sqrt((q-p)**2+(t-s)**2))
if (first==0) then
print*,'jtest is undefined, but a lot'
```

```
else
distance=second/first
end if
numadd=numadd+8
nummult=nummult+12
numother=numother+2
end function distance

function bessk.t(Rd,p) ! Modified Bessel function
! using Ramesh and Lean's formula
type(taylor),intent(inout)::Rd
real,intent(in)::p
type(taylor)::bessk.t
type(taylor)::A,A1,B,B1
type(taylor),dimension(8)::Rdd,nextA,nextB
type(taylor)::sumB,finalA,finalB
type(taylor)::first,second,third
real::q
integer::i,j

call init.taylor(A)
A%series(1)=1.0
A%series(2)=3.5156229
A%series(3)=3.0899424
A%series(4)=1.2067492
A%series(5)=0.2659732
A%series(6)=0.0360786
A%series(7)=0.0045813

call init.taylor(B)
B%series(1)=-0.57721566
B%series(2)=0.42278420
B%series(3)=0.23069756
B%series(4)=0.03488590
B%series(5)=0.00262698
B%series(6)=0.00010750
B%series(7)=0.00000740

call init.taylor(A1)
A1%series(1)=A%series(1)
do i=2,7
A1%series(2*i-1)=A%series(i)*((p/3.75)**(2*(i-1)))
numadd=numadd+2
nummult=nummult+6
```

```
end do
call init.taylor(B1)
B1%series(1)=B%series(1)
do i=2,7
B1%series(2*i-1)=B%series(i)*((p/2.)**(2*(i-1)))
numadd=numadd+2
nummult=nummult+6
end do
call init.taylor(sumA)
call init.taylor(sumB)
nextA(1)=Rd
nextB(1)=Rd
Rdd(2)=Rd
do j=2,7
nextA(j)=Rdd(j)
nextB(j)=Rdd(j)
do i=1,2*(j-1)
nextA(j)=shright(nextA(j))
nextB(j)=shright(nextB(j))
end do
Rdd(j+1)=mult(Rdd(j),Rd)
end do
do j=2,7
nextA(j)=nextA(j)*A1%series(2*j-1)
nextB(j)=nextB(j)*B1%series(2*j-1)
sumA=sumA+nextA(j)
sumB=sumB+nextB(j)
numadd=numadd+2
end do
sumA%series(1)=A1%series(1)
sumB%series(1)=B1%series(1)
q=p*p/4.
first=Rd*q
second=tlog(first)
third=second*0.5
finalA=mult(third,sumA)
finalB=sumB
bessk.t=finalB-finalA
numadd=numadd+1
nummult=nummult+4
end function bessk.t
subroutine get.sumA() ! returns sumA to program
type(taylor)::sumA
sumA=sumA
```

```
! return sumA
end subroutine get.sumA

function fact(n)
integer,intent(in)::n
integer::fact
integer::i
fact=1
do i=1,n
fact=fact*i
nummult=nummult+1
end do
end function fact

function fbit(t1) ! parts of module for newbess
type(taylor),intent(in)::t1 ! for A & S formula
type(taylor)::fbit
type(taylor),dimension(taylor.degree)::next
type(taylor)::qRd,b,c
real::a
integer::i
call init.taylor(fbit)
call init.taylor(next(1))
qRd=t1*0.25
next(1)=shright(qRd)
next(1)=shright(next(1))
nummult=nummult+1
do i=2,((taylor.degree+1)/2)
a=1.0/(i*i)
b=shright(next(i-1))
c=shright(b)
next(i)=mult(c,qRd)
next(i)=next(i)*a
nummult=nummult+3
end do
do i=1,taylor.degree
fbit=fbit+next(i)
numadd=numadd+1
end do
end function fbit

function bigb(Rd)
type(taylor),intent(in)::Rd
type(taylor)::bigb
```

```
type(taylor)::first,second
integer::i
call init.taylor(first)
call init.taylor(bigb)
do i=1,(Taylor.degree-1)/2
first%series((2*i)+1)=first%series((2*i)-1)+(1.0/i)
numadd=numadd+3
nummult=nummult+4
end do
second=fbit(Rd)
do i=1,taylor.degree
bigb%series(i)=first%series(i)*second%series(i)
nummult=nummult+1
end do
end function bigb

function bigi(Rd)
type(taylor),intent(in)::Rd
type(taylor)::bigi
bigi=fbit(Rd)
bigi%series(1)=bigi%series(1)+1.0
numadd=numadd+1
end function bigi

end module taylormod
```