A TIME-DEPENDENT GREEN ELEMENT FORMULATION FOR SOLUTION OF POTENTIAL FLOW PROBLEMS IN 3 DIMENSIONAL DOMAINS

Edwin Nyirenda

A thesis submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfilment of the requirements for the degree of Doctor of Philosophy.

Johannesburg, 2010

DECLARATION

I declare that this thesis is my own unaided work. It is being submitted for the degree of Doctor of Philosophy in Engineering at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

____
(Signature of candidate)

__________________ day of ____________________ year ___________________
day *month* *year*

In memory of my late uncle and aunt, Mr. Netofati Gelesom Ngoma and Mrs. Nelly Jere Ngoma, for supporting me to pursue education when I was at a vulnerable age, my parents Mr. Gilbert Donald Nyirenda and Mrs. Titsauke Ngoma Nyirenda for having been there for me all my life, and to my wife and best friend, Florence Chiseng'antambu Nyirenda.

ABSTRACT

In this work we develop a generalised methodology for the solution of the time-dependent second order parabolic differential equation of potential flow in heterogeneous media using the Green element method. Parabolic differential equations are one class of differential equations, the others being elliptic partial differential equations and hyperbolic differential equations. Since elliptic differential equations generally arise from a diffusion process that has reached equilibrium, they can also be solved using the methodology developed, and represent a simplification because of the steady state situation.

Potential flow problems are of great interest in many engineering applications such as flow in aquifers, heat transfer processes, electro-magnetic field problems, etc. Traditionally, the finite difference method and the finite element method have proved to be powerful techniques to solve such potential flow problems, but each has limitations and challenges which have led to continued research in numerical methods. The finite difference method is more applicable to domains with regular boundary, and the finite element method, though extremely versatile, exhibits unacceptable inaccuracies with coarse meshes, thus requiring fine meshes with the associated high computation costs.

In view of some of the limitations with these earlier methods, several numerical schemes are now being developed as viable alternatives to these conventional methods. Among such methods are the boundary element method, the finite volume method, and the analytic element method. The boundary element method has been particularly promising because of its domain-reduction feature and the second order accuracy that can generally be achieved. The domain-reduction feature of the boundary element method, though achieved for restricted class of problems, lends it to efficient grid generation algorithm, while its second-order accuracy ensures reliability and consistency of the numerical solutions.

The boundary element method in its original formulation is unable to deal with heterogeneities in the domain. For physical problems, especially in groundwater flow, heterogeneities and anisotropy are a natural and frequent occurrence, and this has fuelled research into boundary element techniques that are capable of accommodating these features.

The Green element method is one technique which is based on the boundary element theory and which has been proven to be very effective in handling heterogeneities and anisotropy in 1D and 2D domains. However, development of techniques to implement the Green element method in 3D domains has remained largely unexplored. This work represents an effort in this direction.

We have investigated the adoption of the general tetrahedral and hexahedra elements for use with the Green element method, and found that the large number of degrees of freedom generated precludes retention of the internal normal direction as in 1D and 2D formulations. Furthermore, some of the complicated surface and domain integrations with these elements can only be addressed with quadrature methods. The compatibility issues that arise between element faces, which present considerable challenges to multi-domain boundary element techniques, are innovatively addressed in the computer code that has been developed in this work.

The Green element method is implemented for steady and time-dependent problems using regular hexahedra elements, and the results show that the performance is slightly better than the results obtained using FEMWATER. FEMWATER is an established finite element method software. No attempt is made to compare the computation efficiencies of the 3D GEM code and FEMWATER because the two codes were not developed on a common platform.

ACKNOWLEDGEMENTS

First and foremost I offer my sincere gratitude to my supervisor, Professor A.E. Taigbenu, who has supported me throughout the period of my thesis with his patience and knowledge, whilst allowing me the room to work in my own way. I attribute the success of my PhD degree to his encouragement and effort. One simply could not wish for a better or friendlier supervisor.

Not only did Professor A.E. Taigbenu support me in the academic work, but he also ensured that I had the financial stability to concentrate on my research activity. For the first two years my tuition fees and living expenses were met in part through the University of the Witwatersrand Post Graduate merit award and in part through the Department of Civil and Environmental Engineering research funds, both awarded to me with the support of Prof Taigbenu. For the final two years the expenses above were met through the National Research Foundation Grant-holder linked research funds, and the grant holder was Prof. Taigbenu. I am really grateful for these sources of funding.

I would also like to thank Dr. J Ndiritu for many encouraging comments during the course of my work on the research project. These comments helped in many ways to strengthen me in times of doubt.

In my daily work I have been blessed with a friendly and cheerful group of fellow students. Many thanks go to Mr. Zacharia Katambara and Mr. Paulo Kagoda who have shared many programming tips and general knowledge on a daily basis. Their support helped in many ways to assure me in times of stress. Many thanks also go to Jean Marc Mwenge Kahinda for interpreting the relevant sections of a journal published in French to me. His input is greatly appreciated.

I also want to thank the Department of Civil and Environmental Engineering at the University of Zambia, and the University of Zambia Staff Development Office for giving me permission to commence this thesis in the first instance and to do the necessary research work.

Finally, I would like to give my special thanks to my wife Florence and our children whose patience and love enabled me to complete this research work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

NOMENCLATURE

u*	**Fundamental solution**
q*	**Derivative of the fundamental solution**
u	**Potential, a scalar field**
q	**Potential gradient,** $q_x = \frac{\partial u}{\partial x}, q_y = \frac{\partial u}{\partial y}, q_z = \frac{\partial u}{\partial z}, q_n = \frac{\partial u}{\partial n}$
n	**Normal to surface**
c	**Distance between collocation point and normal to the surface on which integration is being done.**
k	**Coefficient of the potential at the collocation node**
AEM	**Analytic Element Method**
FEM	**Finite Element Method**
BEM	**Boundary Element Method**
GEM	**Green Element Method**
FVM	**Finite Volume Method**
PDE	**Partial Differential Equation**
x, y, z	**Global coordinate system (Coordinate System 1)**
η_1, η_2, η_3	**Local coordinate system for surface on which integration is being done with origin at collocation point (Coordinate System 2)**
ξ_1, ξ_2, ξ_3	**Local coordinate system for a standard isoparametric unit surface element (Coordinate System 3)**
ϕ_i	**Interpolation functions** ϕ_1**,** ϕ_2**,** ϕ_3**, and** ϕ_4
J	**Jacobian of** η_1, η_2, η_3 *with respect to* ξ_1, ξ_2, ξ_3
Ω	**Volume integration space defined by coordinate system 1**

Γ	**Surface integration space defined by coordinate system 2**
Ψ	**Surface integration space defined by coordinate system 3**
$w_1(\xi_1,\xi_2)$, $w_2(\xi_1,\xi_2)$	**Functions with respect to ξ_1, ξ_2**
H, G, W	**Nodal potential, Flux, and Forcing term multiplier matrices, respectively.**

1. INTRODUCTION

In this chapter an introduction to the problem of the solution of potential flow problems is given. The discussion starts with a brief background to the problem which necessitated the study of partial differential equations before defining the problem that is the subject of the current study. Finally, the objectives of study and a brief discussion on the format of the report are given.

1.1 Background

The study of partial differential equations began in the 18th Century with the work of pioneers like Laplace, d'Alembert, Euler and Lagrange as a principal tool in the study of mechanics of continua and physical sciences. During the 19th Century, the subject was picked up by mathematicians particularly with the work of Riemann, as reported by Brezis and Browder (1998). Partial differential equations generally involve two or more independent variables, which in practice are often space and time variables, a number of dependent variables that describe the physical phenomena, and parameters that reflect the constitutive relationships. Due to the fact that more than one independent variable is present, the "derivatives" that occur are partial derivatives. Such equations are widespread in science and engineering, and they model physical phenomena. They also arise frequently in the form of systems of equations whose coupling reflect the interdependence among the dependent variables.

Seemingly distinct physical phenomena may have identical mathematical formulations, and thus be governed by the same underlying principle. This fact was first stated in the late 19th century (Poincaré, 1890). Poincaré emphasized that a wide variety of physically significant problems arising in very different areas (such as electricity, hydrodynamics, heat transfer, magnetism, optics, elasticity, etc) have a family resemblance that lends to treatment by common methods.

The most satisfactory solution of a field problem is an exact mathematical one. The appropriate method of solution has to be determined for each particular problem. For example, for steady potential flow in two dimensions, the most powerful analytical method appears to be the method of complex variables, which involves such concepts as the velocity hodograph and conformal mapping (Bear and Verrijt, 1988). A hodograph is a diagram that gives a vectorial visual representation of the movement of a body or a fluid. It is the locus of one end of a variable vector, with the other end fixed, and has found some uses in the analytical solution of partial differential equations. An example of its applications can be found in the work of McCarthy (1994). A powerful method to deal with nonsteady problems is the Laplace transform technique. However, in many practical cases an analytical solution cannot be obtained due to the complexity of the loading and/or the physical geometry, and we must resort to numerical approximate solutions. To use analytical methods, the geometry must be regular, e.g. circular, rectangular, or infinitely extensive. The properties of the medium must be homogeneous, or at least homogeneous in sub-regions. Knochenmus and Robinson (1996) have pointed out that because of the idealisations necessary in order to use analytical methods, flurries of developments of numerical methods now abound in the literature primarily because of their ability to produce solutions to more realistic and practical problems. Analytical solutions are however still useful in checking solutions obtained from numerical methods.

Traditionally, the finite difference method (FDM) and the finite element method (FEM) have proved to be powerful numerical techniques for solving potential flow problems. The two methods do basically the same thing, i.e. approximate the set of partial differential equations to be solved by a set of algebraic equations. The FDM has had and continues to have strong appeal in numerical circles because of its ease of understanding, relying on classical calculus in approximating differential operations, which are approximated by difference formulae constructed from values of the dependent variable at a number of predefined points. This makes it most appropriate to domains that are regular. The FEM, on the other hand, approaches the solution in a more indirect way.

Firstly, it is not the differential operator that is approximated by difference formulae constructed from values of the dependent variable at a number of predefined points, but the variation of the independent variables (i.e. the given function is approximated locally over each element by continuous functions which are uniquely defined in terms of the values of the function and possibly its derivatives at the nodes on each element). Secondly, the algebraic equations are formed as the result of either the local minimisation of some appropriate energy function natural to the physical system or the piecewise application of a weighted residual method to the differential equations.

The FDM and the FEM each has limitations and challenges which have led to continued research in numerical methods. In view of this, several numerical schemes are now being developed to circumvent the limitations and challenges of the FDM and the FEM. Among such methods are the boundary element method, the finite volume method, and the analytic element method. Although these new methods do address the problems of the FDM and the FEM, each does come with its set of limitations and challenges. The boundary element method has been particularly promising because second order accuracy is maintained in its formulation, which ensures reliability and consistency of the numerical solutions.

The boundary element method (BEM) in its original formulation is unable to deal with heterogeneities in the domain. Onyejekwe (2006) has summarized the various efforts that have been made to adapt the method to physical problems. He notes that the paucity of literature on BEM application to transport problems involving media heterogeneity underscores the fact that a lot of work still needs to be done in this area. For physical problems, especially in groundwater flow, heterogeneities and anisotropy are a natural and frequent occurrence, and this has been a motivation for a lot of research into techniques to adapt the boundary element method to deal with these situations.

To utilise the second order accuracy of the boundary element method for non-linear heterogeneous systems, Taigbenu proposed the Green element method (GEM) in 1990 (Taigbenu, 1990) which he defined as an element by element implementation of the boundary element theory over the meshes of the finite element method. Taigbenu (1995) presented a theoretical discussion of Green element method and demonstrated that GEM required 15% to 45% less computational time than the standard boundary element method. Since then, some authors have picked up on the method as a limiting case of the multi-domain technique in boundary element analysis in which the size of the domains is taken to the limiting case of the elements used in finite element methods (Gao and Davies, 2000; Ramšak and Škerget, 2007; Ramšak and Škerget, 2009).

Numerical formulations that are based on the Green element method (GEM) have been used to solve various problems of practical engineering concern, and there are several papers which have been published which highlight the applications in the various fields. Some examples of the publications in these various fields are as follows.

- Heat transfer

  Simulation of coupled non-linear electromagnetic heating with the Green element method (Taigbenu, 2006). In this paper, the author demonstrated how the flux-based Green element formulation is used to solve the non-linear coupled differential equations that govern the problem of heat transfer in food materials that are electrically heated. This is a problem of major interest to the food and related industries interested in the heating of food substances by electrical currents.

- Stream-unconfined aquifer flows

  A time-dependent Green's function based model for stream-unconfined aquifer flows (Taigbenu, 2003). In this paper the author used the Green element method to solve the non-linear stream-unconfined aquifer flow problem, which demonstrates the computational flexibility that is achieved with a Green element sense of implementing the boundary integral theory.

- Petroleum Engineering

  Dual reciprocity boundary element method and the Green element method were used by Archer et al. (1999) to solve some petroleum reservoir engineering problems. Petroleum reservoir engineering requires solutions to the transient diffusion and convection-diffusion equations. Traditionally these calculations are carried out using finite difference methods. In this paper the authors applied the Dual Reciprocity Boundary element method and the Green element method to flow in heterogeneous media and to the convection-diffusion equation. Numerical experiments showed the Green element method to be more accurate and more stable than the Dual Reciprocity Boundary element method at high Peclet number. Both methods allow reservoir heterogeneity to be treated more efficiently than with the perturbation-based methods.

- Chemical Engineering

  Application of the Green Element Method to Chemical Engineering Problems (Abashar, 2004). In this work the Green element method (GEM) is implemented on chemical engineering problems. The author reports that the method overcomes some of the limitations of classical boundary element approach and uses the finite element methodology to achieve optimum inter nodal connectivity. The global coefficient matrix is banded and numerical difficulties from a densely populated matrix are eliminated. Two numerical examples are used to demonstrate the capabilities of the method. The results are compared with orthogonal collocation method, finite element method and experimental data. It was shown that the Green element method is very reliable and efficient.

The problem at hand is that there is need to extend the Green element formulation to three dimensions, and to provide a tool for solving three dimensional potential flow problems. An understanding of differential equations is crucial to implementing such a tool, and a brief discussion on differential equations is therefore herein provided.

1.2 Problem Definition and Importance of Study

The Green Element Method (GEM) is a new, powerful, and promising technique for solving boundary value problems (Pecher et al., 2001). It is derived from the Boundary Element Method (BEM) and applied over the meshes of the Finite Element Method (FEM), the Green element method combines the second order accuracy of the BEM with the efficiency and versatility of the FEM. Solutions of 1-D and 2D potential flow problems using the Green element method have been successfully implemented, and the results show that the limitations of the BEM method in solving non-linear potential flow problems that exhibit medium heterogeneities are overcome with the GEM, and that significant computational gains can be achieved by using GEM instead of the standard BEM (Taigbenu, 1995). The current research project is aimed at implementing the GEM numerical solution strategy for time-dependent three-dimensional potential flow problems and investigating if the observed powerful advantages of the GEM do extend to 3-Dimensional domains.

1.3 Objectives of Study

The objective of the current research project is to develop a numerical formulation based on the Green element method for time-dependent potential flow problems that extends the work done to date for the one and two spatial dimensions.

We have maintained the name “Green element method” for the technique of applying the boundary element method at the scale of the elements of the finite element method. This is in accordance with the name originally given to the method when it was first presented (Taigbenu, 1995).

Ramsak and Skerget have adopted the name “multidomain BEM” for the same method as the Green element method (Ramšak and Škerget, 2009). They exposed the paucity of research papers on the subject when they stated that there are various 2D BEM applications using the multidomain method, but that only one 3D application is found in open literature. They refer to the work of Gao and Davies (2000) who used more than two sub-domains in 3D for the first time. Gao and Davis actually used four sub-domains, and in their later work (Gao, 2004) the multidomain approach was abandoned. (Gray and Kaplan, 2001) also make the same observation about the paucity of research work in 3D applications of singular integration when they state that while the analysis of two-dimensional singular integrals is relatively easy, the three dimensional integrals are less straight forward, and that this explains the dearth in sources for such techniques. Furthermore, the inter-element connectivity relations in 3D are of a much higher order of complexity than in 2D (Peratta and Popov, 2006; Ramšak and Škerget, 2009).

The hypothesis is raised that it is possible to develop a three-dimensional formulation for the Green element method. In the event that the three dimensional formulation is developed, it is also the objective of the current project to test if the computational gains that have been reported for the one-dimensional and two-dimensional cases do extend to three dimensions. Unknown to the author of this thesis, some work was actually being conducted on the same subject, albeit under the heading “multidomain BEM” instead of “Green element method” by Ramsak and Skerget who published a paper on the subject in 2009 (Ramšak and Škerget, 2009). In their work they have used forty-eight Gauss integration points in each integration direction (i.e. x, y, and z directions). In the course of the research for this thesis over a period of four years from 2006 to 2009, we have investigated both Gauss quadrature integration points and have also investigated the use of analytic integration using MATHEMATICA software.

1.4 Format of the Report

Chapter 2 of this report presents the literature reviewed both prior to the commencement of the development of the numerical formulation based on the Green element method for time-dependent potential flow problems in 3-dimensions and during the development stages. The following areas of concern were reviewed:

- Ordinary and partial differential equations in general
- Techniques available for the solution of partial differential equations
- A more focussed review of the boundary element method and the formulations for the 3-dimensional Laplace equation, together with the techniques for handling domain integrals arising when inhomogeneity is introduced to the equation.

In Chapter 3 we delve into the Green element solution of the 3D Laplace equation. The solution developed in that chapter served as the basis for further enhancements of the Green element formulation which incorporates the forcing term (Poisson equation) and the temporal derivative term (diffusion or heat conduction equation). These numerical enhancements of incorporating the forcing term and the temporal derivative term are presented in Chapter 4.

Chapter 5 presents numerical examples with which the performance of the developed Green element method is compared against analytical solutions and those of a finite element commercial package, FEMWATER. The main findings from the research are discussed in Chapter 6.

The computer code that has been developed to implement the 3D GEM for the time-dependent potential problem is written in Visual Basic for Applications and runs under Microsoft Excel. The MS Excel platform enhanced visualization of the output at every stage of the development of the code.

In addition to the main programming being carried out in Visual Basic, various utilities were developed to aid with the analysis. The discretizing elements were built with the GID software. A need arose for a script to output the grid built with GID into the main Visual Basic code. Output from GID is accomplished by creating a set of customization text files called a “Problem Type”. Appendix B is the script listing of the main text file of this Problem Type, and it has the specific purpose of outputting information from GID by generating a file that can be processed into a form suitable for the developed Green element code. The processing into a form suitable for the Green element code is done by the FORTRAN script listed in Appendix C.

2. LITERATURE REVIEW

In view of the fact that the study of partial differential equations has been on-going since the 18th Century, a large collection of information on the subject of the solution of such equations is available in the literature. The literature search reveals that whereas analytical solutions are well developed for idealised geometries and boundary conditions which can only be applicable to few practical problems, numerical schemes are more suited to solving partial differential equations that govern practically based problems. A discussion on the available numerical schemes in literature is provided, and special attention accorded to the Boundary element method since this is the basis on which the Green element method (the numerical scheme for the research work) is based.

2.1 Introduction to Ordinary and Partial Differential Equations

2.1.1 Classification of Differential Equations

- ***Ordinary and Partial Differential Equations***

Differential equations can be ordinary or partial. An *ordinary* differential equation (ODE) contains differentials with respect to only one variable while a *partial* differential equation (PDE) contains differentials with respect to several independent variables. In mathematical terms, a partial differential equation (PDE) is any equation involving a function of more than one independent variable and at least one partial derivative of that function. The order of a PDE is the order of the highest derivative that appears in the PDE. The principal part of a PDE is the collection of terms in the PDE containing derivatives of order equal to the order of the PDE (Hemker, 2004).

Ordinary differential equations that have time as the independent variable are also called '***initial value problems'*** because the information about the behaviour of the system is known at the initial time t_0 and it is required to find the solution at times subsequent to t_0. For a partial differential equation in more than one spatial dimension, the problem is to solve for the dependent variable given some boundary conditions. The problem is therefore a '***boundary value problem'***. A boundary value problem may also be time dependent, and to solve such a problem we need the initial conditions specified over the entire spatial domain in addition to the boundary conditions. This problem is concerned with a solution technique to partial differential equations of the latter type, 'boundary value problems that incorporate time as an independent variable'.

- ***Linear and non-linear Differential Equations***

A PDE in a dependent variable u is classified as linear if all of the terms involving u and any of its derivatives can be expressed as a linear combination of u in which the coefficients of the u-terms are independent of u i.e. the unknown function and its derivatives appear to the power of unity (products are not allowed). In a linear PDE, the coefficients can depend at most on the independent variables. If the terms involving the dependent variables (u or any of its derivatives) involve products of the variables then the differential equation is non-linear.

- ***Classes of Differential Equations***

In addition to the distinction between linear and nonlinear PDEs, it is important for the computational scientist to know that there are different classes of PDEs. Just as different solution techniques are called for in the linear versus the nonlinear case, different numerical methods are required for the different classes of PDEs, whether the PDE is linear or nonlinear. The need for this specialization in numerical approach is rooted in the physics from which the different classes of PDEs arise. By analogy with the conic sections (*ellipse, parabola* and *hyperbola*) partial differential equations have been classified as elliptic, parabolic and hyperbolic. Just as an ellipse is a smooth, rounded object, solutions to elliptic equations tend to be quite smooth. Elliptic equations generally arise from a physical problem that involves a diffusion process that has reached equilibrium, for example a steady state heat transfer over a solid object. The hyperbola is the disconnected conic section. By analogy, hyperbolic equations are able to support solutions with discontinuities, for example a shock wave problem. Hyperbolic PDEs usually arise in connection with mechanical oscillators, such as a vibrating string, or in convection driven transport problems. Mathematically, parabolic PDEs serve as a transition from the hyperbolic PDEs to the elliptic PDEs. Physically, parabolic PDEs tend to arise in time dependent diffusion problems, such as the transient transport of heat in accordance with Fourier's law of heat conduction.

2.1.2 Partial Differential Equation for the Research Problem

It is intended in this research to extend the use of the Green element method to the time-dependent second order parabolic differential equation in homogeneous media. The differential equation that describes potential flow problems in three dimensional domains has the mathematical form given by equation (2.1) below.

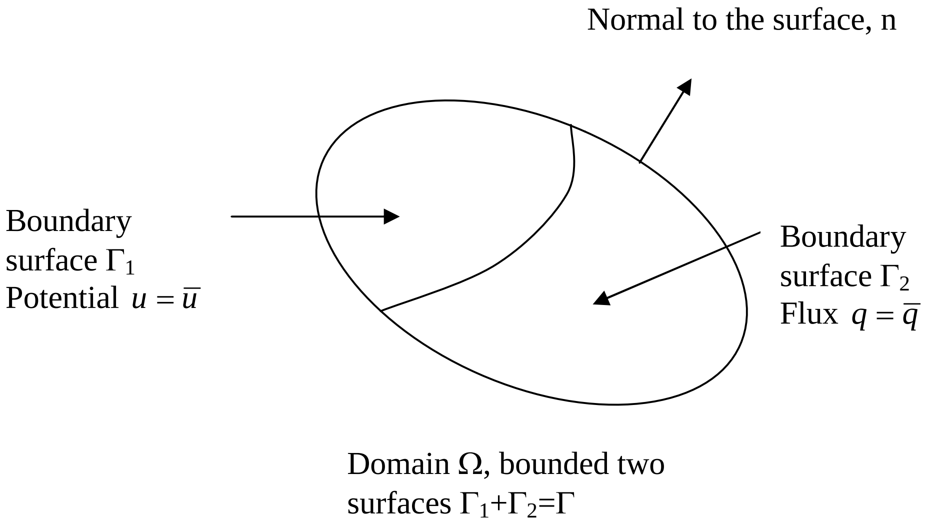
$$K\nabla^2 u = S\frac{\partial u}{\partial t} + f(x,y,z,t) \quad \text{in } \Omega \tag{2.1}$$

Where $\nabla = i\frac{\partial}{\partial x} + j\frac{\partial}{\partial y} + k\frac{\partial}{\partial z}$, $S\frac{\partial u}{\partial t}$ is the temporal derivative term, and $f(x,y,z,t)$ is the forcing term. Ω is the domain in which the equation applies and is bounded by Γ. The boundary Γ may be divided into several section surfaces, for example Γ_1 and Γ_2. See Figure 2.1.

Equation (2.1) is a linear partial differential equation if the main domain parameters K and S are not functions of u, and the interpretation of the quantities K and S will depend on the engineering application to which the equation is applied. There are several areas of study in which the equation is applicable, for example:

- Flow through porous media
- Heat conduction
- Ideal fluid flow
- Diffusion
- Circulation of fluid flow
- Electrostatic field
- Shaft Torsion
- Pressurized membranes

In flow through porous media (like groundwater flow), for example, K is the hydraulic conductivity in the x, y, and z directions, u is the hydraulic head, and S is the storativity for confined aquifers and specific yield for unconfined aquifers (Bear, 1972). For heat conduction problems, K is thermal conductivity, u is the temperature, and S is the heat capacity (Drebushchak, 2009; Tsilingiris, 2006).



The dashes on u or q indicate that those values are known

Figure 2.1. Three dimensional domain under consideration for basic definitions.

2.1.3 Boundary Conditions

Differential equations usually apply to a region or domain which is either finite or infinite, but the computational domain will have some boundaries (points, lines, or surfaces). At these boundaries, some conditions must be imposed on the primary variable and/or the flux. Without these boundary conditions, there are infinitely many distributions of the primary variable that can satisfy the differential equation. The boundary conditions are, in general of three types, and it is important to appreciate their physical implications. A discussion on the three types follows:

- ***Boundary condition of the first kind – distribution of primary variable at boundary is known***.

  Mathematically, this is formulated as

$$u = u_1(x, y, z, t) \text{ on } \Gamma_1 \tag{2.2}$$

Where Γ_1 is part of the boundary of the computational domain.

Such conditions are called *Dirichlet*, or *essential conditions*, and the physical interpretation is that at such boundaries there is an absorbing screen such that as $t \to \infty$ the value of the variable u within the domain (and due to such conditions alone) will tend toward the value at the boundary (Gil et al., 2002).

- ***Boundary condition of the second kind – distribution of flux at boundary is known***.

Mathematically, this is formulated as

$$-K\nabla u \bullet \underline{n} = q(x, y, z, t) \text{ on } \Gamma_2 \tag{2.3}$$

Where $\underline{n}$ is the outward pointing normal vector on the surface of the boundary Γ_2. Such conditions are called *Neumann*, or *flux conditions*. They specify the distribution of the normal flux on a part of the boundary.

- ***Boundary condition of the third kind – linear combination of primary variable and flux at a boundary is known***.

Mathematically, this is are formulated as

$$\lfloor au(x, y, z, t) - bK\nabla u \bullet \underline{n} \rfloor = v(x, y, z, t) \tag{2.4}$$

Here a and b are known coefficients and v(x,y,z,t) is a known value of the linear combination of the primary variable and the flux. If a=0 we get the Neumann conditions, while if b=0 we get the Dirichlet conditions.

2.2 Solution Techniques for Partial Differential Equations

If a solution can be developed for a partial differential governing equation, preference is to develop an analytical solution. But unfortunately, for many problems, it is extremely *difficult*, if not ***impossible***, to get an analytic solution. Reasons for the limitation of analytic solution techniques range from irregular geometry, through heterogeneity of the domain to the nonlinearity of the differential equation. Sometimes, the analytic solution may be too *complicated* and one would prefer to use an approximation. Numerical methods (methods in which approximate solutions are derived and coded on computers) have proved to be extremely useful in overcoming the limitations of analytical solution techniques and in addressing practically realistic problems that are on irregular geometries, heterogeneous, and nonlinear.

In solving partial differential equations using numerical techniques, the primary challenge is to create a set of relationships which approximates the equation of interest so that the solution is consistently reliable with respect to the element size and is numerically stable, implying that errors in the input data and intermediate calculations do not accumulate and cause the resulting output to be meaningless. There are many ways of doing this, all with advantages and disadvantages. Kraemer (2003) has made a classification by numerical method of groundwater flow publications to give an indication of the scale of adoption of the various methods in groundwater flow modelling. The distribution has been presented in form of a pie chart, with the following percentages: finite elements (55%), finite differences (37%), boundary elements (5%), and analytic elements (3%). This information was obtained from citation information obtained from SCI Journal Citation Reports and the Web of Scientific Information (ISI) for the period 1980 to 2003. This indicates that the finite element method enjoys the largest share of application in groundwater flow modeling. While this is the case, knowledge and innovation are enhanced by continued research and development of new computational techniques.

The thrust of this research project is to extend the work that has been done in an existing numerical solution technique, the Green element method (GEM), to the 3-dimensional case. In view of the fact that the Green element method is based on the finite element method (FEM) and the Boundary element method (BEM), brief discussions on the FEM, BEM, and the GEM are given in the sections below. To complete the introduction on the various numerical techniques, brief discussions on the finite difference method, the finite volume method, and the analytic element method are also presented.

2.2.1 The Finite Difference Method

Finite difference methods approximate the solutions to differential equations by replacing derivative expressions with approximate equivalent difference quotients. They represent the oldest numerical solution techniques that elicit classical interpretation of the derivative in calculus. Because of their ease of understanding, finite difference schemes continue to enjoy wide acceptance and use in many engineering applications. As reported by Kraemer (2003), the 37% of all publications on groundwater flow modeling in the period 1980 to 2003 were with the finite difference method.

Various commercial software have been developed using the finite difference method. One of the most popular ones in groundwater applications is MODFLOW which does modeling in 3D. Abdel-Fattah et al (2007), with the aid of MODFLOW, utilized particle tracking techniques to evaluate transport of river water through an alluvial aquifer in a bank infiltration testing site in El Paso, Texas, USA, while Sun and Zheng (1999) have used a MODLOW based dynamic optimization tool to develop policies to achieve various groundwater management objectives. MODFLOW is undergoing continuous development. For example, the recent advances in graphical user interfaces and geographic information systems have resulted in increasingly complex MODFLOW files. Complex source/sink data, including transient stages and pumping rates can result in a single MODFLOW simulation taking as much as one gigabyte of space on disk using the standard MODFLOW text file format. Jones et al (2005) have presented a new binary format for MODFLOW arrays and MODFLOW stress packages. The format is based on the HDF5 library developed by the National Center for Supercomputing Applications at the University of Illinois at Urbana-Champaign. MODFLOW data can be saved and retrieved from this format using an API built on top of the HDF5 object code. This results in a set of MODFLOW input files where the bulkier parts of the files are stored in a compressed, platform independent binary format. MODFLOW files written in this format can be read in a few seconds and the resulting file sizes are typically less than 1-2% of the original text file format. Wang et al (2008) have used MODFLOW on a geographic information system platform for groundwater evaluation in North China Plain, China. The simulation included water budgeting to ascertain groundwater availability for industrial, agricultural, and domestic usage.

2.2.2 The Finite Element Method

Finite elements were first used in the 1950s in aircraft design to study stresses in the complex airframe structures (Huebner et al., 2001; Livesley, 1983)$^{.}$ However, in recent years the approach has increasingly been used in other branches of engineering. This widening of the area of application has revealed that the finite element method is a general procedure for obtaining approximate solutions to partial differential equations.

The finite element method is one in which the subdivision of the region into subdomains, finite elements, or cells is an essential part of the procedure, with some functional representation of the solution being adopted over the elements so that the parameters of the representation become the unknowns of problem. The Finite element method is not based directly on the partial differential equation, but rather on a weak (or variational) formulation of the problem. The weakness is achieved by a process of integration by parts that reduces (or ‘weakens’) the order of the continuity required for the dependent variable (Brebbia and Dominguez, 1992).

For potential flow problems, it is possible to solve for the dependent variable using the finite element method, while the flux must be computed by numerically differentiating the primary variable (Taigbenu and Onyejekwe, 1995). This was the original formulation of the finite element method, and this weakness was one of the motivations for the development of the Boundary element method.

It must also be pointed out that advanced finite element techniques, collected under the name mixed finite element methods, have had enormous impact in computational mechanics. The term mixed method was first used in the 1960's to describe finite element methods in which both stress and displacement fields are approximated as primary variable (Bergamaschi et al., 1994). Numerous applications are now available of the mixed finite element methods. See for example (Arnold, 1990; Bergamaschi et al., 1994; Maryska et al., 1995; Park, 2005; Van Criekingen and Beauwens, 2007).

2.2.3 The Boundary Element Method

Since the publication of the first book on Boundary Elements (Brebbia, 1978) many such works have appeared in the literature, some dealing with potential flow, others with elastostatics, and many other areas of application. Boundary elements are a powerful alternative to finite elements, particularly in cases where higher accuracy is required. The most important feature of boundary elements is that they only require discretization of the surface rather than the volume. Hence boundary element codes are easier to use with existing solid modellers and mesh generators. This advantage is particularly important for design purposes as the process usually involves a series of modifications (Brebbia and Dominguez, 1992; Hsieh et al., 1992; Maduramuthu and Fenner, 2004). Moreover, as has been demonstrated by some authors, the conventional BEM can be enhanced to the general BEM in which iteration is not necessary for nonlinear problems (Liao, 1998; Zhao and Liao, 2003). This verifies the validity and great potential of the general boundary element method for highly non-linear problems, which may greatly enlarge application regions of the boundary element method in science and engineering.

Green's Functions: Basic concepts

Green's functions play an important role in the solution of partial differential equations, and are a key component in the development of boundary integral equations. Consider a linear differential equation written in the general form

$$\mathcal{L}(x)u(x)=f(x) \tag{2.5}$$

Where $\mathcal{L}(x)$ is a linear, self-adjoint differential operator and $f(x)$ is a known non-homogeneous term (A self-adjoint operator is one that is its own adjoint. If one thinks of operators on a Hilbert space as "generalized complex numbers", then the adjoint of an operator plays the role of the complex conjugate of a complex number). Operationally, we can write the solution to the above equation as

$$u(x)= \mathcal{L}^{-1}(x)f(x) \tag{2.6}$$

where $\mathcal{L}^{-1}$is the inverse of the differential operator $\mathcal{L}$. Since $\mathcal{L}$ is a differential operator, it is reasonable to expect its inverse to be an integral operator. We expect the usual properties of inverses to hold,

$$\mathcal{L}^{-1}\mathcal{L}=I \tag{2.7}$$

where *I* is the identity operator. More specifically, we define the inverse operator as

$$\mathscr{L}^{-1}\, f = \int G(x,s)\, f(s)\, ds \qquad \textbf{(2.8)}$$

where the kernel G(*x*,s) is the Green's function associated with the differential operator $\mathscr{L}$. G(*x*,s) is a two-point function which depends on x and s. To complete the idea of the inverse operator, we introduce the Dirac delta function $\Delta(x-s)$ as the identity operator *I*.

The solution to Equation (2.5) can then be written directly in terms of the Green's function as

$$u(x) = \int_{-\infty}^{\infty} G(x,s)\, f(s)\, ds \qquad \textbf{(2.9)}$$

It is to be noted as of great significance that by making use of the Green's function for the differential operator $\mathscr{L}$, an expression for the dependent variable has been obtained without solving the differential equation (2.5). This fact is utilised in the development of the Boundary element method and is the basis of all work on Green element method that has been conducted for one-dimensional and two-dimensional domains.

2.2.4 The Finite Volume Method

The finite volume method is a method for representing and evaluating partial differential equations in the form of algebraic equations. Similar to the finite difference method, values are calculated at discrete positions on a meshed geometry. "Finite volume" refers to the small volume surrounding each node point on a mesh. In the finite volume method, volume integrals in a partial differential equation that contain a divergence term are converted to surface integrals, using the divergence theorem. These terms are then evaluated as fluxes at the surfaces of each finite volume. Because the flux entering a given volume is identical to that leaving the adjacent volume, these methods are conservative. Another advantage of the finite volume method is that it is easily formulated to allow for unstructured meshes. The method is used in many computational fluid dynamics packages.

The finite volume method has been used to solve some problems of a practical nature (Onate et al., 1994; Rühaak et al., 2008; Xia et al., 2007). These authors point out that finite volume methods may be considered as a particular case of finite element methods with non Galerkin weighting. Unlike the finite element approach, in which the relevant conservation principle and/or equilibrium equation of forces are only satisfied in a global sense, the finite volume procedure is conservative from the whole-domain scale down to the cell or control volume level.

2.2.5 The Analytic Element Method

The analytic element method (AEM) is a numerical method used for the solution of partial differential equations and was initially developed by O.D.L. Strack at the University of Minnesota (Strack, 1989). It is similar in nature to the Boundary Element Method (BEM), as it does not rely upon discretization of volumes or areas in the modeled system; only internal and external boundaries are discretized (internally, boundaries exist at the interaction surfaces of various zones). One of the primary distinctions between AEM and BEMs is that the boundary integrals are calculated analytically. The analytic element method is most often applied to problems of groundwater flow governed by the Poisson equation, though it is applicable to a variety of linear partial differential equations, including the Helmholtz, and biharmonic equations.

The basic premise of the analytic element method is that, for linear differential equations, elementary solutions may be superimposed to obtain more complex solutions. A suite of 2D and 3D analytic solutions ("elements") are available for different governing equations. These elements typically correspond to a discontinuity in the dependent variable or its gradient along a geometric boundary (e.g., point, line, ellipse, circle, sphere, etc.). This discontinuity has a specific functional form (usually a polynomial in 2D) and may be manipulated to satisfy Dirichlet, Neumann, or Robin (mixed) boundary conditions. Each elementary analytic solution is of infinite extent in space and/or time. In addition, each analytic solution contains degrees of freedom (coefficients) that may be calculated to meet prescribed boundary conditions along the element's border. To obtain a global solution (i.e., the correct element coefficients), a system of equations is solved such that the boundary conditions are satisfied along all of the elements (using collocation, least-squares minimization, or a similar approach). Notably, the global solution provides a spatially continuous description of the dependent variable everywhere in the infinite domain, and the governing equation is satisfied everywhere exactly except along the border of the element, where the governing equation is not strictly applicable due to the discontinuity.

Though powerful and easy to use, applications of the analytic element method are not as widespread as finite-difference or finite-element models due in part to its relative youth. Although reviews that focus primarily on the mathematical development of the method have appeared in the literature, a systematic review of applications of the method is not available. Hunt (2006) has given an overview of the general types of applications of analytic elements in groundwater modelling. While not fully encompassing, the applications described cover areas where the method has been historically applied (regional, two-dimensional steady-state models, analyses of ground water–surface water interaction, quick analyses and screening models, well head protection studies) (Csoma, 2005; Fredrick et al., 2004; Matott et al., 2006; Wuolo et al., 1995) as well as more recent applications (grid sensitivity analyses, estimating effective conductivity and dispersion in highly heterogeneous systems) (Jankovic et al., 2003; Luther and Haitjema, 1998). The review of applications also illustrates areas where more method development is needed (three-dimensional and transient simulations).

2.2.6 Hybrid Methods

Hybridization refers to the use of more than one numeric technique in the solution of potential flow problems. Mixed formulations are also widely used. The term mixed method was first used in the 1960's to describe finite element methods in which both stress and displacement fields are approximated as primary variables. Park (2005) applied the mixed finite element method for generalized Forchehimer Flow in Porous Media. Other examples of the utilization of mixed formulations are found in the following references: (Arnold, 1990; Bergamaschi et al., 1994; Ganis et al., 2008; Park, 2005).

Mixed-hybrid methods combine attractive features of both mixed and hybrid methods, namely the simultaneous approximation of the flux and potential, and the use of Lagrangian multipliers to enforce regularity constraints (Maryska et al., 1995; Van Criekingen and Beauwens, 2007).

In the Green element method, hybridisation is effected between the finite element method and the boundary element method. While the boundary element method is a powerful alternative to the finite element method, it has become apparent that to apply the theory to more robust, practically-based, and nonlinear problems, it is necessary to break faith with the boundary-only feature of the boundary element method. This is what has been done by adopting the Green element method. The Green element method is based on the boundary element theory, and as such the second-order accuracy commonly associated with the boundary element method is retained. However, Green element method implements the theory in an element-by-element manner (local support) so that, in contrast to the implementation procedure of BEM in which solution information is coupled for all nodes in the computational domain (global support), GEM couples information of nodes that share common elements (Taigbenu, 1999). In this way domain integrations are more easily carried out, medium variations can be readily accommodated, and non-linear potential flow problems are easier to handle. It is pertinent to comment that within recent years concerted efforts are now being made to adapt the boundary element technique to the solution of nonlinear problems (Taigbenu and

Onyejekwe, 1999). Taigbenu (1999) states that the difficulty in extending the theory of the boundary element method to nonlinear problems stemmed from the fact that the boundary element method was implemented to have “global support”, whereas methods like FDM and FEM which proceed in an element-by-element fashion or have “local support” are better able to handle the nonlinearity and variation of medium parameters across elements.

The Green element implementation procedure follows that of the finite element method. Thus the integral representations that result after applying the standard weighting procedure to the original differential equation in each element are assembled into a summation of the representations for all the elements.

The method briefly described above was given the name “Green element method” by Prof. Jim Liggett of Cornell University in 1987 (Taigbenu, 1999). Green element method computations may be conducted in either the Cartesian or the polar coordinate system. Onyejekwe (2006) points out that there are applications where the use of polar coordinates ensures better accuracy. The use of polar coordinates is not investigated in this work.

2.3 Boundary Element Formulation for 3D Laplace Equation

2.3.1 Basic Integral Equation

The development of the 3D Green element method formulation to the research problem is a process that has to begin with the development of the solution for the 3D Laplace equation, which is the steady state case of the research problem with no forcing term. The following is standard boundary element theory for the solution of the 3D Laplace equation. GEM is predicated on this theory.

Reference is made to Figure 2.1 on page 14. Consider that we are seeking to find the solution of the Laplace equation in a 3D domain.

$$\nabla^2 u = 0 \tag{2.10}$$

with the following conditions on the Γ boundary:

(i) Essential conditions of the type $u = \overline{u}$ on Γ_1

(ii) Natural conditions of the type $q = \frac{\partial u}{\partial n} = \overline{q}$ on Γ_2

Where n is the normal to the boundary $\Gamma = \Gamma_1 + \Gamma_2$ and the dashes indicate that those values are known. More complex boundary conditions such as a combination of the above, (i.e. $\alpha u + \beta q = \overline{V}$ where α and β are known parameters, and $\overline{V}$ represents a known value) could be included but will not be considered for the sake of simplicity.

The error introduced in the above equation if the exact but unknown values of u and q are replaced by an approximate solution can be minimised by orthogonalising them with respect to a weighted function u*, with normal derivative $q^* = \frac{\partial u^*}{\partial n}$.

If R are the residuals, then in general

$$\begin{aligned} R &= \nabla^2 u \neq 0 \\ R_1 &= u - \overline{u} \neq 0 \\ R_2 &= q - \overline{q} \neq 0 \end{aligned} \tag{2.11}$$

A standard weighting procedure can then be carried out and the following expression is obtained (Brebbia and Walker, 1980; Brebbia and Dominguez, 1992):

$$\int_{\Omega} (\nabla^2 u) u^* d\Omega = \int_{\Gamma_2} (q - \overline{q}) u^* d\Gamma - \int_{\Gamma_1} (u - \overline{u}) q^* d\Gamma \tag{2.12}$$

Or, after passing the derivatives on u to the weighting function u* by the process of carrying out integration by parts twice we get

$$\int_{\Omega}(\nabla^2 u^*)ud\Omega = -\int_{\Gamma_2}\bar{q}u^* d\Gamma - \int_{\Gamma_1} qu^* d\Gamma + \int_{\Gamma_2} uq^* d\Gamma + \int_{\Gamma_1}\bar{u}q^* d\Gamma \qquad \textbf{(2.13)}$$

Our aim is now to render equation (2.13) into a boundary integral equation, and this is done by using a special type of function u* called the fundamental solution which reduces the integral $\int_{\Omega}(\nabla^2 u^*)u\,d\Omega$ to values of u on the boundary.

2.3.2 Fundamental Solution

The fundamental solution u* satisfies Laplace's equation and represents the solution generated in an infinite space by a concentrated unit forcing function acting at a point x_i commonly referred to as the source point. Stating this in mathematical terms, the fundamental solution is obtained from:

$$\nabla^2 u^* + \Delta_i = 0, \quad or\ \nabla^2 u^* = -\Delta_i \qquad \textbf{(2.14)}$$

Where $\Delta_i \equiv \Delta(x - x_i)$ is the unit forcing function or Dirac delta function which comprises two arguments, namely the spatial field x and the source point x_i. It is a particularly interesting function that takes a value of zero everywhere except at the source point where it is infinite, yet its integral over the spatial domain is unity. That is:

$$\int_{\Omega}\Delta_i d\Omega = 1 \qquad \textbf{(2.15)}$$

Further, integrating the product of the Dirac delta function and any other function results in the value of the latter at point x_i, hence:

$$\int_{\Omega} u(\nabla^2 u^*) d\Omega = \int_{\Omega} u(-\Delta_i) d\Omega = -u_i \tag{2.16}$$

Equation (2.13) can now be written as:

$$u_i + \int_{\Gamma_2} uq^* d\Gamma + \int_{\Gamma_1} \bar{u}q^* d\Gamma = \int_{\Gamma_2} \bar{q}u^* d\Gamma + \int_{\Gamma_1} qu^* d\Gamma \tag{2.17}$$

The 1D, 2D, and 3D fundamental solutions for isotropic media are as given in the equations below (Brebbia and Walker, 1980; Taigbenu, 1999). In these equations x_i is the source node and r is the distance from the source node to the field node.

$$u^* = \frac{|x - x_i|}{2} \qquad \text{(1D)} \tag{2.18}$$

$$u^* = \frac{1}{2\pi} \ln \frac{1}{r} \qquad \text{(2D)} \tag{2.19}$$

$$u^* = \frac{1}{4\pi r} \qquad \text{(3D)} \tag{2.20}$$

In the case of anisotropic problems, or if the governing equation has nonconstant coefficients or nonlinear terms, such a straightforward fundamental solution is not available. Shiah et al. (2008) point out that the main impediment to the development of efficient algorithms for the stress analysis of 3D anisotropic elastic solids using the BEM over the years is the complexity of the fundamental solutions and the computational burden to evaluate them. To illustrate the complexity which the fundamental solution takes for anisotropic media, reference is made to the work of Wang and Schweizerhof (1995) who have presented fundamental solutions for laminated anisotropic shallow shells. Some authors have presented approximations whereby fundamental solutions for isotropic media can be used for anisotropic media (Narayanan et al., 1992; Schclar and

Partridge, 1993). Schclar and Partridge proceeded to express the anisotropic constants as a sum of average isotropic values and residuals, thus permitting the use of an isotropic fundamental solution to solve 3D anisotropic elasticity problems with the dual reciprocity boundary element method. In this work, anisotropy is not considered.

2.3.3 Boundary Integral Equation

Equation (2.17) is valid for any point x_i in the 3D domain Ω. In boundary element theory, it is preferable to apply the equation on the boundary, and hence we need to investigate what happens when x_i is on the boundary. A way to achieve this is to place x_i on the boundary Γ, and then to assume that the point x_i is at the centre of a sphere of radius ε as presented in Figure 2.2. As ε tends to zero, the resulting equation becomes the special case of (2.17) on the boundary Γ. Treatment of corner points is tackled at a later stage.

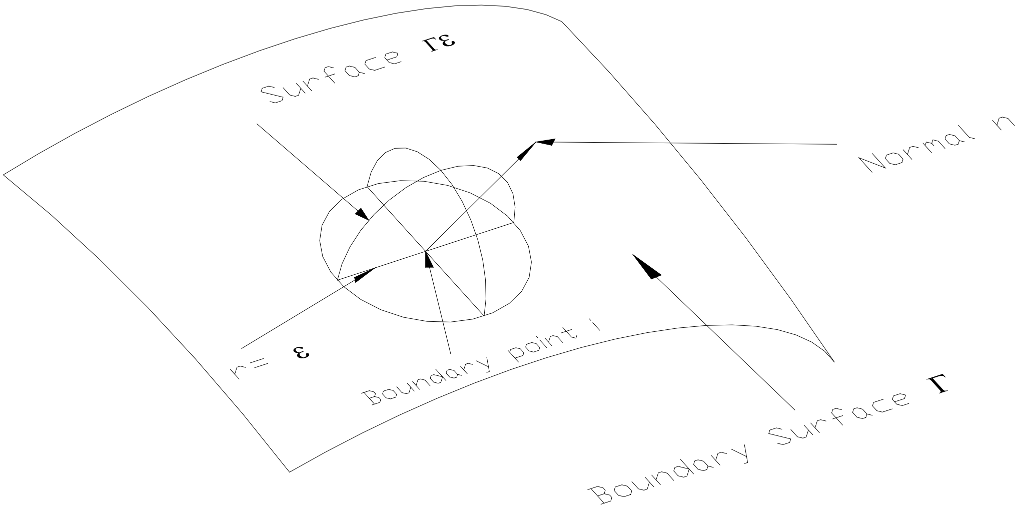


Figure 2.2. Boundary points for the three dimensional case, augmented by a small hemisphere.

Two types of boundary integrals can be identified in Equation (2.17). They include the primary variable u and the normal flux q which can be considered not to have been applied on any part of the boundary, i.e.:

$$u_i + \int_{\Gamma} uq^* d\Gamma = \int_{\Gamma} qu^* d\Gamma \tag{2.21}$$

The integral of the type shown on the right of Equation (2.21) tends to zero as ε tends to zero, as shown by the following derivation:

$$\begin{aligned} \lim_{\varepsilon \to 0} \left\{ \int_{\Gamma_\varepsilon} qu * d\Gamma \right\} &= \lim_{\varepsilon \to 0} \left\{ \int_{\Gamma_\varepsilon} q \frac{1}{4\pi\varepsilon} d\Gamma \right\} \\ &= \lim_{\varepsilon \to 0} \left\{ q \frac{2\pi\varepsilon^2}{4\pi\varepsilon} \right\} \equiv 0 \end{aligned} \tag{2.22}$$

The left hand side behaves in a different manner when ε tends to zero, as demonstrated below:

$$\begin{aligned} \lim_{\varepsilon \to 0} \left\{ \int_{\Gamma_\varepsilon} uq * d\Gamma \right\} &= \lim_{\varepsilon \to 0} \left\{ - \int_{\Gamma_\varepsilon} u \frac{1}{4\pi\varepsilon^2} d\Gamma \right\} \\ &= \lim_{\varepsilon \to 0} \left\{ -u \frac{2\pi\varepsilon^2}{4\pi\varepsilon^2} \right\} = -\frac{1}{2} u_i \end{aligned} \tag{2.23}$$

Thus the type of integral on the left hand side of equation (2.21) produces a free term. For this reason, we can write the following expression as the boundary integral equation for three dimensional problems:

$$\frac{1}{2} u_i + \int_{\Gamma} uq^* d\Gamma = \int_{\Gamma} qu^* d\Gamma \tag{2.24}$$

In general, the singular integral equation for a source point for the Laplace equation is:

$$ku_i + \int_\Gamma uq^* d\Gamma = \int_\Gamma qu^* d\Gamma \tag{2.25}$$

Notice that the coefficient 0.5 on u_i has been replaced by a general coefficient k because 0.5 only applies for the case when the included angle is 180 degrees. Liggett and Liu (1983) have shown that the general case when the included angle is not 180 degrees is $k = \lim_{R_0 \to 0}\left(A_s / R_0^2\right)$ where A_s is the surface area of the portion of a sphere of radius R_0 that lies inside the domain under consideration. For a domain discretised in the manner of the finite element method, each of the elements represents a domain in its own right.

For completeness, we present the form of equation (2.25) for a 1D spatial domain $x \in [0, L]$ and for a 2D spatial domain Ω with boundary Γ. They are:

$$ku_i + \left[uq^*\right]_{x=0}^{x=L} = \left[qu^*\right]_{x=0}^{x=L} \tag{2.26}$$

for 1D (where k=1 when x_i is within the domain, and 0.5 when it is at the end points) and

$$ku_i + \int_\Gamma uq^* d\Gamma = \int_\Gamma qu^* d\Gamma \tag{2.27}$$

for 2D where *k*=1 when the source point is within the domain, 0.5 on a smooth boundary, and $\theta/2\pi$ at a corner point with nodal angle θ .

2.3.4 Dual Reciprocity Boundary Element Formulation

The boundary element formulation is based on singular integrals, and these are intimately connected with the study of partial differential equations. Broadly speaking, a singular integral is an integral operator as already given in Equation (2.9), and repeated below for ease of reference,

$$u(x) = \int_{-\infty}^{\infty} G(x,s)\, f(s) ds \tag{2.9}$$

whose kernel function G is singular along the diagonal *x* = *s*. Calderon and Zygmund (1952) noted that while solutions for functions of one variable were readily available, the corresponding problems for functions of several variables had little been investigated, and proceeded to present work on functions in two variables.

Between 1952 to date, problems of two variables have been reasonably resolved and the solutions have been applied with adequate success to the solutions of potential flow problems in 2D domains. Solutions to problems in 3D using the singular integral theory have remained largely unresolved.

One of the disadvantages of employing the BEM is that the fundamental solution of the given differential operator is required. This, in principle, means that only homogeneous linear differential equations can be solved by the BEM. There are various ways to extend the applicability of the BEM to other types of partial differential equations. Where the differential equation has a forcing term on the right hand side of equation (2.10) (Poisson equation) and/or a temporal derivative term (diffusion equation), the relevant boundary integration requires evaluation of a domain integral, and BEM looses most of its attractiveness due to the fact that domain discretization and integrations are required (Chen et al., 2003).

Techniques for handling this problem of non-homogeneous terms include methods for transferring the domain integral to the boundary, and singularity programming. Among various methods for converting domain integrals into boundary integrals in the BEM for inhomogeneous PDEs, the most successful one is the so-called dual reciprocity boundary element method (DRBEM), which was first proposed Brebbia and Nardini (1983). While many variations, such as the Laplace transform dual reciprocity method, the separation of variables dual reciprocity method, and the perturbation dual reciprocity method have been proposed since its birth, the DRBEM is still evolving and many researches are currently actively involved in this area of research (Liu and Zhu, 2002).

The main idea of the DRBEM is to divide the solution into two parts: a known particular solution of the inhomogeneous PDE plus a complementary solution of its homogeneous counterpart (Portapila and Power, 2007). Since particular solutions to complex problems are very difficult or sometimes impossible to obtain, the inhomogeneity is approximated by a series of simpler radial basis functions of the form $\sum_{j=1}^{n} \alpha_j f_j(x)$ in which $f_j(x)$ are interpolation functions and α_j are the coefficients to be determined. The particular solutions for the basis functions can easily be determined.

There are many ways of choosing the basis functions $f_j(x)$, but it has been shown that while the interpolation functions work well for 2D problems, they become non-differentiable in 3D. In the series of radial basis functions $\sum_{j=1}^{n} \alpha_j f_j(x)$, $n = N + L$ and consists of the N boundary collocation and L internal nodes. To ensure the accuracy of the DRBEM solution, some internal nodes normally have to be included. A new technique without any internal collocation points, called the Multiple Reciprocity boundary element method has been proposed by Nowak and Brebbia (1992). The MRBEM can be thought of as an extension of the idea of DRBEM. However, instead of approximating the source term by the set of RBFs, a sequence of functions related to the fundamental solution is introduced. These functions constitute a set of higher order fundamental solutions which permit the second Green's identity to be applied to each term of the sequence. As a result, the MRBEM leads, in the limit, to the exact boundary only formulation of the domain integrals and therefore no internal collocation points are needed. Numerous applications of the DRBEM are now available (Natalini and Popov, 2007; Niku and Adey, 1996).

Singularity programming applies to the case where the right hand side is from a singular forcing function, for example borehole operation, and decomposes the solution into singular and non-singular components (Archer and Horne, 2000; Archer and Horne, 2002). The boundary element solution scheme is used to solve for the nonsingular solution, while the singular solution is known (Dake, 1978).

The keen interest in the boundary element method has lead some researchers to pursue developments for meshless integrations. Some such methods require the mesh only for background evaluation of the integrals appearing in their weak form, while others completely eliminate the use of a mesh by approximating the domain integral by a summation of products of weights and the values of the integrand at a set of nodes. The weight at each node represents the fraction of the total area associated with that node (Dai et al., 2004; Dolbow and Belytschko, 1999; Hu et al., 2007; Khosravifard and Hematiyan, 2009; Rosca and Leitao, 2008). Such methods are applicable to domains of homogeneous structure and can not be adopted for the solutions of potential flows in field aquifers where inhomogeneity is common.

3. SOLUTION OF THE 3D LAPLACE EQUATION

The objective of the current research project is to develop a numerical formulation based on the Green element method for time-dependent potential flow problems that extends the work done to date for the one dimensional and two dimensional cases. First the hypothesis was raised that it is possible to develop a three-dimensional formulation for the Green element method. Assuming that the three dimensional formulation could be developed, it was also the objective of the current project to test if the computational gains that have been reported for the one dimensional and two-dimensional cases do extend to three dimensions.

To accomplish the goal of developing a solution to the 3D Laplace equation, a number of objectives were undertaken. These objectives are listed below:

- development of the Green element formulation for the 3D Laplace equation
- the selection and generation of discretization elements
- development of a naming convention for potentials and fluxes to ensure ease in the handling of these variables during computations
- integrations for the solutions of the partial differential equations within the elements
- matrix manipulations

The result of the work in these objectives is outlined in the sections below.

3.1 Green Element Formulation for 3D Laplace Equation

The essence of Green element method is an element by element implementation of the boundary element theory. All the above theory for BEM is still applicable. The relevant integral equation expressed globally for the entire domain (see Section 2.3.3) as

$$ku_i + \int_{\Gamma} uq^* d\Gamma = \int_{\Gamma} qu^* d\Gamma \qquad (2.25)$$

is equally applicable to any isolated part of the domain, which in our case is a 3D element denoted as Ω_j with boundary Γ_j, that is:

$$k_j u_i + \int_{\Gamma_j} uq^* d\Gamma = \int_{\Gamma_j} qu^* d\Gamma \qquad (3.1)$$

This equation can be referred to as the element or discretised integral equation. However, it is of interest to obtain the global solution that comprises the solutions at prescribed positions of all elements used to discretize the entire domain Ω. That solution is obtained by aggregating the contributions of equation (3.1) for all the elements in such a way that the continuity of the potential u and the compatibility of the flux are maintained across inter-element boundaries. This is a huge challenge which is not as easy to implement as stated. It has been addressed in section 3.4 with the help of a carefully devised flux and potential labelling convention.

3.2 Selection of Discretization Elements

The choice of an element depends on the type of problem, the number of elements desired, the accuracy required, and the available computing resources. In 3D analysis, an element can refer either to the surfaces enclosing the domain as opposed to curvilinear elements for 2D problems, or the solids created by discretizing the domains. The surface elements are usually of two types: quadrilateral and triangular, and both can be either flat or curved. In this research, quadrilateral and triangular flat surface elements have been adopted. An element can also refer to the solids. We need to use solid polyhedrons. A polyhedron is any 3D body made up entirely of plane surfaces. A polyhedron is named after its number of faces. Thus a tetrahedron is composed of four triangular faces, three of which meet at each vertex, and a hexahedron has six plane faces. Prisms are a subset of polyhedrons, and they are defined as solid figures whose bases or ends have the same size and shape and are parallel to one another, and each of whose sides is a parallelogram. A prism is named according to the name of its base (i.e. shape of its cross-section).

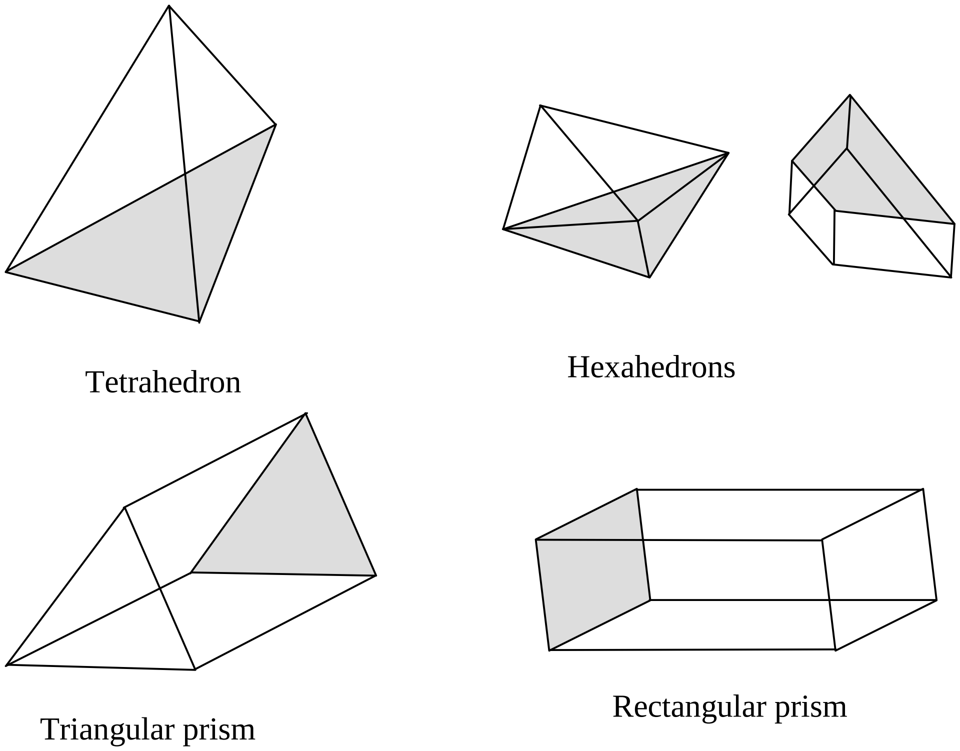


Figure 3.1. Examples of polyhedra shapes

For our purposes, we have investigated the usage of tetrahedral and hexahedra elements. For hexahedra elements, a special kind of the elements, in which the plane faces are quadrilateral (four-sided), are used.

Tetrahedral elements are the most adaptable type of elements to use for any domain shape, and would be preferable over hexahedra elements. For this reason, the investigation started with the adoption of tetrahedral elements, and thereafter the hexahedral elements were investigated.

The study of triangular and quadrilateral elements calls for a technique by which we can pass from the x, y, and z global coordinate system (coordinate system 1) to an element-localised system η_1, η_2, η_3 where η_1, η_2, are coordinates in a plane parallel to the plane of the surface and η_3 is the normal to the plane (see Figure 3.2). The η_1, η_2, η_3 system has been referred to as coordinate system 2. In the process of deriving standardized integration procedures with the aid of Jacobians, it became necessary to do a further pass between coordinate system 2 and a standardized system with either an isometric triangle of lengths 1x1x1 or a quadrilateral element of size 2x2 with a local origin at (0,0) and corner points at (1,-1), (1,1), (-1,1), and (-1,-1). Hence we define ξ_1, ξ_2, ξ_3 coordinate system as Coordinate System 3 which relates to the standardized first order elements as shown in Table 3.1.

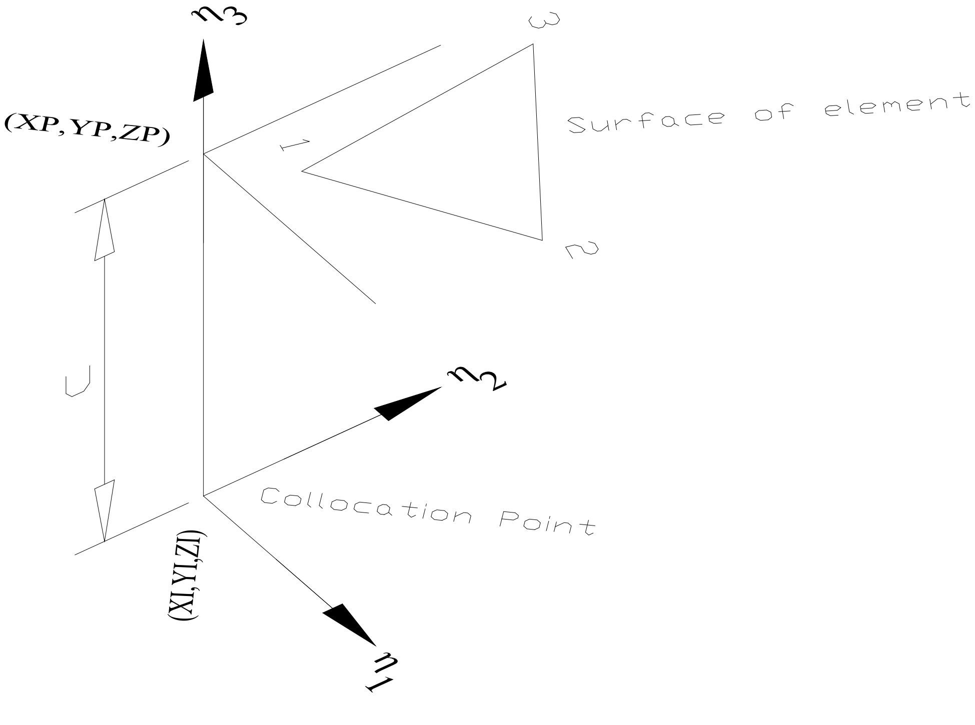
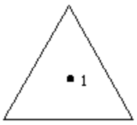
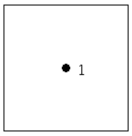
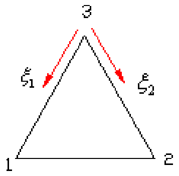
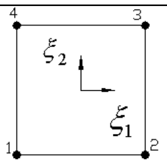


Figure 3.2. Triangular surface elements for 3D problems.

A choice of triangular and quadrilateral elements has been made for this research, since these are adequate for the type of problem being addressed. A further advantage of the choice of these elements is that a grid generation program, GID, can be utilised for the chore of grid generation.

Table 3.1 shows some of the triangular and quadrilateral surface elements which can be used, together with their interpolation polynomials. The choice of first order surfaces elements has been made. The signs within the parenthesis of each of the equations for ϕ are easily remembered because they correspond to the signs of ξ_1 and ξ_2 for the quadrant in which the node is located. It must be noted that the elements in Table 3.1 are standardized unit elements.

Table 3.1. Triangular and quadrilateral surface elements for 3D problems.

Type	Triangular		Quadrilateral	
Zero Order *Constant* *Elements*		$\phi_1 = 1$		$\phi_1 = 1$
First Order *Bilinear* *Elements*		$\phi_1 = \xi_1$ $\phi_2 = \xi_2$ $\phi_3 = 1 - \xi_1 - \xi_2$		$\phi_1 = \frac{1}{4}(1-\xi_1)(1-\xi_2)$ $\phi_2 = \frac{1}{4}(1+\xi_1)(1-\xi_2)$ $\phi_3 = \frac{1}{4}(1+\xi_1)(1+\xi_2)$ $\phi_4 = \frac{1}{4}(1-\xi_1)(1+\xi_2)$

3.3 Generation of Discretization Elements

Development and generation of discretization elements have been the subject of considerable studies and in this research it sufficed to procure an element generation algorithm. Various grid generation algorithms available on the market were studied and the results are provided in Appendix A.

From the investigations, GID grid generation program developed by the International Centre for Numerical Methods in Engineering (**CIMNE**) was selected. CIMNE is an autonomous research and development centre dedicated to promoting and fostering advances in the development and application of numerical methods and computational techniques for the solution of engineering problems in an international context.

3.4 Potential and Flux Labelling Conventions

The labelling convention for the flux is straightforward. At any node x in the discretised domain the potential is denoted by u_x.

(Lorinczi et al., 2009) adopted the following labelling scheme for fluxes in 2D domains. The x components of the flux at an internal node are denoted by q_x^{LT} and q_x^{LB} for the left-top and the left bottom components of q_x in the positive x direction, and by q_x^{RT} and q_x^{RB} for the right-top and the right-bottom components of q_x in the negative x direction, respectively. Similarly, for the y components of the flux, the left-bottom and the right-bottom components of the of q_y in the positive y direction are denoted by q_y^{LB} and q_y^{RB}, and the left-top and the right-top components of q_y in the negative y direction by q_y^{LT} and q_y^{RT} respectively. Needless to say, such a labelling system cannot be adopted in 3D domains where the concept of left, right, top, and bottom becomes confusing. We have developed a flux labelling system as explained below.

The first step in our flux-labelling system is to identify the unique surfaces in the discretised domain, and to label them sequentially. Two elements in contact with each other do so across one unique surface, not across two surfaces as would result if we allowed a surface from each of the elements. Our grid generation program, GID, generates 3D elements defined by their surfaces, and this generates duplicate surfaces even though the surface numbers do not repeat. We have created a script in FORTRAN to identify and remove the duplicate surfaces, and to label them appropriately. This script is provided in provided in Appendix C.

The next step is to associate each surface with a reference element. This is important because in the computations, the positive direction for flux is assumed to be in the direction normal to the surface and out of the volume. Hence if it is established in the computations that the surface is not referenced to the element the sense of the flux will be reversed.

The final step in the flux labelling scheme is to note the node at which the flux is positioned. Thus in Figure 3.3 $q_{2\text{-}6}$ is referenced to element 1 and refers to the flux that is normal to surface 2 and located at node 6. The variable is stored with information about the element to which it is referenced.

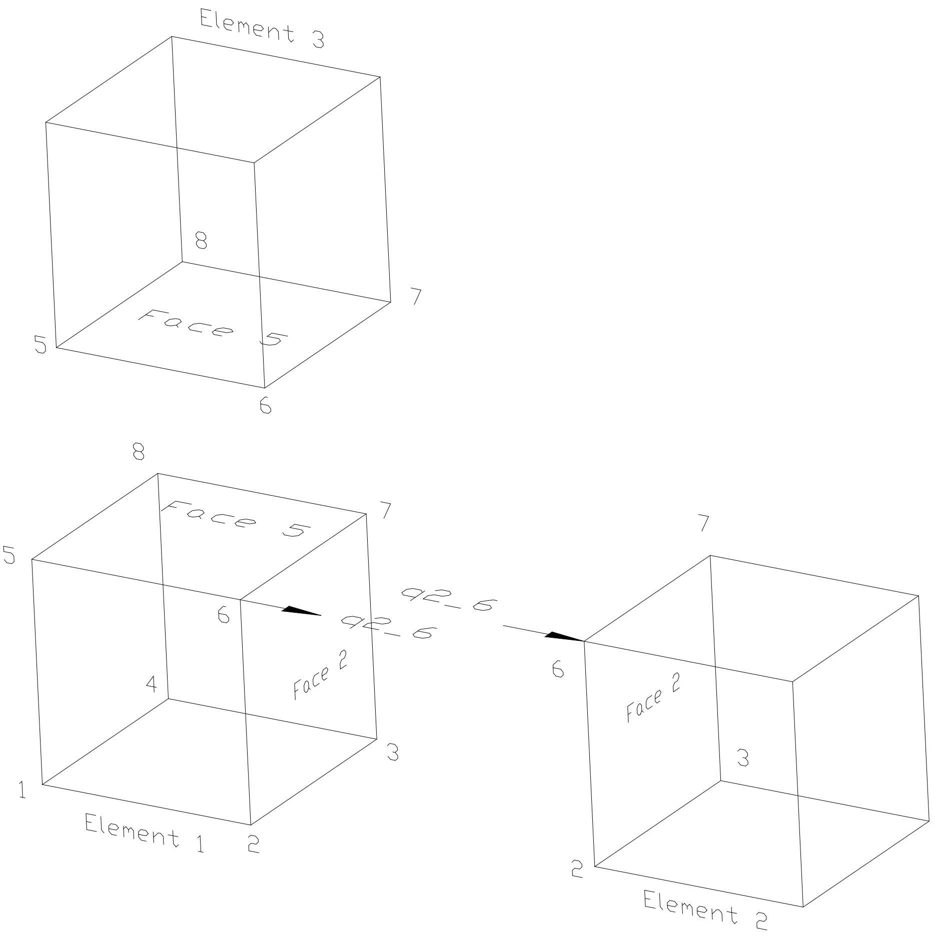


Figure 3.3. Flux labelling convention.

3.5 Element Integrations for Tetrahedra Elements in 3D Domains

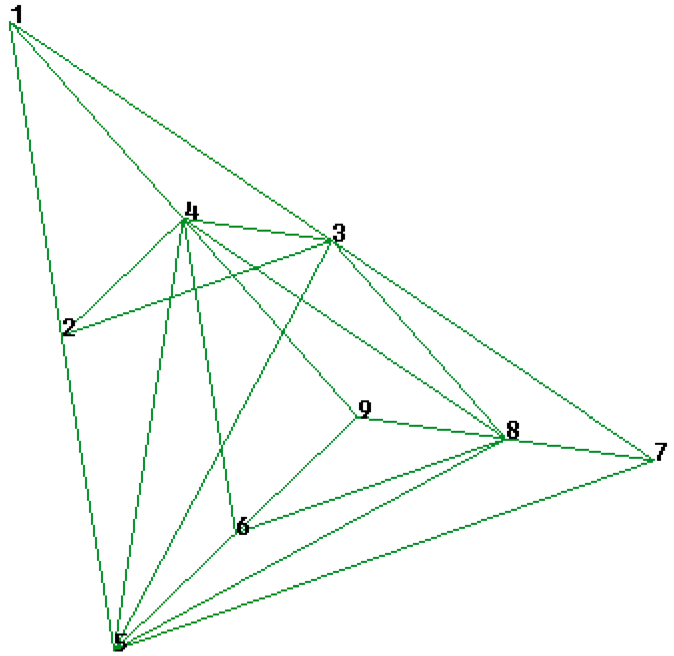


Figure 3.4. Example of tetrahedral volume elements generated by GID software.

For the tetrahedral volume elements selected, the plane surfaces are triangles. To ease the computations, a local coordinate system is defined for each surface over which the integral is being carried out. The local coordinate is defined in such a way that the origin is at the collocation point, the η_1 and η_2 coordinates are in a plane parallel to the surface over which the integration is being carried out but passing through the collocation point, and the η_3 coordinate is perpendicular to the surface (Figure 3.2). By choosing the local coordinate system in this manner, the derivative $\partial r / \partial n$ in three-dimensional space reduces to a derivative in two-dimensional space over the surface, because the normal distance from the collocation point to the surface is a constant. The derivation of the derivative for the fundamental solution q* follows from the expression of the fundamental solution as follows:

The fundamental solution for the Laplace equation, as has already been observed in Section 2.3.2, is given by:

$$u^* = \frac{1}{4\pi r} \qquad \textbf{(2.20)}$$

The derivative of the fundamental solution over the surface is given by:

$$q^* = \frac{\partial u^*}{\partial \eta} = \frac{\partial u^*}{\partial \eta_3} = -\frac{1}{4\pi r^2}\frac{\partial r}{\partial \eta_3} \qquad \textbf{(3.2)}$$

In Equation (3.2) r is the distance from the collocation point to the surface. With the local coordinate system defined, we note that over the surface r is given by:

$$r = \sqrt{\eta_3{}^2 + \eta_1{}^2 + \eta_2{}^2} \qquad \textbf{(3.3)}$$

So that

$$\frac{\partial r}{\partial \eta_3} = \left(\eta_3{}^2 + \eta_1{}^2 + \eta_2{}^2\right)^{-1/2} . \eta_3 \qquad \textbf{(3.4)}$$

Because of the choice of coordinate System 2, η_3 over the surface is a constant, denoted by c. Therefore, from Equation (3.2) and Equation (3.4), it follows that

$$\begin{aligned} q^* = \frac{\partial u^*}{\partial n} = \frac{1}{4\pi r^2}\frac{\partial r}{\partial \eta_3} &= -\frac{1}{4\pi r^2}\left(c^2 + \eta_1{}^2 + \eta_2{}^2\right)^{-1/2} .c \\ &= -\frac{c}{4\pi r^{3/2}} \end{aligned} \qquad \textbf{(3.5)}$$

This is the same as for hexahedra elements.

For a discretised integration domain, Equation (2.25) can be written as:

$$ku_i + \sum_{j=1}^{N}\left(\int_{\Gamma_j} uq^* d\Gamma\right) = \sum_{j=1}^{N}\left(\int_{\Gamma_j} qu^* d\Gamma\right) \tag{3.6}$$

The solutions for u and q are unknown, but can be written in terms of the nodal values using interpolating functions as given in Table 3.1. Our goal is then to solve for these unknown nodal values. The solution will be a mixed solution, because the nodal values of both the potential u and the flux q will be obtained.

From Equation (3.6), and using the interpolation functions ϕ_1, ϕ_2, and ϕ_3 for the elements, we obtain the following result for the term $\int_{\Gamma} uq^* d\Gamma$.

$$\int_{\Gamma} uq^* d\Gamma \approx \int_{\Gamma} [\phi_1 \quad \phi_2 \quad \phi_3] q^* d\Gamma \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \tag{3.7}$$

If the product of the interpolation functions ϕ and q* is denoted by *h*, then we can write:

$$\int_{\Gamma} uq^* d\Gamma \approx \int_{\Gamma} \left[h^{ij}{}_1 \quad h^{ij}{}_2 \quad h^{ij}{}_3\right] d\Gamma \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \tag{3.8}$$

Where i= source node

j= Surface on which integration is being done

1, 2, 3= Boundaries of surface j

Similarly, for the term $\int_{\Gamma} qu^{*} d\Gamma$ we have

$$\int_{\Gamma} qu^{*} d\Gamma \approx \int_{\Gamma} \begin{bmatrix} g^{ij}{}_{1} & g^{ij}{}_{2} & g^{ij}{}_{3} \end{bmatrix} d\Gamma \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \end{Bmatrix} \tag{3.9}$$

To perform the integrations, it is necessary to obtain the equation of the surface on which the integrations are being done, the coordinates of the collocation point, and the coordinates on which the normal line from the collocation point intersects the surface, all in the global coordinate system (Coordinate System 1) initially, and then using this information to define the local Coordinate System 2 with origin at the collocation point. All integrations are done in the Coordinate system 3 in a standard fashion. After the integrations are done there is need to scale the result to the Coordinate system 2. Therefore, the Jacobian of System 2 with respect to Coordinate System 3 is evaluated.

The equation of a plane in the global coordinate System 1 is given by

$$Ax + By + Cz + D = 0 \tag{3.10}$$

Where A, B, C, and D are constants and can be computed using the three corner coordinates of the surface. In Equation (3.10) A, B, and C represent the components of the vector normal to the plane. Because of this,

$$\begin{aligned} (XP - XI) &= cA/\sqrt{A^2 + B^2 + C^2} \\ (YP - YI) &= cB/\sqrt{A^2 + B^2 + C^2} \\ (ZP - ZI) &= cC/\sqrt{A^2 + B^2 + C^2} \end{aligned} \tag{3.11}$$

Where (XP, YP, ZP) denotes the intersection point on the plane, (XI, YI, ZI) denotes the collocation point, and c denotes the distance from the collocation point to the intersection point on the plane (see Figure 3.2).

Expressing Equations (3.10) and (3.11) in matrix notation we have

$$\begin{bmatrix} A & B & C & 0 \\ 1 & 0 & 0 & -\frac{A}{\sqrt{A^2+B^2+C^2}} \\ 0 & 1 & 0 & -\frac{B}{\sqrt{A^2+B^2+C^2}} \\ 0 & 0 & 1 & -\frac{C}{\sqrt{A^2+B^2+C^2}} \end{bmatrix} \begin{Bmatrix} XP \\ YP \\ ZP \\ c \end{Bmatrix} = \begin{Bmatrix} -D \\ XI \\ YI \\ ZI \end{Bmatrix} \tag{3.12}$$

Thus XP, YP, ZP and c can be evaluated in one sweep using matrix algebra. With XP, YP, and ZP evaluated we proceed to calculate $\eta_1^1, \eta_1^2, \eta_1^3$ and $\eta_2^1, \eta_2^2, \eta_2^3$ as coordinates of corner nodes 1, 2, and 3 of the surface element in the coordinate System 2. The superscripts represent the nodes, while the subscripts represent the axes.

Finally, isoparametric mapping is done. The transformation is achieved by the following relationships:

$$\eta_1 = [\phi_1 \quad \phi_2 \quad \phi_3] \begin{bmatrix} \eta_1^1 \\ \eta_1^2 \\ \eta_1^3 \end{bmatrix} \text{and } \eta_2 = [\phi_1 \quad \phi_2 \quad \phi_3] \begin{bmatrix} \eta_2^1 \\ \eta_2^2 \\ \eta_2^3 \end{bmatrix} \tag{3.13}$$

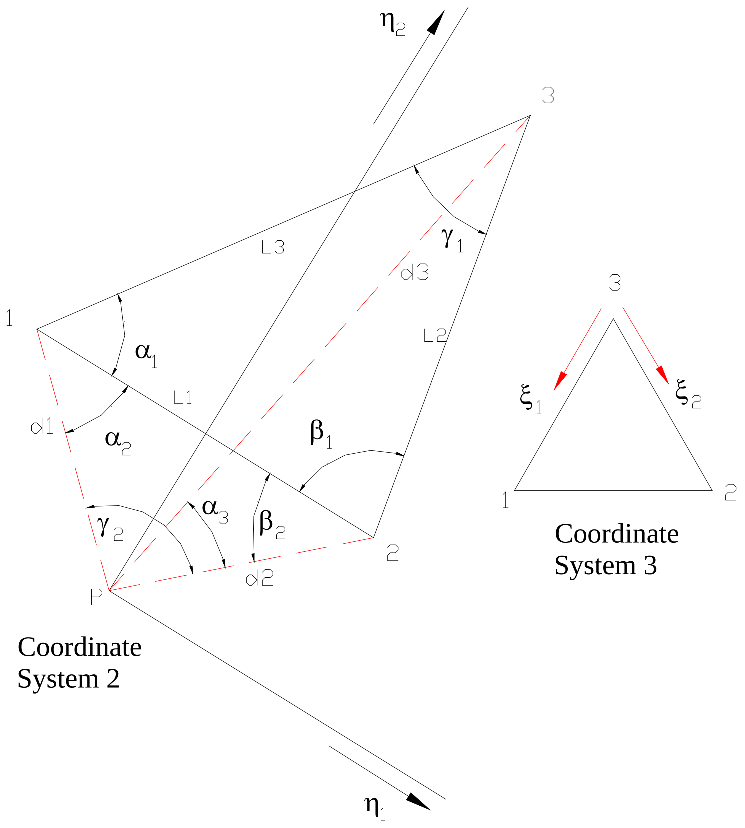


Figure 3.5. Transformation from Coordinate System 3 to System 2.

Since η_1^1, η_1^2, η_1^3 and η_2^1, η_2^2, η_2^3 are constants for any surface element, the above equations yield relationships between η_1 and ξ_1, ξ_2 and between η_2 and ξ_1, ξ_2. This relationship arises because the interpolating functions ϕ_1, ϕ_2, and ϕ_3 are functions of ξ_1 and ξ_2.

Hence the Jacobian J of η_1, η_2 system (System 2) with respect to the standardised ξ_1, ξ_2 (System 3) can be evaluated.

The transformation is given by the equations

$$\eta_1 = w_1(\xi_1, \xi_2)$$
$$\eta_2 = w_2(\xi_1, \xi_2)$$

Where

$$w_1 = A0 + A1\xi_1 + A2\xi_2$$
$$w_2 = B0 + B1\xi_1 + B2\xi_2 \qquad \textbf{(3.14)}$$

A0, A1, A2 and B0, B1, B2 are constants given by

$$A0 = \eta_1^3$$
$$A1 = \eta_1^1 - \eta_1^3$$
$$A2 = \eta_1^2 - \eta_1^3$$
$$B0 = \eta_2^3$$
$$B1 = \eta_2^1 - \eta_2^3$$
$$B2 = \eta_2^2 - \eta_2^3 \qquad \textbf{(3.15)}$$

The required Jacobian J is defined as

$$\text{J=}\begin{bmatrix} \dfrac{\partial \eta_1}{\partial \xi_1} & \dfrac{\partial \eta_1}{\partial \xi_2} \\ \dfrac{\partial \eta_2}{\partial \xi_1} & \dfrac{\partial \eta_2}{\partial \xi_2} \end{bmatrix} \qquad \textbf{(3.16)}$$

This has been evaluated as

$$J = A1 * B2 - A2 * B1 \qquad \textbf{(3.17)}$$

The above considerations lead to the following expressions for the integrals

$$h_k^{ij} = \int_{\Gamma_j} \phi_k q_i^{\,*} d\Gamma_j \text{ and } g_k^{ij} = \int_{\Gamma_j} \phi_k u_i^{\,*} d\Gamma_j \,. \qquad \textbf{(3.18)}$$

To achieve transformation from Coordinate system 3 over the standardised surface Ψ to Coordinate System 2 over the surface Γ, we use the following relationship.

$$\iint_{\Gamma} f(\eta_1, \eta_2) d\eta_1 \eta_2 = \iint_{\Psi} f\left[w_1(\xi_1, \xi_2), w_2(\xi_1, \xi_2)\right] * J * d\xi_1 d\xi_2 \qquad \textbf{(3.19)}$$

Where $\eta_1 = w_1(\xi_1, \xi_2)$ *and* $\eta_2 = w_2(\xi_1, \xi_2)$ as given by (3.14) above. The integrals are listed in Table 3.2.

Table 3.2. Representations for surface integrals h_k^{ij} and g_k^{ij}.

Item	Coordinate System 2	Coordinate System 3	Exact Representation
h_1^{ij}	$\int_{\Gamma} \phi_1 q_i^{\,*} d\Gamma$	$\int_0^1\int_0^1 \phi_1 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * J * \int_0^1\int_0^1 \frac{\xi_1}{\left(k_0 + k_1\xi_1 + k_2\xi_2 + k_3\xi_1\xi_2 + k_4\xi_1^2 + k_5\xi_2^2\right)^{3/2}} d\xi_1 d\xi$
h_2^{ij}	$\int_{\Gamma} \phi_2 q_i^{\,*} d\Gamma$	$\int_0^1\int_0^1 \phi_2 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * J * \int_0^1\int_0^1 \frac{\xi_2}{\left(k_0 + k_1\xi_1 + k_2\xi_2 + k_3\xi_1\xi_2 + k_4\xi_1^2 + k_5\xi_2^2\right)^{3/2}} \xi_1 d\xi$
h_3^{ij}	$\int_{\Gamma} \phi_3 q_i^{\,*} d\Gamma$	$\int_0^1\int_0^1 \phi_3 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * J * \int_0^1\int_0^1 \frac{(1 - \xi_1 - \xi_2)}{\left(k_0 + k_1\xi_1 + k_2\xi_2 + k_3\xi_1\xi_2 + k_4\xi_1^2 + k_5\xi_2^2\right)^{3/2}} d\xi_1 d\xi_2$
g_1^{ij}	$\int_{\Gamma} \phi_1 u_i^{\,*} d\Gamma$	$\int_0^1\int_0^1 \phi_1 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * J * \int_0^1\int_0^1 \frac{\xi_1}{\left(k_0 + k_1\xi_1 + k_2\xi_2 + k_3\xi_1\xi_2 + k_4\xi_1^2 + k_5\xi_2^2\right)^{1/2}} d\xi_1 d\xi$
g_2^{ij}	$\int_{\Gamma} \phi_2 u_i^{\,*} d\Gamma$	$\int_0^1\int_0^1 \phi_2 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * J * \int_0^1\int_0^1 \frac{\xi_2}{\left(k_0 + k_1\xi_1 + k_2\xi_2 + k_3\xi_1\xi_2 + k_4\xi_1^2 + k_5\xi_2^2\right)^{1/2}} d\xi_1 d\xi$
g_3^{ij}	$\int_{\Gamma} \phi_3 u_i^{\,*} d\Gamma$	$\int_0^1\int_0^1 \phi_3 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * J * \int_0^1\int_0^1 \frac{(1 - \xi_1 - \xi_2)}{\left(k_0 + k_1\xi_1 + k_2\xi_2 + k_3\xi_1\xi_2 + k_4\xi_1^2 + k_5\xi_2^2\right)^{1/2}} d\xi_1 d\xi$

Because η_1 and η_2 are given by (3.14) above, $k_0, k_1, k_2, k_3, k_4,$ *and* k_5 in Table 3.2 above have the following expressions.

$$
\begin{aligned}
k_0 &= c^2 + A0^2 + B0^2 \\
k_1 &= 2(A0 * A1 + B0 * B1) \\
k_2 &= 2 * (A0 * A2 + B0 * B2) \\
k_3 &= 2 * (A1 * A2 + B1 * B2) \\
k_4 &= A1^2 + B1^2 \\
k_5 &= A2^2 + B2^2
\end{aligned}
\tag{3.20}
$$

The Jacobian (J) is used to make the scaling from coordinate system 3 to coordinate system 2. The exact representations of the integrals are given in column 4 of the table. The integrations were done with *Mathematica* suite of numerical analysis. One of the important features of *Mathematica* is that it can do *symbolic* as well as *numerical* calculations (Wolfram Research, 2010). Symbolic computation or algebraic computation relates to the use of machines, such as computers, to manipulate mathematical equations and expressions in symbolic form, as opposed to manipulating the approximations of specific numerical quantities represented by those symbols (Watt, 2006). Such a system is useful for symbolic integration or differentiation, substitution of one expression into another, simplification of an expression, etc. Mathematica can handle algebraic formulas as well as numbers.

When an attempt was made to reconcile the degrees of freedom with the number of generated equations at an internal node, it was found that there are four (4) discrete equations and sixteen (16) degrees of freedom (12 for the fluxes, and 4 for the potentials). In any mesh, rules exist that relate the number of internal and external sides, vertices, etc. and the total number of elements. These have been given explicitly for plane meshes of triangles and quadrilaterals, and for solid meshes of tetrahedral and cuboidal elements (Ewing et al., 1970). Thus, for solid meshes of T elements having F_b boundary and F_i internal faces, V_i internal nodes, E_i internal and E_b boundary edges, and H through holes and h cavities, the following relationships relate the number of elements to the number of faces, the number of internal nodes, and the number of boundary edges. Holes that go through a domain (H) and cavities that are internal (h) are frequent stress concentrators that require careful analysis. The relationships are

$$T = \frac{1}{6}\left(F_b + 2F_i\right) = \frac{1}{8}E_b + \frac{1}{2}\left(E_i - V_i + H - h - 1\right) \qquad \textbf{(3.21)}$$

for cuboidal elements, and

$$T = \frac{1}{4}\left(F_b + 2F_i\right) = \frac{1}{3}E_b + E_i - V_i + H - h - 1 \qquad \textbf{(3.22)}$$

for tetrahedral elements.

The number of nodes represents the number of integral equations that allow for nodal solution computations, while the number of internal faces is directly related to the number of unknown fluxes by a factor of three for tetrahedral elements four for cuboidal elements. Therefore, when an attempt is made to calculate all normal fluxes, a closure problem is encountered as there are too few integral equations than unknowns.

This problem of too many degrees of freedom is one that was recognised in 1D and 2D formulations of the Green element method. The resolution for 2D formulations is presented in Taigbenu (2008). In his work, Taigbenu presented what he termed the flux-correct Green element formulation for linear and nonlinear heat transport in heterogeneous media, together with a methodology to resolve the closure problem. He goes further to explain that in the past this closure problem was one among other reasons why it was unattractive to carry out direct calculations of the normal derivatives of the primary variable within internal boundaries in singular boundary integral formulations. The alternative to retaining the internal fluxes is approximating them by a finite difference expressed in terms of the nodal potentials as presented in Taigbenu (1995). Although this is a viable approach for the 3D tetrahedral elements, it is only explored in this work for the simpler cuboidal elements. This approach for the cuboidal elements is presented in a subsequent section.

3.6 Element Integrations for Hexahedra Elements in 3D Domains

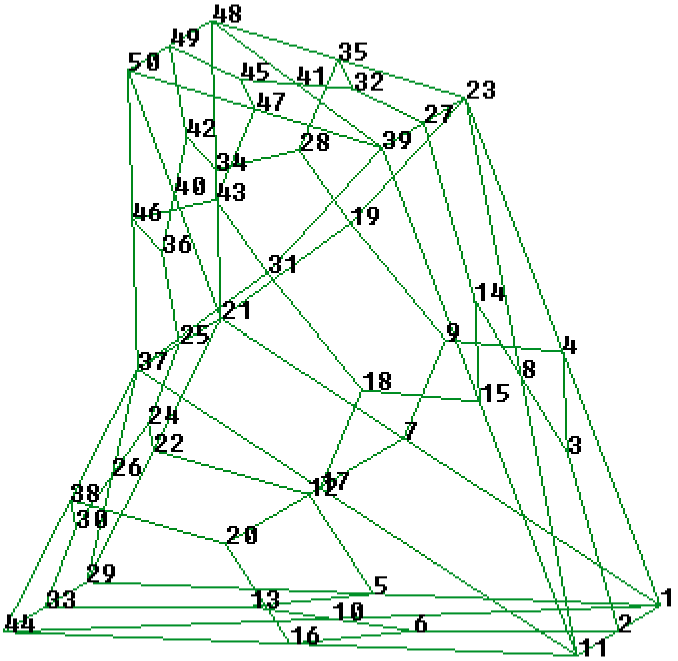


Figure 3.6. Example of hexahedral volume elements generated by GID software.

As earlier done for tetrahedral elements, the Green element calculations for hexahedra elements involves surface integrations, but this time over quadrilateral surfaces as opposed to triangular surfaces. To ease the computations, a local coordinate system is defined for each surface over which the integral is being carried out (Figure 3.7).

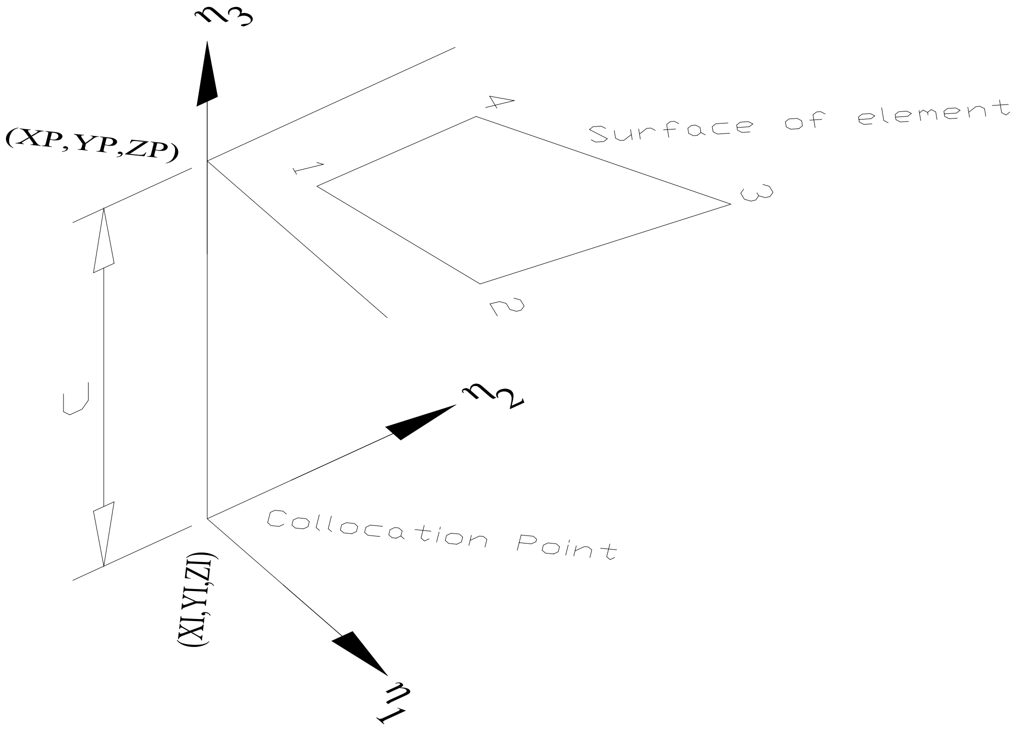


Figure 3.7. Quadrilateral surface elements for 3D problems.

The local coordinate is defined in such a way that the origin is at the collocation point, the η_1 and η_2 coordinates are in a plane parallel to the surface over which the integration is being carried out but passing through the collocation point, and the η_3 coordinate is perpendicular to the surface as was done in Section 3.5 above. This is illustrated by equations (3.2) to (3.5) in that section.

From Equation (3.6), and using the interpolation functions ϕ_1, ϕ_2, ϕ_3, and ϕ_4 for the elements, we obtain the following result for the term $\int_\Gamma uq^* d\Gamma$.

$$\int_{\Gamma} uq^{*}d\Gamma \approx \int [\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4]q^{*}d\Gamma \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} \tag{3.23}$$

The nodal values of the potential have been taken out of the integral operator because they do not depend on the spatial variables.

If the product of the interpolation functions ϕ and q* is denoted by h, then we can write:

$$\int_{\Gamma} uq^{*}d\Gamma \approx \int_{\Gamma} \left[h^{ij}{}_1 \quad h^{ij}{}_2 \quad h^{ij}{}_3 \quad h^{ij}{}_4 \right] d\Gamma \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{Bmatrix} \tag{3.24}$$

Where i= source node

j= Surface on which integration is being done

1, 2, 3, 4= Boundaries of surface j

Similarly, for the term $\int_{\Gamma} qu^{*}d\Gamma$ we have

$$\int_{\Gamma} qu^{*}d\Gamma \approx \int_{\Gamma} \left[g^{ij}{}_1 \quad g^{ij}{}_2 \quad g^{ij}{}_3 \quad g^{ij}{}_4 \right] d\Gamma \begin{Bmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{Bmatrix} \tag{3.25}$$

To perform the integrations, a procedure similar to what was followed in the previous section is followed. Use is made of the Jacobian (J) of coordinate System 2 with respect to System 3.

The equation of a plane in the global coordinate System 1 is also obtained in the same fashion as was done in section 3.5 for tetrahedral elements. Thus XP, YP, ZP and c can be evaluated in one sweep using matrix algebra, and with XP, YP, and ZP evaluated we proceed to calculate $\eta_1^1, \eta_1^2, \eta_1^3, \eta_1^4$ and $\eta_2^1, \eta_2^2, \eta_2^3, \eta_2^4$ as coordinates of corner nodes 1, 2, 3 and 4 of the surface element in the coordinate System 2. The superscripts represent the nodes, while the subscripts represent the axes.

The interpolation functions given in Table 3.1 for bilinear elements are used to achieve isoparametric mapping between Coordinate System 2 and Coordinate System 3. Isoparametric mapping is achieved by the following relationships:

$$\eta_1 = [\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4]\begin{bmatrix}\eta_1^1\\ \eta_1^2\\ \eta_1^3\\ \eta_1^4\end{bmatrix} \text{ and } \eta_2 = [\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4]\begin{bmatrix}\eta_2^1\\ \eta_2^2\\ \eta_2^3\\ \eta_2^4\end{bmatrix} \qquad \textbf{(3.26)}$$

Since $\eta_1^1, \eta_1^2, \eta_1^3, \eta_1^4$ and $\eta_2^1, \eta_2^2, \eta_2^3, \eta_2^4$ are constants for any surface element, the above equations yield relationships between η_1 and ξ_1, ξ_2 and between η_2 and ξ_1, ξ_2 as has been explained before. Hence the Jacobian J of η_1, η_2 system (System 2) with respect to the standardised ξ_1, ξ_2 (System 3) can be evaluated.

The transformation is given by the equations

$$\eta_1 = w_1(\xi_1, \xi_2)$$
$$\eta_2 = w_2(\xi_1, \xi_2)$$

(3.27)

where

$$w_1 = B0 + B1\xi_1 + B2\xi_2 + B3\xi_1\xi_2$$
$$w_2 = B0 + B1\xi_1 + B2\xi_2 + B3\xi_1\xi_2$$

A0, A1, A2, A3 and B0, B1, B2, B3 are constants given by

$$
\begin{aligned}
&A0=\left(\eta_1^1+\eta_1^2+\eta_1^3+\eta_1^4\right)/\,4\\
&A1=\left(-\eta_1^1+\eta_1^2+\eta_1^3-\eta_1^4\right)/\,4\\
&A2=\left(-\eta_1^1-\eta_1^2+\eta_1^3+\eta_1^4\right)/\,4\\
&A3=\left(\eta_1^1-\eta_1^2+\eta_1^3-\eta_1^4\right)/\,4\\
&B0=\left(\eta_2^1+\eta_2^2+\eta_2^3+\eta_2^4\right)/\,4\\
&B1=\left(-\eta_2^1+\eta_2^2+\eta_2^3-\eta_2^4\right)/\,4\\
&B2=\left(-\eta_2^1-\eta_2^2+\eta_2^3+\eta_2^4\right)/\,4\\
&B3=\left(\eta_2^1-\eta_2^2+\eta_2^3-\eta_2^4\right)/\,4
\end{aligned}
\qquad \textbf{(3.28)}
$$

The required Jacobian J is already defined by Equation (3.16)

$$
\mathrm{J}=\begin{bmatrix}\dfrac{\partial\eta_1}{\partial\xi_1} & \dfrac{\partial\eta_1}{\partial\xi_2}\\ \dfrac{\partial\eta_2}{\partial\xi_1} & \dfrac{\partial\eta_2}{\partial\xi_2}\end{bmatrix} \qquad \textbf{(3.16)}
$$

This has been evaluated as

$$
J = L0 + L1\xi_1 + L2\xi_2 \qquad \textbf{(3.29)}
$$

where L0, L1, and L2 are constants evaluated as

$$
\begin{aligned}
&L0 = A1 * B2 - A2 * B1\\
&L1 = B1 * (A1 - A3)\\
&L2 = A3 * B2 - A2 * B3
\end{aligned}
\qquad \textbf{(3.30)}
$$

Note the simplicity of the Jacobian for the coordinate transformation for the tetrahedral elements with triangular surfaces in Equation (3.17) compared to that for the hexahedral elements with quadrilateral surfaces in Equation (3.29).

The above considerations lead to the following expressions for the integrals

$$h_k^{ij} = \int_{\Gamma_j} \phi_k q_i^{\ *} d\Gamma_j \text{ and } g_k^{ij} = \int_{\Gamma_j} \phi_k u_i^{\ *} d\Gamma_j \,. \qquad \textbf{(3.31)}$$

To achieve transformation from Coordinate system 3 to Coordinate System 2, we use the following relationship.

$$\iint_{\Gamma} f(\eta_1,\eta_2) d\eta_1\eta_2 = \iint_{\Psi} f\left[w_1(\xi_1,\xi_2), w_2(\xi_1,\xi_2)\right] * J * d\xi_1 d\xi_2 \qquad \textbf{(3.32)}$$

Where $\eta_1 = w_1(\xi_1,\xi_2)$ *and* $\eta_2 = w_2(\xi_1,\xi_2)$ are as given by Equation (3.27) above. Ψ and Γ represent the standardised surface in Coordinate system 3 and the integration surface in Coordinate System 2 respectively. The integrals are listed in Table 3.3.

Table 3.3. Representations for surface integrals h_k^{ij} and g_k^{ij}.

Item	Coordinate System 2	Coordinate System 3	Exact Representation
h_1^{ij}	$\int_\Gamma \phi_1 q_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_1 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1-\xi_1-\xi_2+\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{3/2}} d\xi_1 d\xi_2$
h_2^{ij}	$\int_\Gamma \phi_2 q_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_2 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1+\xi_1-\xi_2-\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{3/2}} d\xi_1 d\xi_2$
h_3^{ij}	$\int_\Gamma \phi_3 q_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_3 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1+\xi_1+\xi_2+\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{3/2}} d\xi_1 d\xi_2$
h_4^{ij}	$\int_\Gamma \phi_4 q_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_4 q^* J d\xi_1 d\xi_2$	$\frac{-c}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1-\xi_1+\xi_2-\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{3/2}} d\xi_1 d\xi_2$
g_1^{ij}	$\int_\Gamma \phi_1 u_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_1 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1-\xi_1-\xi_2+\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{1/2}} d\xi_1 d\xi_2$
g_2^{ij}	$\int_\Gamma \phi_2 u_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_2 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1+\xi_1-\xi_2-\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{1/2}} d\xi_1 d\xi_2$
g_3^{ij}	$\int_\Gamma \phi_3 u_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_3 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1+\xi_1+\xi_2+\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{1/2}} d\xi_1 d\xi_2$
g_4^{ij}	$\int_\Gamma \phi_4 u_i^* d\Gamma$	$\int_{-1}^{1}\int_{-1}^{1} \phi_4 u^* J d\xi_1 d\xi_2$	$\frac{1}{16\pi} * \int_{-1}^{1}\int_{-1}^{1} \frac{(1-\xi_1+\xi_2-\xi_1\xi_2)(L0+L1\xi_1+L2\xi_2)}{(c^2+\eta_1^2+\eta_2^2)^{1/2}} d\xi_1 d\xi_2$

Similar to the previous section, the Jacobian (J) is used to make the transformation from coordinate system 3 to coordinate system 2. The exact representations of the integrals are given in column 4 of the table. A transformation of the denominator of the integrand into Coordinate system 3 rendered the integrand highly complicated to handle, even with the use of MATHEMATICA integration routines. However, when the generalised hexahedral elements were restricted to cuboids, the symmetry of the elements enabled replacement of those integrals which had complex (imaginary) results with the simpler versions, simply by consideration of symmetry.

When an attempt was made to reconcile the degrees of freedom with the number of generated equations at an internal node, it was found that there are six (6) discrete equations and thirty (30) degrees of freedom (24 for the fluxes, and 6 for the potentials). Approximating the fluxes by a differencing operation along the lines of (Taigbenu, 1995) is straightforward in this case, provided that cuboid elements are used. This provides the motivation for the adoption of cuboid hexahederal elements. As was noted in 3.5, mesh rules exist that relate the number of internal and external sides, vertices, etc. and the total number of elements. These have been given explicitly for plane meshes of triangles and quadrilaterals, and for solid meshes of tetrahedral and cuboidal elements (Ewing et al., 1970).

4. POISSON AND THE TIME-DEPENDENT POTENTIAL FLOW EQUATIONS

To date most applications of GEM are in 1D and 2D domains in which time dependent and nonlinear problems in heterogeneous domains have been addressed. This chapter describes the 3D GEM formulation for the Poisson and time dependent potential problems.

4.1 General Considerations

Up to Chapter 3, a formulation of the Green element method for the solution of the Laplace equation has been presented. Attempts to incorporate the more general tetrahedral or hexahedral elements into the formulation meets with the challenges of expressing the internal fluxes in terms of the nodal potentials. However, this is easier addressed when cuboid hexahedral elements are used.

Extension of the developed methodology to the Poisson equation (with a forcing term) and time-dependent second order parabolic differential equation follows naturally, and is of more practical application since a lot of problems in real life involve externally-induced dynamic forces that alter equilibrium states from time to time (Azoury, 1992; Incropera et al., 2006). For this reason, the extension was carried out to solve the equation as first introduced in Section 3.1.2, but repeated below for ease of reference.

$$K\nabla^2 u = S\frac{\partial u}{\partial t} + f(x,y,z,t) \quad \text{in } \Omega \tag{2.1}$$

Where $\nabla = i\frac{\partial}{\partial x} + j\frac{\partial}{\partial y} + k\frac{\partial}{\partial z}$, $S\frac{\partial u}{\partial t}$ is the temporal derivative term,

and $f(x,y,z,t)$ is the forcing term.

We note that the left hand side of Equation (2.1) has been the subject of the work carried out in Chapter 3, and that it now remains to evaluate the terms on the right.

Three approaches can in general be followed. The first involves eliminating the temporal derivative by taking the Laplace transform of the governing equation so that it becomes an elliptic one. In that case a Poisson equation is solved. Many authors have followed this approach in Boundary element circles for 2D problems (Davies and Honnor, 2002; Davies and Crann, 2004; Moridis, 1992; Pozrikidis, 2006; Tanaka et al., 1991). The additional challenge of this approach is recovering the real solution from the Laplace transformed space.

The second approach involves treating the right hand side of Equation (2.1) as a forcing term so that it requires solving the Poisson equation. In that case the Green's function for the Laplacian operator is used to construct integrations. The latter is particularly numerically unattractive in boundary element circles, hence the use of dual reciprocity principles to transform the domain integrals to boundary ones (see Section 2.3.4). In Green element circles where the source node always lies in the element over which integration is done, the domain integration is straight forwardly carried out (Archer, 2005; Taigbenu, 1990). That does not preclude the use of dual reciprocity principles as has been carried out by Popov and Power (Popov and Power, 1996; Popov and Power, 1998; Popov and Power, 1999).

The third approach involves using the time-dependent fundamental solution to construct the integral equations. That fundamental solution is given as

$$u^*(r,t;r_i,\tau)=\frac{\exp\left(\dfrac{-(r-r_i)^2}{4K(t-\tau)}\right)}{\left(4\pi K(t-\tau)\right)^{m/2}}H(t-\tau) \tag{4.1}$$

in which $r_i=(x_i,y_i)$ is the source node, $r=(x,y)$ is the field node, m is the spatial dimension number, and $H(t)$ is the Heaviside function (Carslaw and Jaeger, 1959). The approach is more complicated than that of the Laplace operator and presents greater challenges in evaluating the integrals. It still requires evaluating domain integrals of the distribution of the potential at the initial time, and the forcing term. In 2D, this approach has been pursued by (Bozkaya and Tezer-Sezgin, 2008; Carrer and Mansur, 2002; Gatmiri and Van Nguyen, 2005; Taigbenu, 2001; Taigbenu, 2004; Young et al., 2004).

The second approach is followed in this work. The Green's function of the Laplacian operator has been used, implying that the boundary integrations from the previous chapter remain applicable.

On applying the fundamental solution to Equation (2.1) and applying the standard weighting procedure in the same fashion as was done in Equation (2.12), the following equation is the result.

$$\begin{aligned}&\int_{\Omega}[\nabla^2 u - S\frac{\partial u}{\partial t} - f(x,y,z,t)]u^* d\Omega \\ &= \int_{\Gamma_2}(q-\overline{q})u^* d\Gamma - \int_{\Gamma_1}(u-\overline{u})q^* d\Gamma\end{aligned} \qquad \textbf{(4.2)}$$

It is noted that what remains to be evaluated are the 3D domain integrals as given below.

$$\begin{aligned}I_1 &= \iiint f(x,y,z,t)\,u^* dxdydz \\ I_2 &= \iiint\left(S\frac{\partial u}{\partial t}\right)u^* dxdydz\end{aligned} \qquad \textbf{(4.3)}$$

I_1 accounts for the forcing term while I_2 for the temporal derivative term.

4.2 The Forcing Term

The forcing term could either arise from source/sink as in the case of operating wells or from a distributed input as in the case of recharge from precipitation into an aquifer. In the first case, the forcing term can be represented by making use of the Dirac delta function discussed earlier. In that case:

$$F_k \Delta_k = F_k \Delta(r - r_k) \tag{4.4}$$

where and F_k is the strength of the source k^{th} source/sink at $r_k \equiv (x_k, y_k, z_k)$

$$I_1 = \int_\Omega F_k \Delta(r - r_k) u^* d\Omega = F_k u_{ik}{}^* \tag{4.5}$$

where $u_{ik}{}^* = \dfrac{1}{4\pi(r_i - r_k)}$.

For the second case, the forcing term in the domain is expressed in terms of the nodal values of the function using 3D interpolation functions. Using the 3D interpolation functions ϕ_1, ϕ_2, ϕ_3, ϕ_4 ϕ_5, ϕ_6, ϕ_7, and ϕ_8 for the hexahedra element (Figure 4.1), we obtain the following result for the term $I_1 = \iiint f(x, y, z, t) u^* dxdydz$:

$$I_1 = \iiint [\phi_1 \quad \phi_2 \quad \phi_3 \quad \phi_4 \quad \phi_5 \quad \phi_6 \quad \phi_7 \quad \phi_8] u^* dxdydz \begin{Bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \end{Bmatrix} \tag{ 4.6}$$

Where F_i represents the nodal values of the forcing term, and have been taken out of the integral operator because they do not depend on the spatial variables.

The interpolation functions ϕ_1, ϕ_2, ϕ_3, ϕ_4 ϕ_5, ϕ_6, ϕ_7, and ϕ_8 for a linear hexahedra element with nodes labelled as given in Figure 4.1 and listed in Table 4.1.

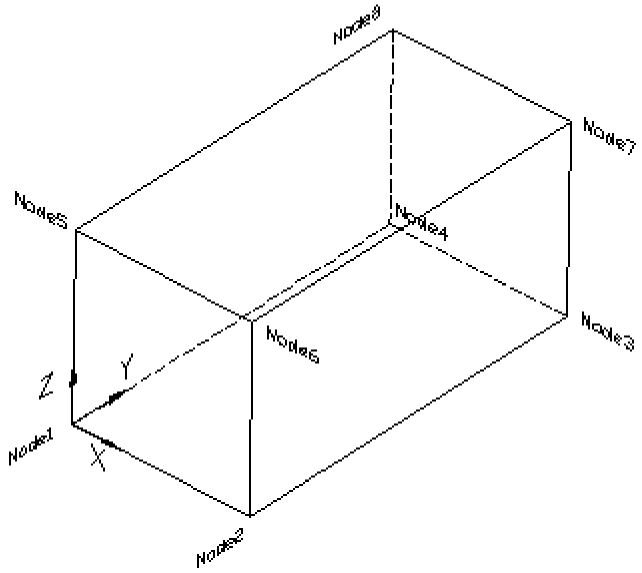
 Node8 Node7 Node5 Node4 Node6 Node3 Z Y Node1 X Node2	Coordinates Node1 (0,0,0) Node2 (a,0,0) Node3 (a,b,0) Node4(0,b,0) Node5 (0,0,c) Node6 (a,0,c) Node7 (a,b,c) Node8 (0,b,c)

Figure 4.1.Interpolation functions for a hexahedral element.

Table 4.1. Interpolation Functions

$\phi_1 = -\frac{(x-a)(y-b)(z-c)}{abc}$	$\phi_5 = \frac{(x-a)(y-b)(z)}{abc}$
$\phi_2 = \frac{(x)(y-b)(z-c)}{abc}$	$\phi_6 = -\frac{(x)(y-b)(z)}{abc}$
$\phi_3 = -\frac{(x)(y)(z-c)}{abc}$	$\phi_7 = \frac{(x)(y)(z)}{abc}$
$\phi_4 = \frac{(x-a)(y)(z-c)}{abc}$	$\phi_8 = -\frac{(x-a)(y)(z)}{abc}$

Since the fundamental solution is given by $u^* = \dfrac{1}{4\pi r}$, a generic expression for the term $\iiint \phi_i u^* dxdydz$ can be obtained by using the term $\phi_i u^* = \dfrac{(x-a)(y-b)(z-c)}{\left(x^2+y^2+z^2\right)^{1/2}}$ and allowing either a, b, or c in the numerator to assume a value of zero depending on the expression of the ϕ_i as given in Table 4.1 above. This results in the generic expression for $\iiint \phi_i u^* dxdydz$ as given by Equation (4.7)

$$\iiint \phi_i u^* dxdydz = \iiint \frac{xyz - cxy - bxz - ayz + bcx + acy + abz - abc}{\left(x^2+y^2+z^2\right)^{1/2}} dxdydz \tag{4.7}$$

Mathematica was then used to evaluate the integrations of each of the eight terms represented in Equation (4.7).

4.3 The Temporal Derivative Term

The temporal derivative can be evaluated in a manner similar to that for the distributed forcing term. Using the 3D interpolation functions ϕ_1, ϕ_2, ϕ_3, ϕ_4 ϕ_5, ϕ_6, ϕ_7, and ϕ_8 for the linear hexahedra element, we perform the following analysis to derive an expression of the term $I_2 = \iiint \left(S \dfrac{\partial u}{\partial t} \right) u^* dxdydz$ in Equation (4.3):

We have

$$u \approx \phi_j(x,y,z)\, u_j(t) \tag{4.8}$$

or

$$u \approx \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \end{bmatrix} \tag{4.9}$$

Therefore:

$$\frac{\partial u}{\partial t} \approx \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 & \phi_7 & \phi_8 \end{bmatrix} \begin{bmatrix} \frac{du_1}{dt} \\ \frac{du_2}{dt} \\ \frac{du_3}{dt} \\ \frac{du_4}{dt} \\ \frac{du_5}{dt} \\ \frac{du_6}{dt} \\ \frac{du_7}{dt} \\ \frac{du_8}{dt} \end{bmatrix}$$

$$\approx \phi_j \frac{du_j}{dt} \tag{4.10}$$

$$\begin{aligned} I_2 &= \iiint \left(S \frac{\partial u}{\partial t} \right) u^* dxdydz \\ &\approx S \frac{du_j}{dt} \iiint \phi_j u^* dxdydz \end{aligned} \qquad \textbf{(4.11)}$$

The distribution of u throughout the medium at initial time t^0 is known and our aim is to start with this known distribution and match forward to determine the distribution of u at the next time step. Once the distribution of u at this time step is known we can match forward to the next time step. In general, given the distribution of u at a previous time step t, our aim is to match forward to the current time step (t+1).

Representing $\frac{du_j}{dt}$ by a finite difference expression, ie $\frac{du_j}{dt} = \frac{u_j^{t+1} - u_j^t}{\Delta t}$, we have

$$I_2 = S \bullet \left(\iiint \phi_j u^* dxdydz \right) \bullet \left(\frac{u_j^{t+1} - u_j^t}{\Delta t} \right) \qquad \textbf{(4.12)}$$

Incorporating the forcing term and the temporal derivative term into Equation (3.6) results in:

$$ku^i + \sum_{m=1}^{N} \left(\int_{\Gamma_m} uq_i^* d\Gamma \right) - \iiint f(x,y,z,t) u_i^* dxdydz - S \iiint \phi_j u_i^* dxdydz \left(\frac{u_j^{t+1} - u_j^t}{\Delta t} \right) = \sum_{m=1}^{N} \left(\int_{\Gamma_m} q_i u^* d\Gamma \right) \qquad \textbf{(4.13)}$$

To simplify the expression, we adopt the following abbreviations for matrices in Equation (4.13) above.

$$H_{ij} = \sum_{m=1}^{N} \left(\int_{\Gamma_m} \phi_j q_i^{*} d\Gamma \right) \tag{4.14}$$

$$G_{ij} = \sum_{m=1}^{N} \left(\int_{\Gamma_m} \phi_j u_i^{*} d\Gamma \right)$$

$$W_{ij} = \iiint \phi_j u_i^{*} \, dxdydz$$

With these simplifications Equation (4.13) becomes

$$ku_i + H_{ij} u_j - W_{ij} F_j - SW_{ij} \left(\frac{u_j^{t+1} - u_j^{t}}{\Delta t} \right) = G_{ij} q_j \tag{4.15}$$

We utilize an implicit generalized time-differencing scheme to solve the equation. Explicit methods calculate the state of a system at the current time from its state at the previous time, while an implicit method finds it by solving an equation involving both the previous and current states of the system. Mathematically, if u^t is the previous system state and u^{t+1} is the current state (Δt is a small time step), then, for an explicit method

$$u^{t+1} = Fu^{t} \tag{4.16}$$

while for an implicit method one solves an equation of the form

$$G\left(u^{t}, u^{t+1}\right) = 0 \tag{4.17}$$

and requires matrix manipulations in order to evaluate u^{t+1} (Burden and Faires, 2001).

It is clear that implicit methods require extra computations, making them much harder to implement. Implicit methods are used because many problems arising in real life are stiff, for which the use of an explicit method requires impractically small time steps Δt to keep the error in the result bounded (achieve numerical stability). For such problems, to achieve a given level of accuracy, it takes much less computational time to use an implicit method with larger time steps, despite the additional computational effort from solving a matrix equation of the form (4.17) at each time step. This is the reason why an implicit scheme is used in this work.

If the interval between the current time step (t+1) and the previous time step t is denoted by Δt, equation (4.15) can be written at any point within the time interval for an implicit scheme. If u_i^{t+1}, q_i^{t+1} represent the state of the potential and flux at a node at the current time (t+1), and u_i^t, q_i^t represent the state at the previous time t, and α is the time weighting factor representing the fraction of Δt between t and t+1 where the equation is written, linear interpolation between (t+1) and t dictates that Equation (4.15) may be re-written as Equation (4.18) below.

$$\begin{aligned} &k\left[\alpha u_j^{t+1} + (1-\alpha)u_j^t\right] + H_{ij}\left[\alpha u_j^{t+1} + (1-\alpha)u_j^t\right] - \\ &W_{ij}\left[\alpha F_j^{t+1} + (1-\alpha)F_j^t\right] - SW_{ij}\left(\frac{u_j^{t+1} - u_j^t}{\Delta t}\right) = G_{ij}\left[\alpha q_j^{t+1} + (1-\alpha)q_j^t\right] \end{aligned} \qquad \textbf{(4.18)}$$

After gathering all the unknowns to the left and all the knowns to the right, equation (4.18) can be re-written as Equation (4.19) below.

$$\begin{aligned} &\left(k\alpha + H_{ij}\alpha - \frac{SW_{ij}}{\Delta t}\right)u_j^{t+1} - G_{ij}\alpha q_j^{t+1} = \\ &G_{ij}(1-\alpha)q_j^t - \left[k(1-\alpha) + H_{ij}(1-\alpha) + \frac{SW_{ij}}{\Delta t}\right]u_j^t + W_{ij}\left[\alpha F_j^{t+1} + (1-\alpha)F_j^t\right] \end{aligned} \qquad \textbf{(4.19)}$$

α takes a value between 0 and 1. $\alpha = 0$ results in the explicit scheme.

Equation (4.19) represents 3D Green element formulation for potential flow which takes into account the forcing term and the temporal derivative term

4.4 Implementation of the Time-Dependent Green Element Formulation

Figure 4.2 below demonstrates how the left hand and the right hand sides of Equation (4.19) were incorporated into the matrices for the solution of the potentials at each time step. The algorithm begins with the original H and G matrices that were developed for the steady state situations. This applies to both the left hand and right hand sides of the equation.

In incorporating the time factor into the Green element method calculations, we assume a quasi-steady state model for performing dynamic simulations. A quasi-steady state model is defined as a sequence of steady-state profiles (Walsum and Groenendijk, 2008). In our case this implies that we assume that at each time step the potentials and the fluxes throughout the domain satisfy the governing partial differential equation in such a way that instantaneous potential and flux values would represent the steady-state values if the prevailing boundary conditions at that instance were to be held constant.

In this work we have assumed that the conditions of steady-state at each time step are applicable. This assumption has come into question in recent years (Seibert et al., 2003). In his work, Seibert tested this assumption with an analysis of detailed groundwater level data along two opposing hillslopes along a stream reach in a Swedish till catchment at Svartberget. Groundwater levels in areas close to the stream followed the dynamics of the runoff. The correlation between groundwater level and runoff decreased markedly for wells further than approximately 40 m from the stream. The levels were often independent of streamflow: Upslope area groundwater could be rising when riparian groundwater and runoff were falling, and vice versa. There was a high degree of correlation between groundwater levels at similar distances from the stream. Despite the widespread acceptance of the steady state assumption previously in this and other study catchments, Seibert’s study shows that it is not valid for the investigated hillslope site.

4.5 Matrix Manipulations

In this work, matrices have been used for the storage of the information required for the analyses. Data such as the element types, node lists, element connectivity, element surfaces, and potential and flux variable lists and values are all stored in matrices.

A feature of numerical computation based on discretization elements is the generation of a large number of simultaneous equations, and matrix solvers are required to solve for the unknowns. In this work, the need to solve simultaneous equations arises in two instances. These instances are:

- the determination of the collocation point and the plane on which surface integration is being done,
- the calculation of the potentials and the fluxes using the potential and flux coefficient matrices within steady and transient simulation situations.

These applications of matrix algebra are straightforward and have been implemented with matrix manipulations using a simple matrix solving routine that utilises the Gauss elimination method and provides for interchanging rows when a zero diagonal coefficient is encountered. The routine is called several times as the need arises to solve a particular set of linear equations.

A matrix solver that solves a system of linear equations in the manner that has been implemented in this work is called a direct solver. Adopting a commercial direct solver for our work would yield some benefit, because such solvers normally employ a sparse matrix storage technology. Since the potential and flux coefficient matrices that are generated by our GEM routines are very sparse, the data access efficiency would increase significantly. Commercial direct solvers also come with significant implementation specialties aimed at taking advantage of the structure of the matrices and of the computer architecture.

Iterative solvers use a different approach to solving large systems of linear equations. They start with an initial guess which may be a zero vector and go through an iterative process to update the solution vector using the system matrix and a pre-conditioner matrix to achieve solution converge. A convergence criterion is used to determine whether a solution has been reached. If a pre-determined maximum number of iterations is reached without obtaining the desired accuracy, the solver exits. For a discussion on the preconditioned conjugate method see Golub and Van Leon (1989).

The decision of whether to use a direct or an iterative solver in a given situation is not an easy one. The decision depends heavily on the analysis type and model characteristics. Ketkar (1993) has shown that in thermal analysis work, for stiff thermal problems the direct solution technique converges much faster than the iterative one and for problems with severe nonlinearities and/or large number of nodes the iterative approach is more efficient.

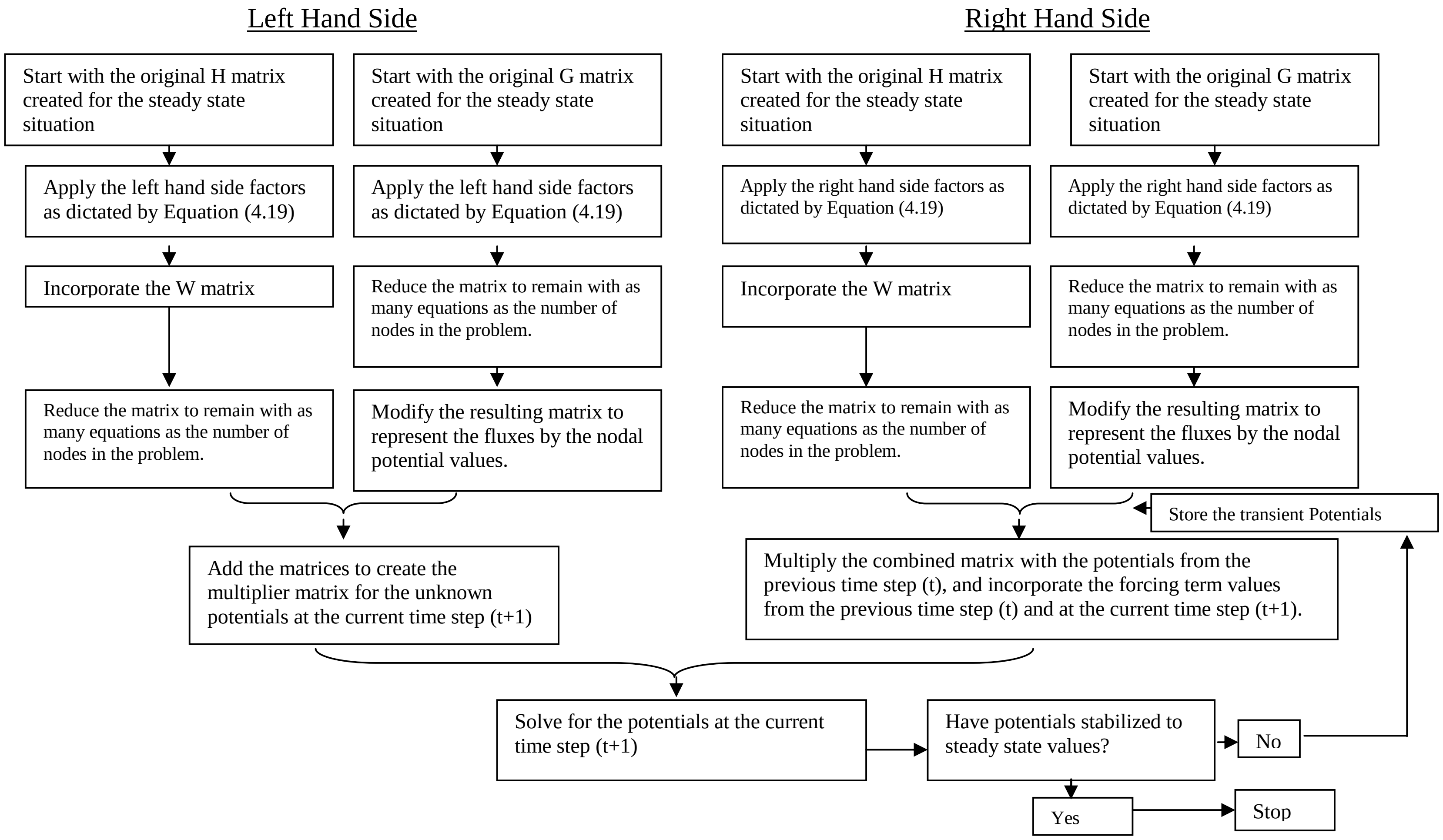

Left Hand Side
Right Hand Side
Start with the original H matrix created for the steady state situation
Start with the original G matrix created for the steady state situation
Start with the original H matrix created for the steady state situation
Start with the original G matrix created for the steady state situation
Apply the left hand side factors as dictated by Equation (4.19)
Apply the left hand side factors as dictated by Equation (4.19)
Apply the right hand side factors as dictated by Equation (4.19)
Apply the right hand side factors as dictated by Equation (4.19)
Incorporate the W matrix
Reduce the matrix to remain with as many equations as the number of nodes in the problem.
Incorporate the W matrix
Reduce the matrix to remain with as many equations as the number of nodes in the problem.
Reduce the matrix to remain with as many equations as the number of nodes in the problem.
Modify the resulting matrix to represent the fluxes by the nodal potential values.
Reduce the matrix to remain with as many equations as the number of nodes in the problem.
Modify the resulting matrix to represent the fluxes by the nodal potential values.
Store the transient Potentials
Add the matrices to create the multiplier matrix for the unknown potentials at the current time step (t+1)
Multiply the combined matrix with the potentials from the previous time step (t), and incorporate the forcing term values from the previous time step (t) and at the current time step (t+1).
Solve for the potentials at the current time step (t+1)
Have potentials stabilized to steady state values?
No
Yes
Stop

Figure 4.2. Implementation of Equation (4.19)

5. RESULTS AND DISCUSSION

Numerical examples serve to demonstrate the computational robustness of the 3D GEM that has been formulated in the previous chapters. As a start, the examples have been constructed so they have analytical solutions with which accuracy comparison can be made. These examples are in domains that have regular geometries. Additional benchmarking is made against FEMWATER finite element package on the basis of solution accuracy, but no attempt is made to compare the computational efficiencies of FEMWATER and our GEM because of their different coding platforms.

5.1 Steady-State Examples

5.1.1 Example 1 – 1D steady flow with potential drop

To test the 3D GEM model herein developed, a three dimensional 5mx10mx5m cuboid figure (Figure 5.1) is used and potentials are applied on the opposite ends of the 10m length, while all the other boundaries are specified as zero flux boundaries. This example is one in which 1D steady flow over a 10m distance occurs because of a potential drop from 8m at one end to 4m at the other. It is assumed the hydraulic conductivity is unity for simplicity and the exact solution is a linear distribution of potential with uniform velocity of 0.4 m/s in the direction of the potential drop. The domain is discretised with a coarse mesh of twelve elements each of size $2.5\,\text{m}$ by 5m by $1.67\,\text{m}$. It was observed that the Green element method replicated the expected result almost exactly, while the commercial finite element package, FEMWATER, produced results with relative error ranging from 0.09% to 1.84%. This illustrates the high accuracy that can be expected of the Green element method.

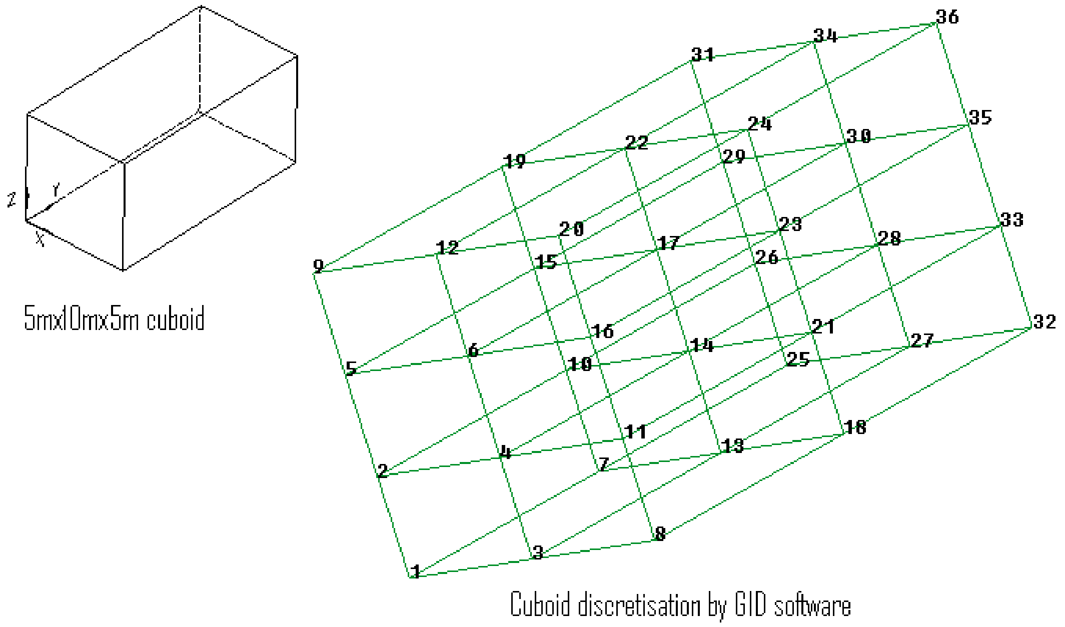


Figure 5.1 Computational domain of 1D steady problem.

Table 5.1. Comparison of the solutions of Example 1 by FEMWATER and GEM.

Node ID	X	Y	Z	Velocity, EXACT	Velocity, FEMWATER	Error, FEMWATER	Velocity, GEM	Error, GEM
1	2	3	5	0.400000000	0.399637000	0.09%	0.400000180	0.000045%
8	7	3	5	0.400000000	0.399632000	0.09%	0.400000006	0.000001%
32	7	13	5	0.400000000	0.400369000	0.09%	0.400000006	0.000001%
25	2	13	5	0.400000000	0.400364000	0.09%	0.400000006	0.000001%
20	7	3	10	0.400000000	0.399250000	0.19%	0.400000006	0.000001%
36	7	13	10	0.400000000	0.392658000	1.84%	0.400000006	0.000001%
31	2	13	10	0.400000000	0.396478000	0.88%	0.400000006	0.000001%
9	2	3	10	0.400000000	0.399521000	0.12%	0.399999917	0.000021%

5.1.2 Example 2 – 2D steady heat conduction

Next we considered the problem of determining the steady-state heat distribution in a thin square metal plate with dimensions 0.5mx0.5m (Burden and Faires, 2001). Two adjacent boundaries (left and bottom) are held at 0^oC, and the temperature on the other two boundaries increases linearly from 0°C at one corner to 100°C at the top right corner where the sides meet. It is assumed that the thermal diffusivity K for the medium has a unit value. If we place the sides with zero boundary conditions along the x- and y- axes, the problem is expressed as

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0 \qquad \textbf{(5.1)}$$

with the boundary conditions $u(0,y)=0, u(x,0)=0, u(x,0.5)=200x, u(0.5,y)=200y$

The analytical solution to the above problem is known, and is $u=400xy$.

Since the GEM method that was developed is for 3D problems, it was necessary to specify a dimension in the z-direction (Figure 5.2). A dimension of 0.5m was specified in the z-direction, while prescribed boundary conditions were adopted on the x=0, y=0, x=0.5 and y=0.5 planes. The z=0 and z=0.5 planes were kept as no-flux boundaries. A grid with 4x4x4 cells in the x, y, and z direction was then created (64 elements).

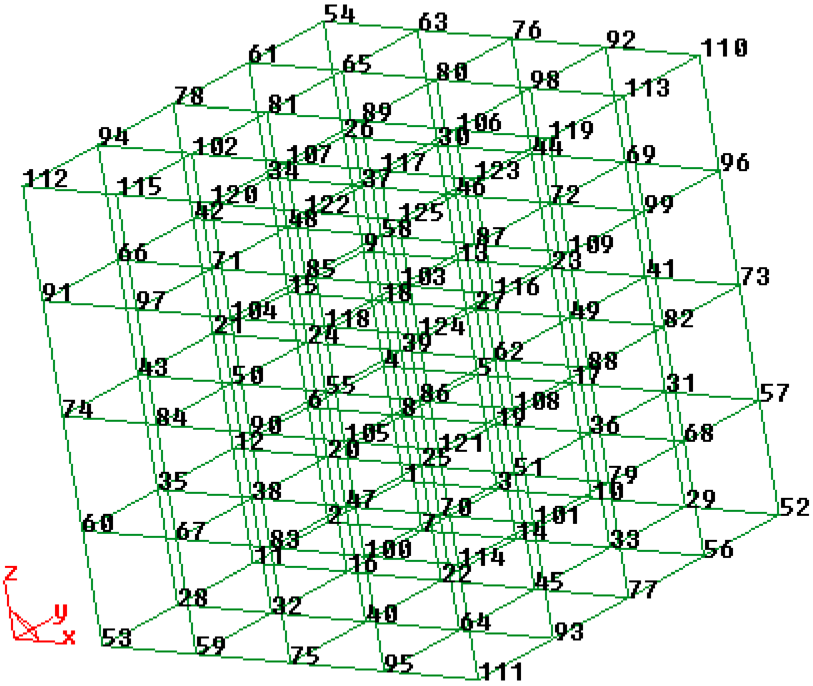


Figure 5.2 Discretisation of a 0.5mx0.5mx0.5m metallic cube

The heat fluxes in the x and y directions are obtained by differentiating the exact solution.

$$\frac{\partial u}{\partial x} = 400y$$
$$\frac{\partial u}{\partial y} = 400x$$
$$\frac{\partial u}{\partial z} = 0 \tag{5.2}$$

Table 5.2. Exact solution to Example 2.

Node	x	y	z	u	du/dx	du/dy	du/dz	Magnitude
7	0.125	0.375	0	18.8	150	50	0	158.1139
14	0.250	0.375	0	37.5	150	100	0	180.2776
33	0.375	0.375	0	56.3	150	150	0	212.132
16	0.125	0.250	0	12.5	100	50	0	111.8034
22	0.250	0.250	0	25	100	100	0	141.4214
45	0.375	0.250	0	37.5	100	150	0	180.2776
32	0.125	0.125	0	6.25	50	50	0	70.71068
40	0.250	0.125	0	12.5	50	100	0	111.8034
64	0.375	0.125	0	18.8	50	150	0	158.1139

The performance of the GEM method is given in Table 5.3 below for all the internal nodes on the z=0 plane. It is evident that the GEM reproduces the exact solution. The values for potentials and fluxes are dimensionless, in line with the governing equation which is dimensionless.

Table 5.3. Performance of the GEM on Example 2.

Node	x	y	z	u	du/dx	du/dy	du/dz	Magnitude
7	0.125	0.375	0	18.8	150	50	0	158.1139
14	0.250	0.375	0	37.5	150	100	0	180.2776
33	0.375	0.375	0	56.3	150	150	0	212.132
16	0.125	0.250	0	12.5	100	50	0	111.8034
22	0.250	0.250	0	25	100	100	0	141.4214
45	0.375	0.250	0	37.5	100	150	0	180.2776
32	0.125	0.125	0	6.25	50	50	0	70.71068
40	0.250	0.125	0	12.5	50	100	0	111.8034
64	0.375	0.125	0	18.8	50	150	0	158.1139

5.1.3 Example 3 – 3D potential flow example

In this third example, a known analytic solution for the potential of the Laplace equation is used to generate the boundary conditions for the numerical model which then calculates the potential and velocity at some specified grid points. Although this example is better analysed in spherical coordinates, it is nonetheless simulated in Cartesian coordinates with GEM and FEMWATER.

Given a scalar field *u*, the Laplace equation in Cartesian coordinates is

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial^2 y}+\frac{\partial^2 u}{\partial z^2}=0 \tag{5.3}$$

The known analytic solution is (Kreysig, 1988)

$$u=\frac{c}{r}+K \tag{5.4}$$

where c and k are constants and $r=\sqrt{x^2+y^2+z^2}$.

The fluxes in the x, y, and z directions are easily derived and given by Equation (5.5).

$$\begin{aligned}\frac{\partial u}{\partial x}&=\frac{-cx}{\left(x^2+y^2+z^2\right)^{3/2}}\\ \frac{\partial u}{\partial y}&=\frac{-cy}{\left(x^2+y^2+z^2\right)^{3/2}}\\ \frac{\partial u}{\partial z}&=\frac{-cz}{\left(x^2+y^2+z^2\right)^{3/2}}\end{aligned} \tag{5.5}$$

To avoid the singularity presented by the analytical solution when any of the nodes is at the origin in the Cartesian coordinate system, the element was displaced by 2m in the x-direction, 3m in the y and 5m in the z-direction. The domain is discretised uniformly into 16 elements (element size 2.5mx2.5mx2.5m) in both GEM and FEMWATER (Figure 5.3).

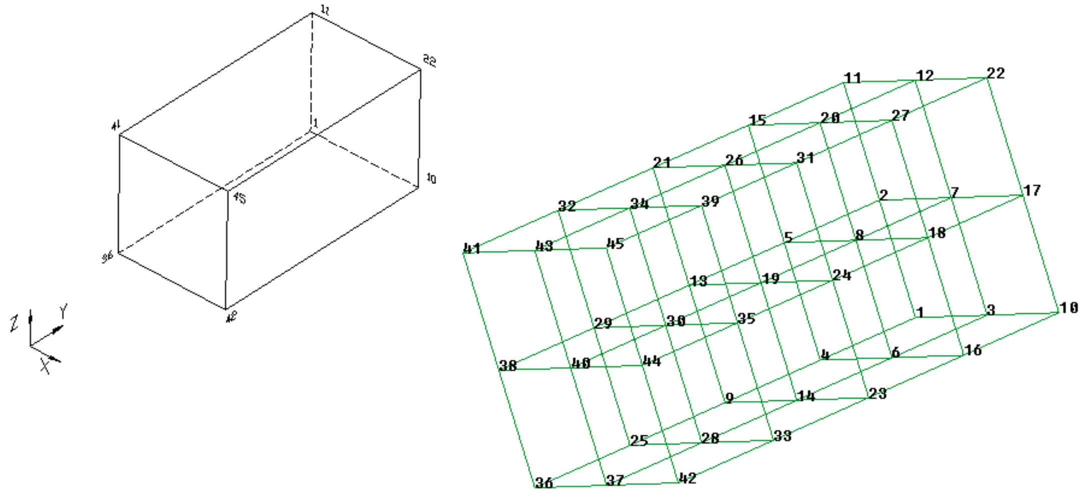


Figure 5.3 Domain discretization for Example 3 (3D potential flow problem).

Taking values of c and K to be 558.1 and 24.2 respectively, the complete solution is tabulated in Table 5.4 below.

Table 5.4 Exact solution of Example 3.

External Nodes	x	y	z	r	u	$\partial u/\partial x$	$\partial u/\partial y$	$\partial u/\partial z$	Magnitude
36	2	3	5	6.16441	114.7323	-4.7649	-7.1474	-11.9123	14.6864
42	7	3	5	9.11043	85.4568	-5.1663	-2.2141	-3.6902	6.7239
10	7	13	5	15.58846	60.0002	-1.0313	-1.9153	-0.7366	2.2966
1	2	13	5	14.07125	63.8604	-0.4006	-2.6040	-1.0015	2.8186
45	7	3	10	12.56981	68.5979	-1.9670	-0.8430	-2.8101	3.5322
22	7	13	10	17.83255	55.4949	-0.6889	-1.2794	-0.9841	1.7550
11	2	13	10	16.52271	57.9758	-0.2474	-1.6084	-1.2372	2.0443
41	2	3	10	10.63015	76.6992	-0.9292	-1.3938	-4.6460	4.9388

Internal Nodes	x	y	z	r	u
8	4.5	10.5	7.5	13.66565	65.0375
19	4.5	8	7.5	11.85327	71.2818
30	4.5	5.5	7.5	10.33199	78.2143

The Green element method and FEMWATER were used to simulate the flow velocities and internal nodal potentials in a 5m X 10m X 5m cuboid similar to that of Figure 5.3. The *Dirichlet* conditions derived from the analytical solution were specified at the boundary nodes of the finite Element mesh, and the GEM and FEMWATER solutions for the velocity at the external nodes and internal node potentials were examined. The results are presented in Table 5.5. It is observed that for the same discretization of mesh sizes of 2.5mx2.5mx2.5m (16 elements) the Green element method produced an average error of 10.5% while FEMWATER produced an average error of 13.9%. To find out how fine the FEMWATER grids would need to be to replicate the results of the Green element method, we performed two more runs with FEMWATER, the second with mesh sizes of 2.5mx2mx2.5m (20 elements) and the third with 2.5mx2mx1.7m (30 elements). At thirty elements the results of the finite element method matched those of the Green element method at 16 elements. Thus in this case the Green element method was able to replicate the finite element method with only 47% of the discretization.

Table 5.5. GEM and FEMWATER solutions for velocity at external nodes of Example 3.

	EXACT	GEM		FEM, Mesh 1		FEM, Mesh 2		FEM, Mesh 3	
Node ID	Velocity	Velocity	Error	Velocity	Error	Velocity	Error	Velocity	Error
36	14.6864	13.2293	9.9%	10.7951	26.5%	9.9066	32.55%	10.4997	28.5%
42	6.7239	7.3320	9.0%	6.7942	1.0%	6.6305	1.39%	6.8716	2.2%
10	2.2966	2.4678	7.5%	2.3875	4.0%	2.3468	2.18%	2.3775	3.5%
1	2.8186	3.3095	17.4%	3.1438	11.5%	3.0499	8.21%	3.1051	10.2%
45	3.5322	3.8035	7.7%	4.2570	20.5%	4.2317	19.80%	4.0132	13.6%
22	1.7550	1.7622	0.4%	1.9517	11.2%	1.9271	9.81%	1.8943	7.9%
11	2.0443	2.1673	6.0%	2.4090	17.8%	2.3597	15.43%	2.3036	12.7%
41	4.9388	6.2293	26.1%	5.8595	18.6%	5.6402	14.20%	5.2299	5.9%
	Average Error		10.5%		13.9%		12.95%		10.6%

Table 5.6. GEM and FEMWATER solutions for potential at some internal nodes of Example 3.

	EXACT	GEM		FEM, Mesh 1	
Internal Nodes	Potential	Potential	Error	Potential	Error
8	65.0375	65.25383	0.33%	60.59915	6.82%
19	71.2818	71.71393	0.61%	67.79269	4.89%
30	78.2143	78.66682	0.58%	75.9464	2.90%

Comparison of the GEM and FEMWATER solutions for potentials at internal nodes is done only with the 16-element discretization; these internal nodes did not correspond to grid points in the 20-element and 30-element discretizations. The calculated potentials at the internal nodes by GEM and FEMWATER are presented in Table 5.6. GEM shows higher accuracy than FEMWATER.

Graphical representation of the solutions for the flux and potential by the three methods is not easy to achieve in this case because the coordinate systems take up the three dimensions and the computed potential or flux becomes an item in the fourth dimension. However, to get a feel of the solution, graphical representations of the solutions at the z-axis elevation of 5.0 were made and the results for the potential are shown in Figure 5.4.

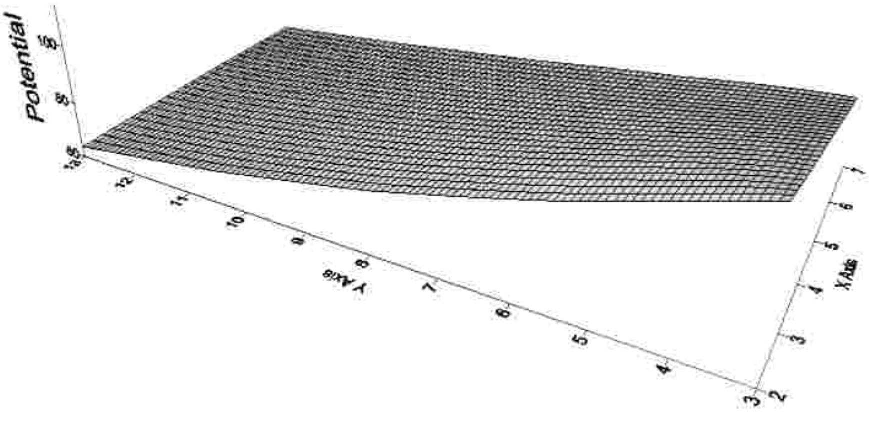
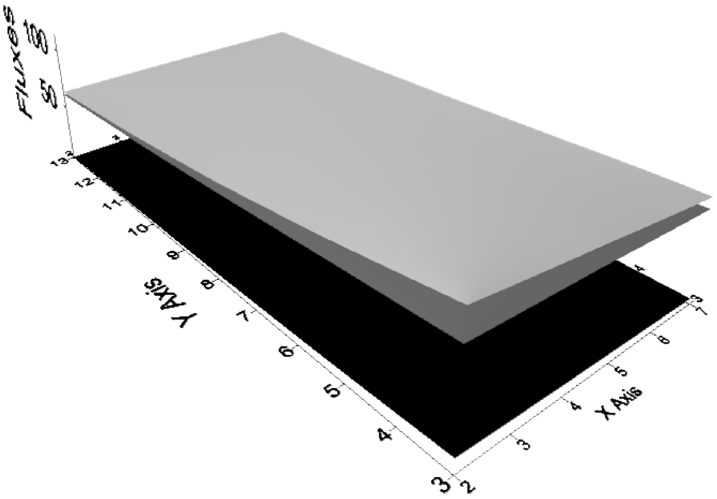


Figure 5.4. Plot of potentials at the elevation of $z = 5.0$ for Example 3.

In Figure 5.4 it is observed that the results generated by the three methods, namely the exact solution, the GEM solution, and the FEMWATER solutions are indistinguishable from each other for the potentials computed. A plot was also made of the fluxes computed by the three methods and presented in Figure 5.5.



Bottom layer=Exact, Middle layer=GEM, Top layer=FEMWATER.

Figure 5.5. Plot of fluxes at elevation of $z = 5.0$ for Example 3.

It is observed that the GEM produces results that are similar to those generated by FEMWATER, although the GEM does perform slightly better.

5.1.4 Example 4 – 2D Poisson equation example

A 2D potential flow problem that is governed by the Poisson equation is used to validate the 3D GEM model. The flow differential equation is

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 4 \tag{5.6}$$

in the domain $0 < x < 1,\ 0 < y < 2$. The boundary conditions are:

$$u(x,0) = x^2, u(x,2) = (x-2)^2,\ 0 \le x \le 1 \qquad \textbf{(5.7)}$$
$$u(0,y) = y^2,\ u(1,y) = (y-1)^2,\ 0 \le y \le 2$$

For this differential equation, the exact solution is known and is equal to $u(x,y,z) = (x-y)^2$ (Burden and Faires, 2001). The example is numerically solved in 3D by specifying a z-dimension of unit length. The z=0 and z=1 planes have zero flux prescribed on them, while the specified boundary conditions are appropriately specified on other external planes.

The domain was uniformly discretised into 64 elements each of size 0.25x0.50x0.25 as shown in Figure 5.6 below.

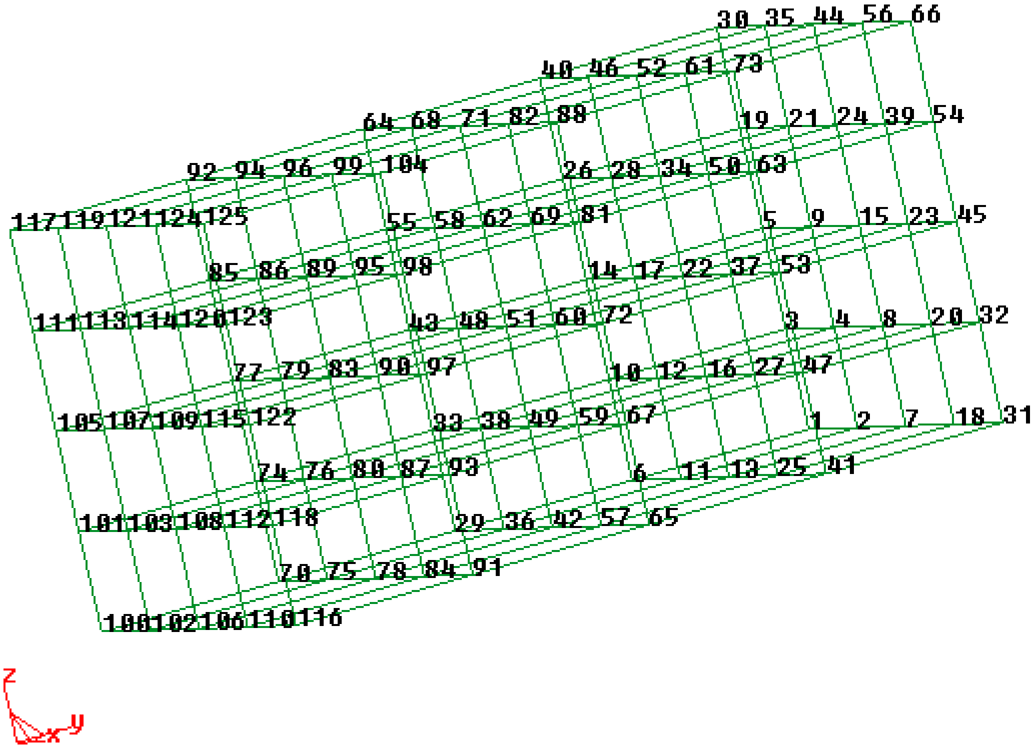


Figure 5.6. Domain discretization for Example 4 (2D Poisson equation).

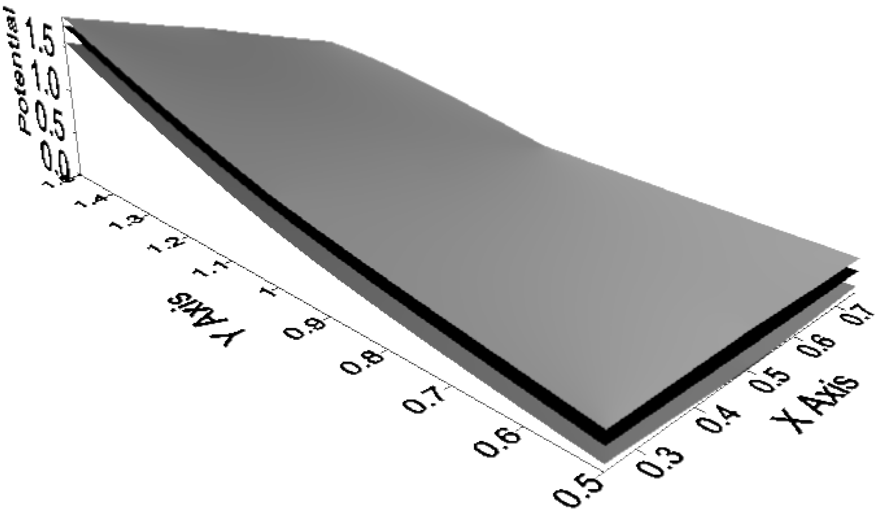
The following were the comparisons between the computations from the three methods for the potential at some selected internal nodes.

Table 5.7. Comparisons between the computations of the potential from exact solution, GEM, and FEMWATER at some selected internal nodes.

						FEMWATER		GEM	
Node ID	X	Y	Z		Exact	Value	Error	Value	Error
12	0.25	1.50	0.25		1.5625	1.843985	18%	1.744653	12%
16	0.50	1.50	0.25		1	1.369952	37%	1.239443	24%
17	0.25	1.50	0.50		1.5625	1.84465	18%	1.744676	12%
22	0.50	1.50	0.50		1	1.371357	37%	1.239496	24%
27	0.75	1.50	0.25		0.5625	0.844555	50%	0.692075	23%
28	0.25	1.50	0.75		1.5625	1.84547	18%	1.744669	12%
34	0.50	1.50	0.75		1	1.373688	37%	1.239491	24%
37	0.75	1.50	0.50		0.5625	0.847233	51%	0.692075	23%
38	0.25	1.00	0.25		0.5625	0.895536	59%	0.777557	38%
48	0.25	1.00	0.50		0.5625	0.896619	59%	0.777546	38%
49	0.50	1.00	0.25		0.25	0.691167	176%	0.512837	105%
50	0.75	1.50	0.75		0.5625	0.851287	51%	0.692087	23%
51	0.50	1.00	0.50		0.25	0.693214	177%	0.512809	105%

It is observed that at all nodes the Green element method performed better than the finite element method.

Listed in Table 5.8 below are the comparisons between the computations from the three methods for the flux at the same internal nodes as in Table 5.7 above, and listed in Table 5.9 are the comparisons between computations at some external nodes. A graphical representation is given below to illustrate the results of the computations. As already pointed out, graphical representation of the solutions by the three methods is not easy to achieve for three dimensional analysis because the coordinate systems take up the three dimensions and the computed potential or flux becomes an item in the fourth dimension. To develop a feel of the computation results, graphical representations are made at one elevation, in this case at z=0.25.



Bottom layer=Exact, Middle layer=GEM, Top layer=FEMWATER.

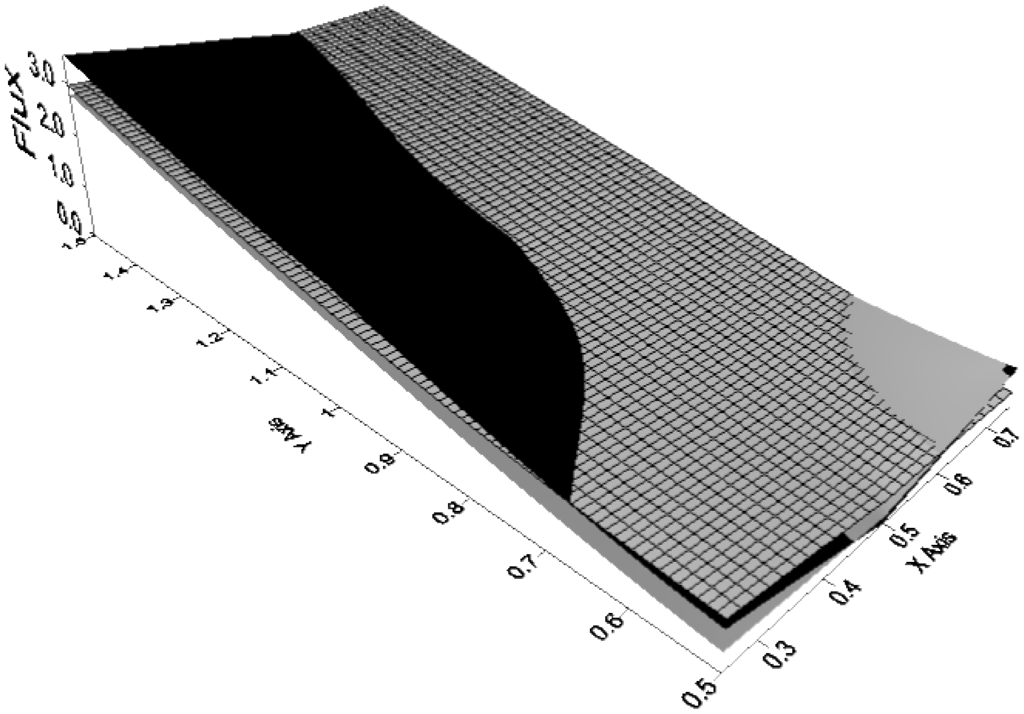
Figure 5.7. Plot of potentials at the elevation of z=0.25 for Example 4.

Table 5.8 Comparisons between the flux computations from the three methods at the same internal nodes as in Table 5.7

	Exact				FEMWATER				GEM			
Node ID	vx	vy	vz	Mag	vx	vy	vz	Mag	vx	vy	vz	Mag
12	-2.5	2.5	0	3.535534	1.93437	-2.25413	-0.00191	2.970336	2.02139	-1.93419	-0.00009	2.7977
16	-2	2	0	2.828427	2.070375	-1.59461	-0.00375	2.613283	2.02084	-1.45321	-0.00021	2.4891
17	-2.5	2.5	0	3.535534	1.931897	-2.25318	-0.00351	2.968006	2.0213	-1.93426	0.00003	2.7977
22	-2	2	0	2.828427	2.067264	-1.59299	-0.00697	2.609836	2.02072	-1.45337	0.00002	2.4891
27	-1.5	1.5	0	2.12132	2.21534	-1.15354	-0.00439	2.49768	2.18947	-0.76868	0	2.3205
28	-2.5	2.5	0	3.535534	1.928443	-2.25172	-0.00334	2.964649	2.02132	-1.93423	0.00012	2.7977
34	-2	2	0	2.828427	2.062267	-1.59026	-0.00699	2.604212	2.02071	-1.45337	0.00003	2.4891
37	-1.5	1.5	0	2.12132	2.21433	-1.14974	-0.00833	2.495041	2.18968	-0.76873	-0.00005	2.3207
38	-1.5	1.5	0	2.12132	0.665098	-1.52379	-0.00254	1.662618	0.88977	-1.13047	0.00004	1.4386
48	-1.5	1.5	0	2.12132	0.662094	-1.5233	-0.00497	1.660974	0.88982	-1.13046	-0.00003	1.4387
49	-1	1	0	1.414214	0.998358	-1.0006	-0.00444	1.413485	1.05888	-0.61806	0.00011	1.2261
50	-1.5	1.5	0	2.12132	2.205499	-1.14232	-0.00814	2.483785	2.18962	-0.76879	-0.00021	2.3207
51	-1	1	0	1.414214	0.995108	-1.00088	-0.00815	1.411406	1.05895	-0.61799	0	1.2261

Table 5.9 Comparisons between the flux computations from the three methods at some external nodes

	Exact				FEMWATER				GEM			
Node ID	vx	vy	vz	Mag	vx	vy	vz	Mag	vx	vy	vz	Mag
11	-2.5	2.5	0	3.535534	1.935228	-2.25442	0	2.971114	2.2006	-2.00603	0	2.977721
13	-2	2	0	2.828427	2.071382	-1.59519	0	2.614432	2.17708	-1.53536	0	2.664024
36	-1.5	1.5	0	2.12132	0.666071	-1.52383	0	1.663042	1.21267	-1.11086	0	1.644557
42	-1	1	0	1.414214	0.999354	-1.0003	0	1.413969	1.23574	-0.56881	0	1.360366
46	-2.5	2.5	0	3.535534	1.92646	-2.25082	0	2.962674	0.22844	-2.16983	0	2.181824
52	-2	2	0	2.828427	2.058092	-1.58793	0	2.599474	1.68192	-1.962	0	2.584238
57	-0.5	0.5	0	0.707107	1.3338	-0.476497	0	1.416359	0.8128	-0.13446	0	0.823847
61	-1.5	1.5	0	2.12132	2.192552	-1.13506	0	2.468936	-0.85635	-2.60431	0	2.741488
68	-1.5	1.5	0	2.12132	0.652808	-1.51474	0	1.649423	-0.4319	-1.75669	0	1.809006
71	-1	1	0	1.414214	0.97915	-0.992006	0	1.393847	1.26625	-1.05741	0	1.649695
75	-0.5	0.5	0	0.707107	-0.21432	-0.844958	0	0.871715	0.43438	-0.15781	0	0.462158
78	0	0	0	0	-0.07199	-0.404772	0	0.411124	0.15164	0.29301	0	0.329925
82	-0.5	0.5	0	0.707107	1.287295	-0.469022	0	1.370077	0.42827	-1.6284	0	1.683776
84	0.5	-0.5	0	0.707107	0.064188	0.254087	0	0.262069	-0.05589	0.89007	0	0.891819
94	-0.5	0.5	0	0.707107	-0.20528	-0.810477	0	0.836069	0.08149	-0.33426	0	0.344046
96	0	0	0	0	-0.06525	-0.39325	0	0.398626	-0.13232	-0.02542	0	0.13474
99	0.5	-0.5	0	0.707107	0.059989	0.237861	0	0.245309	1.57025	1.38471	0	2.093589



Black=Exact, Grey=GEM, Hatched=FEMWATER

Figure 5.8. Plot of fluxes at elevation of $z = 0.25$

5.2 Time-Dependent Examples

The performance of the Green element model is assessed for transient potential flow examples. The first example, for which no exact numerical solution is available, is the transient simulation of the problem earlier solved for the steady state in Section 5.1.4, and the last two numerical examples, for which analytical solutions are available as provided by Carslaw and Jaeger (1959), address the problem of transient linear diffusion in a bar of unit length.

5.2.1 Example 1 - Transient decay to Poisson equation

Our first example addresses the problem that was solved for the steady state situation in Section 5.1.4. The problem was set at an initial potential of 1.5 for all the internal nodes, while the boundary conditions were maintained as in Section 5.1.4. In this case the analytical solution for the transient variation of the nodal unknowns is not available, so it is assessed whether the transient solutions from GEM and FEMWATER converged to their steady state results. The models were run with a time weighting factor of α=0.5 and a time step of Δt=0.25, and the results were as tabulated in Table 5.10.

As is expected, the solution does converge to the steady state values of the nodal potentials. The model was also run with FEMWATER as a check, and the results are listed in the same table.

Table 5.10. Comparisons between transient potentials fluxes at Node 17 and Node 28 as computed by GEM and FEMWATER.

Node 17

		Potential		Flux	
Step No.	Time	GEM	FEMWATER	GEM	FEMWATER
0	0	1.5	1.5	2.166191	2.298097
1	0.25	1.745856	1.845869	2.786336	2.956004
2	0.5	1.744749	1.844698	2.797471	2.967817
3	0.75	1.74471	1.844657	2.797612	2.967967
4	1	1.74469	1.844636	2.797667	2.968025
5	1.25	1.744682	1.844627	2.797688	2.968047
6	1.5	1.744678	1.844623	2.797696	2.968056
7	1.75	1.744677	1.844622	2.797699	2.968059
8	2	1.744676	1.844621	2.7977	2.96806
9	2.25	1.744676	1.844621	2.7977	2.96806

Node 28

		Potential		Flux	
Step No.	Time	GEM	FEMWATER	GEM	FEMWATER
0	0	1.5	1.5	2.168639	2.298097
1	0.25	1.745816	1.846651	2.786366	2.952699
2	0.5	1.744738	1.845511	2.797476	2.964473
3	0.75	1.744702	1.845473	2.797611	2.964616
4	1	1.744682	1.845452	2.797666	2.964674
5	1.25	1.744675	1.845444	2.797687	2.964696
6	1.5	1.744671	1.84544	2.797695	2.964705
7	1.75	1.74467	1.845439	2.797698	2.964708
8	2	1.744669	1.845438	2.797699	2.964709
9	2.25	1.744669	1.845438	2.7977	2.96471

An attempt was made to model the effect of varying the time weighting factor α and the time step size Δt on the convergence characteristics of the GEM. There was no noticeable change in the manner in which the numerical results converged to their steady state values. This example does not demonstrate the accuracy of transient solutions of the GEM; it only serves to show that the transient simulation does replicate the solutions at steady state.

5.2.2 Example 2 - Transient diffusion with Dirichlet conditions

Our second transient example is one of heat conduction in a bar of unit length in which temperatures at both ends are maintained at certain values (steady Dirichlet boundary conditions). The problem is described by

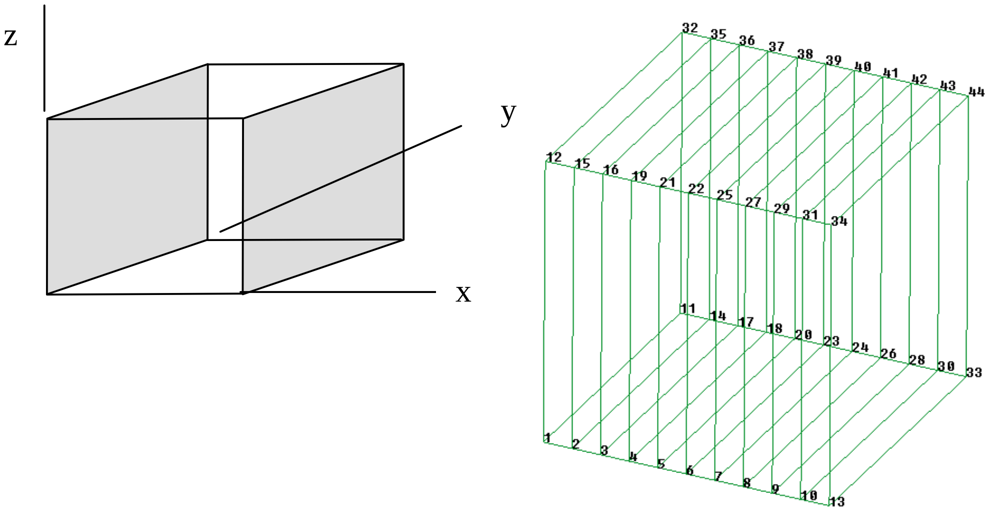
$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \tag{5.8}$$

The boundary conditions are $u(0,t) = 1, \quad u(1,t) = 0, \quad t > 0$ and the condition at $t = 0$ is $u(x,0) = 0$.

The exact solution is derived by method of Laplace transform, and is given by

$$u(x,t) = 1 - x + \frac{2}{\pi}\sum_{n=1}^{\infty}\frac{(-1)^n}{n}\exp\left(-n^2\pi^2 t\right)\sin\left[n\pi(1-x)\right] \qquad \textbf{(5.9)}$$

Our Green element method has been formulated to solve problems in 3D domains, but the problem with the known solution as outlined above is in one dimension. To use our model to simulate the one-dimensional flow, two opposite faces of the domain (shaded faces in Figure 5.9) have prescribed potentials, while the other four faces are described as zero-flux boundaries.



Faces x=0 and x=1 have prescribed fluxes. All other faces are zero-flux faces

Figure 5.9 Boundary value description to simulate one-dimensional flow, and domain discretization.

Domain discretization was done with GID software, and the node labelling is as shown in Figure 5.9. The exact, GEM, and FEMWATER solutions for the potential (heat) are as listed Table 5.11 and as shown in Figure 5.10 and Figure 5.11. At t=0 the potential (temperature) will be zero everywhere in the domain except at *x*=0 where *u*=1 as specified by the boundary conditions, and will begin to rise steadily with time. Figure 5.10 shows that GEM reflects this transient heat conduction better than FEM. The final state will be that the temperature will vary linearly from *u*=1 at *x*=0 to *u*=0 at *x*=1.

Table 5.11 Computed potentials at selected time values, Dirichlet conditions.

		Exact	GEM		FEM	
Time	x	u	u	ERROR	u	ERROR
t=0.1	0	1	1	0.0%	1	0.0%
	0.1	0.82304	0.85627	4.0%	0.889972	8.1%
	0.2	0.65466	0.71787	9.7%	0.783263	19.6%
	0.3	0.50219	0.59037	17.6%	0.679169	35.2%
	0.4	0.37075	0.4737	27.8%	0.577714	55.8%
	0.5	0.26276	0.37025	40.9%	0.478593	82.1%
	0.6	0.17797	0.27949	57.0%	0.381243	114.2%
	0.7	0.11387	0.19945	75.2%	0.285144	150.4%
	0.8	0.06635	0.12826	93.3%	0.190183	186.6%
	0.9	0.03027	0.06273	107.2%	0.095248	214.7%
	1	0	0	0.0%	0	0.0%
t=0.2	0	1	1	0.0%	1	0.0%
	0.1	0.8726	0.88582	1.5%	0.89985	3.1%
	0.2	0.74791	0.77376	3.5%	0.79963	6.9%
	0.3	0.62834	0.66373	5.6%	0.69952	11.3%
	0.4	0.51583	0.55711	8.0%	0.59954	16.2%
	0.5	0.41157	0.45546	10.7%	0.49975	21.4%
	0.6	0.31596	0.35766	13.2%	0.40014	26.6%
	0.7	0.22857	0.26424	15.6%	0.30054	31.5%
	0.8	0.14813	0.17455	17.8%	0.20105	35.7%
	0.9	0.07274	0.08668	19.2%	0.10091	38.7%
	1	0	0	0.0%	0	0.0%
t=0.4	0	1	1	0.0%	1	0.0%
	0.1	0.8962	0.89683	0.1%	0.90018	0.4%
	0.2	0.79278	0.79574	0.4%	0.80024	0.9%
	0.3	0.69006	0.69407	0.6%	0.70033	1.5%
	0.4	0.58832	0.59431	1.0%	0.60048	2.1%
	0.5	0.48772	0.49324	1.1%	0.50071	2.7%
	0.6	0.38832	0.39463	1.6%	0.40102	3.3%
	0.7	0.29006	0.29565	1.9%	0.30127	3.9%
	0.8	0.19278	0.19704	2.2%	0.20155	4.5%
	0.9	0.0962	0.09866	2.6%	0.10118	5.2%
	1	0	0	0.0%	0	0.0%

The average of the error from GEM and FEM were calculated at each time step, and these averages were plotted against time (Figure 5.10, Plot D). The results show that the errors diminish asymptotically to zero with time because the values are tending to the steady state condition.

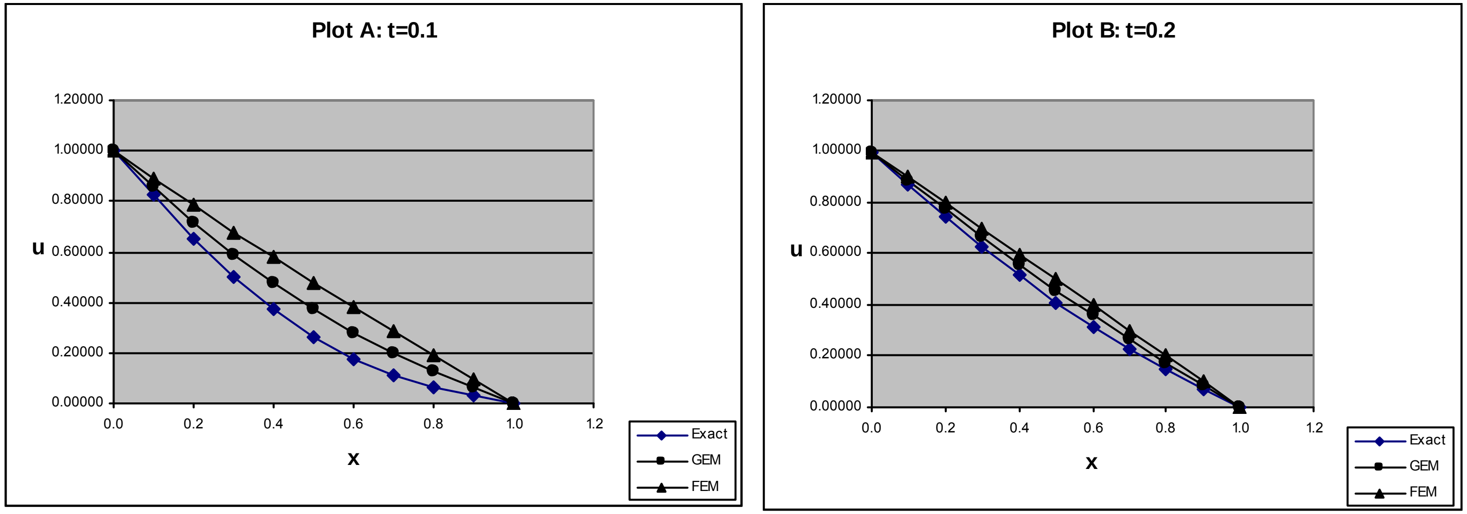


Figure 5.10 Computed potentials at selected time values: $t = 0.1$ **and** $t = 0.2$**, Dirichlet conditions.**

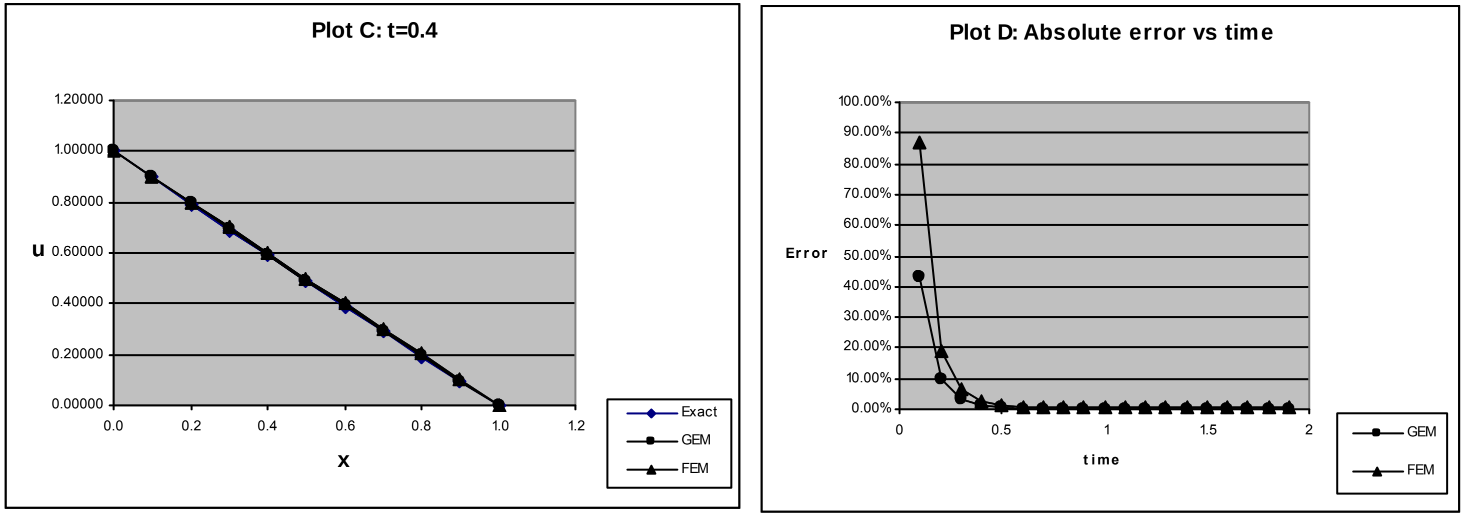


Figure 5.11 Computed potentials at selected time values $t = 0.4$ and absolute error vs. time, Dirichlet conditions.

The exact, GEM, and FEMWATER solutions for the heat flux were as listed in Table 5.12 and as shown in Figure 5.12 and Figure 5.13.

Table 5.12 Computed heat fluxes at selected time values, Dirichlet conditions.

		Exact	GEM		FEM	
Time	x	q	q	ERROR	q	ERROR
t=0.1	0	1.784286	1.4398736	19.3%	1.097148	38.5%
	0.1	1.740318	1.4061601	19.2%	1.072786	38.4%
	0.2	1.614894	1.330073	17.6%	1.04929	35.0%
	0.3	1.425954	1.2237977	14.2%	1.02292	28.3%
	0.4	1.1989	1.0952456	8.6%	0.994034	17.1%
	0.5	0.9614077	0.9679867	0.7%	0.97681	1.6%
	0.6	0.7386564	0.8494519	15.0%	0.960945	30.1%
	0.7	0.5501937	0.7490586	36.1%	0.948331	72.4%
	0.8	0.4089574	0.6741125	64.8%	0.939369	129.7%
	0.9	0.3221266	0.6274037	94.8%	0.933145	189.7%
	1	0.2928997	0.6094826	108.1%	0.928381	217.0%
t=0.2	0	1.278567	1.1343853	11.3%	0.994346	22.2%
	0.1	1.264827	1.1291886	10.7%	0.994586	21.4%
	0.2	1.224993	1.1097501	9.4%	0.994544	18.8%
	0.3	1.16307	1.0779984	7.3%	0.993583	14.6%
	0.4	1.085249	1.0376378	4.4%	0.992154	8.6%
	0.5	0.9992553	0.9948146	0.4%	0.991228	0.8%
	0.6	0.9135457	0.9509114	4.1%	0.990356	8.4%
	0.7	0.8364701	0.9123683	9.1%	0.990442	18.4%
	0.8	0.7754672	0.8827474	13.8%	0.990458	27.7%
	0.9	0.7363778	0.8617718	17.0%	0.990011	34.4%
	1	0.7229224	0.8546944	18.2%	0.986845	36.5%
t=0.4	0	1.038593	1.0140806	2.4%	0.991087	4.6%
	0.1	1.036704	1.012638	2.3%	0.991647	4.3%
	0.2	1.031222	1.0106372	2.0%	0.992012	3.8%
	0.3	1.022684	1.0055702	1.7%	0.991718	3.0%
	0.4	1.011926	1.0015112	1.0%	0.991411	2.0%
	0.5	0.9999997	0.9937334	0.6%	0.991392	0.9%
	0.6	0.988074	0.9896604	0.2%	0.991442	0.3%
	0.7	0.9773158	0.9847891	0.8%	0.992339	1.5%
	0.8	0.968778	0.9801909	1.2%	0.992903	2.5%
	0.9	0.9632965	0.9777798	1.5%	0.992822	3.1%
	1	0.9614077	0.9740249	1.3%	0.989721	2.9%

Note from Figure 5.10 that the gradient of the u plots at small time values near x=0 is much greater than at greater time values neat x=1. This manifests in the flux values being greater at x=0 than at x=1 as is evident in Figure 5.12. This variation is captured much better by GEM than FEM. The flux tends to the value of unit with time everywhere within the domain because of the constant potentials applied at the ends of the rod.

The averages of the absolute errors for the flux values show a similar trend to that shown by the absolute errors of the potentials for the same reason sited there.

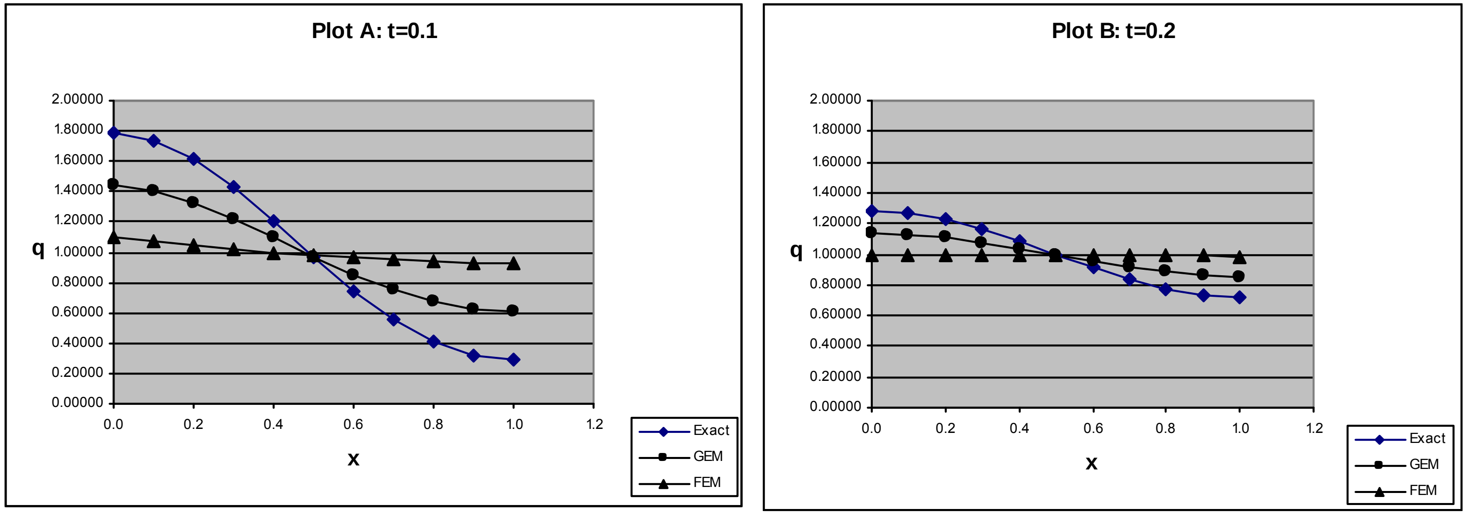


Figure 5.12 Computed heat fluxes at selected time values: $t = 0.1$ and $t = 0.2$, Dirichlet conditions.

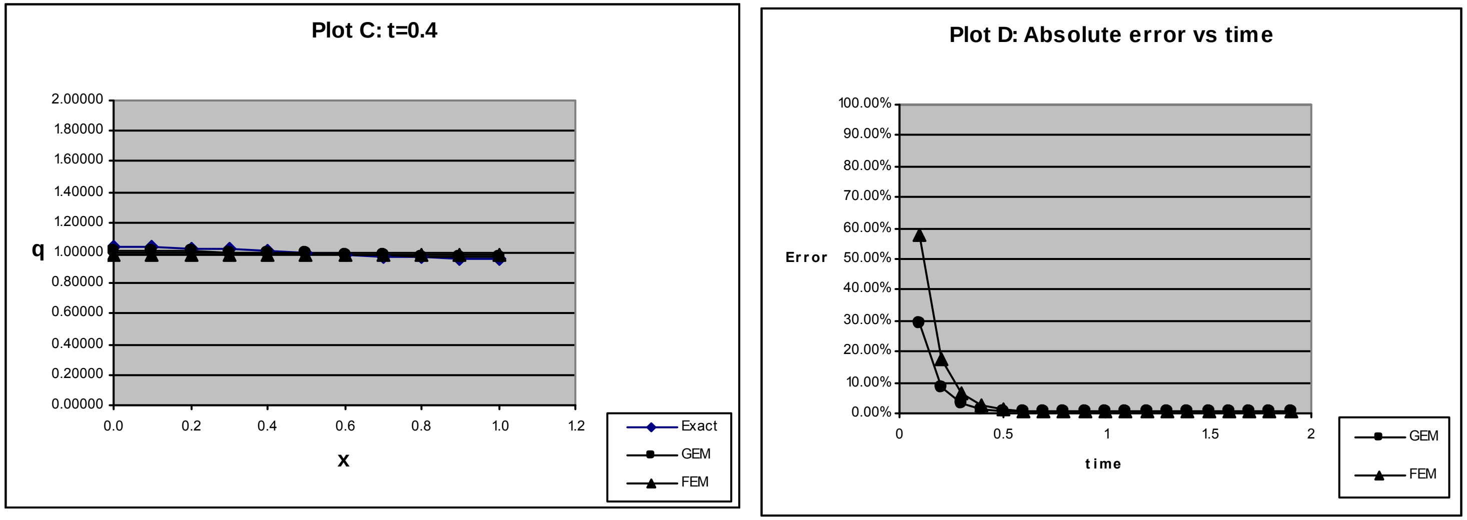


Figure 5.13 Computed heat fluxes at selected time value $t = 0.4$ and absolute error vs. time s, Dirichlet conditions.

5.2.3 Example 3 – Transient diffusion with Neumann conditions

Our third example is similar to that of the second, except that unit magnitude of flux of heat is maintained at the ends of the bar (Neumann boundary conditions). The problem is thus described by

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} \qquad \textbf{(5.10)}$$

The boundary conditions are $\left.\frac{\partial u}{\partial x}\right|_{(0,t)} = -1 \;\; and \;\; \left.\frac{\partial u}{\partial x}\right|_{(1,t)} = 1$. The initial condition is

$u(x,0) = 0$.

The exact solution is derived by method of Laplace transform, and is given by

$$u(x,t) = -x + \frac{1}{2} - \frac{4}{\pi^2} \sum_{n=1}^{\infty} \frac{1}{(2n-1)^2} \exp\left[-(2n-1)^2 \pi^2 t\right] \cos\left[(2n-1)\pi x\right] \qquad \textbf{(5.11)}$$

The same discretization for Example 2 of transient diffusion with Dirichlet conditions was used in this example (Figure 5.9). The exact, GEM, and FEMWATER solutions for the potential (heat) were as listed in Table 5.13 and as shown in Figure 5.14 and Figure 5.15. It is apparent from Figure 5.14 that when Neumann conditions are specified, the both GEM and FEM are able to calculate potentials better than when Dirichlet conditions are specified, as is evidenced by the fact that the shapes of the curves by GEM and by FEM approximate the exact solution well. As has been the case in all absolute error plots so far, the error from GEM is less than the error from FEM.

Table 5.13 Computed potentials at selected time values, Neumann conditions.

		Exact	GEM		FEM	
Time	x	u	u	ERROR	u	ERROR
t=0.1	0	0.348941	0.413126	18.4%	0.477796	36.9%
	0.1	0.256337	0.317784	24.0%	0.379409	48.0%
	0.2	0.177798	0.229515	29.1%	0.281930	58.6%
	0.3	0.111219	0.148043	33.1%	0.185021	66.4%
	0.4	0.053327	0.070500	32.2%	0.087830	64.7%
	0.5	0.000000	-0.005049	N/A	-0.010109	N/A
	0.6	-0.053327	-0.081118	52.1%	-0.108976	104.4%
	0.7	-0.111219	-0.160113	44.0%	-0.209093	88.0%
	0.8	-0.177798	-0.244153	37.3%	-0.310545	74.7%
	0.9	-0.256337	-0.334851	30.6%	-0.413614	61.4%
	1	-0.348941	-0.432728	24.0%	-0.518159	48.5%
t=0.2	0	0.443701	0.466880	5.2%	0.491764	10.8%
	0.1	0.346457	0.369298	6.6%	0.392478	13.3%
	0.2	0.254454	0.273602	7.5%	0.292760	15.1%
	0.3	0.166909	0.179784	7.7%	0.192768	15.5%
	0.4	0.082603	0.087122	5.5%	0.091819	11.2%
	0.5	0.000000	-0.005192	N/A	-0.010388	N/A
	0.6	-0.082603	-0.098020	18.7%	-0.113651	37.6%
	0.7	-0.166909	-0.192108	15.1%	-0.217766	30.5%
	0.8	-0.254454	-0.288496	13.4%	-0.322680	26.8%
	0.9	-0.346457	-0.386616	11.6%	-0.428052	23.6%
	1	-0.443701	-0.488584	10.1%	-0.533684	20.3%
t=0.4	0	0.492180	0.492080	0.0%	0.492717	0.1%
	0.1	0.392562	0.392381	0.0%	0.393391	0.2%
	0.2	0.293673	0.293320	0.1%	0.293536	0.0%
	0.3	0.195403	0.194060	0.7%	0.193346	1.1%
	0.4	0.097583	0.094860	2.8%	0.092166	5.6%
	0.5	0.000000	-0.005148	N/A	-0.010317	N/A
	0.6	-0.097583	-0.105717	8.3%	-0.113872	16.7%
	0.7	-0.195403	-0.206842	5.9%	-0.218296	11.7%
	0.8	-0.293673	-0.308367	5.0%	-0.323469	10.1%
	0.9	-0.392562	-0.410667	4.6%	-0.429007	9.3%
	1	-0.492180	-0.512607	4.2%	-0.534654	8.6%

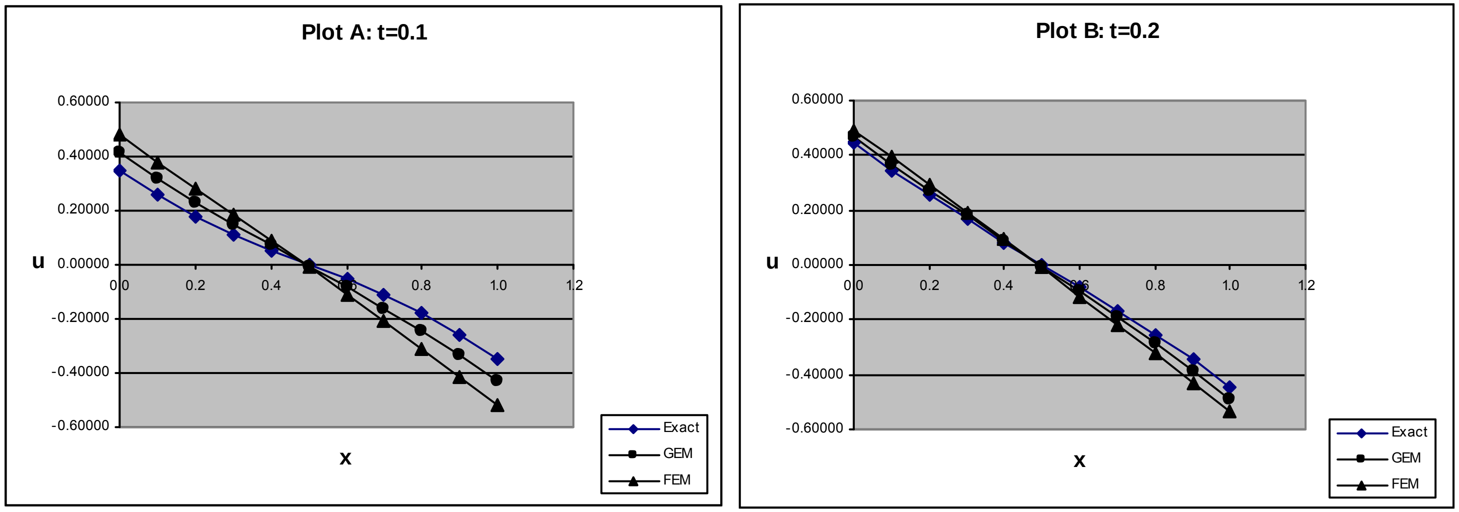


Figure 5.14 Computed potentials at selected time values: $t = 0.1$ and $t = 0.2$, Neumann condition

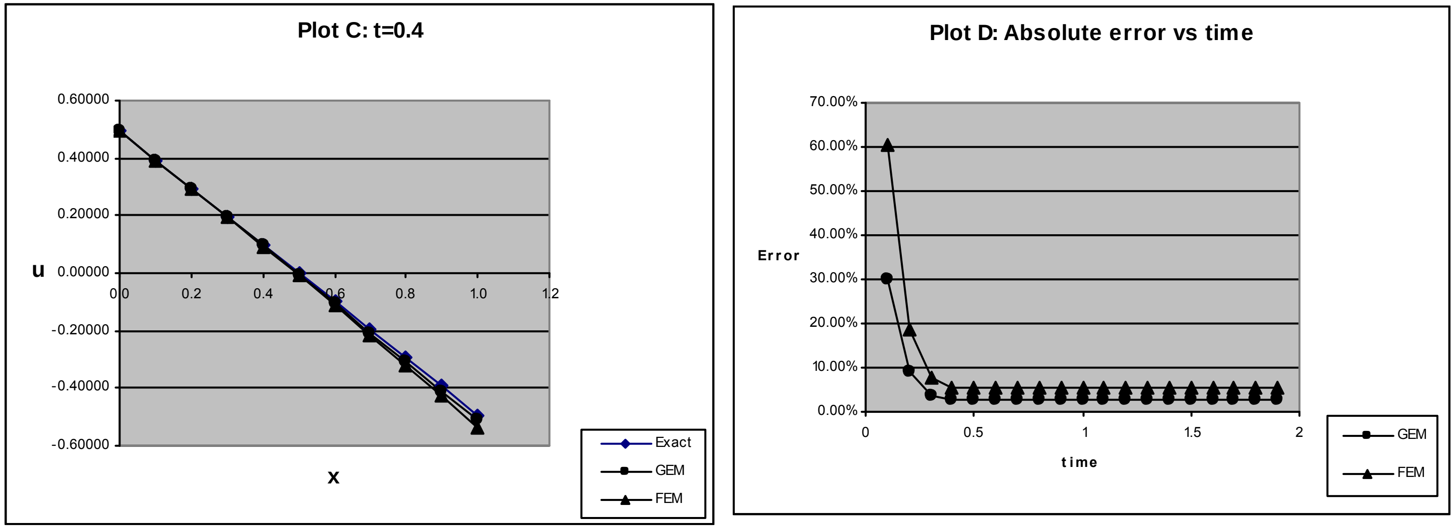


Figure 5.15 Computed potentials at selected time values $t = 0.4$ and absolute error vs. time, Neumann condition.

The exact, GEM, and FEMWATER solutions for the heat flux for this second example were as listed in Table 5.14 and as shown in Figure 5.16 and Figure 5.17.

Table 5.14 Computed heat fluxes at selected time values, Neumann conditions.

		Exact	GEM		FEM	
Time	x	q	q	ERROR	q	ERROR
t=0.1	0	1.000000	1.000000	0.0%	1.000000	0.0%
	0.1	0.853310	0.917726	7.5%	0.984659	15.4%
	0.2	0.721013	0.847391	17.5%	0.976776	35.5%
	0.3	0.616066	0.791330	28.4%	0.969494	57.4%
	0.4	0.548714	0.755657	37.7%	0.962982	75.5%
	0.5	0.525513	0.743157	41.4%	0.961595	83.0%
	0.6	0.548714	0.754671	37.5%	0.962636	75.4%
	0.7	0.616066	0.791211	28.4%	0.969162	57.3%
	0.8	0.721013	0.847946	17.6%	0.975097	35.2%
	0.9	0.853310	0.921111	7.9%	0.989371	15.9%
	1	1.000000	1.000000	0.0%	1.000000	0.0%
t=0.2	0	1.000000	1.000000	0.0%	1.000000	0.0%
	0.1	0.945345	0.972472	2.9%	1.001771	6.0%
	0.2	0.896040	0.947738	5.8%	1.002976	11.9%
	0.3	0.856912	0.929529	8.5%	1.004637	17.2%
	0.4	0.831789	0.917837	10.3%	1.004617	20.8%
	0.5	0.823133	0.913121	10.9%	1.004611	22.0%
	0.6	0.831789	0.917878	10.3%	1.004470	20.8%
	0.7	0.856912	0.929592	8.5%	1.005165	17.3%
	0.8	0.896040	0.949799	6.0%	1.004356	12.1%
	0.9	0.945345	0.973940	3.0%	1.004391	6.2%
	1	1.000000	1.000000	0.0%	1.000000	0.0%
t=0.4	0	1.000000	1.000000	0.0%	1.000000	0.0%
	0.1	0.992408	0.996765	0.4%	1.002775	1.0%
	0.2	0.985559	0.993351	0.8%	1.004500	1.9%
	0.3	0.980123	0.993069	1.3%	1.006746	2.7%
	0.4	0.976634	0.991287	1.5%	1.007282	3.1%
	0.5	0.975431	0.990303	1.5%	1.007470	3.3%
	0.6	0.976634	0.991342	1.5%	1.007411	3.2%
	0.7	0.980123	0.992524	1.3%	1.007880	2.8%
	0.8	0.985559	0.995118	1.0%	1.006266	2.1%
	0.9	0.992408	0.997914	0.6%	1.005329	1.3%
	1	1.000000	1.000000	0.0%	1.000000	0.0%

In Figure 5.14 the V shapes of the flux plots are explained by the fact that the values at the boundaries have been specified. The exact solution shows that GEM is able to handle transients better than FEM. This is evident because the flux plot for GEM is closer to the exact solution that that of FEM. The average absolute errors for the fluxes at each time step is less than that in all the previous cases. This is because of the specification of the Neumann boundary conditions.

That the average errors tends to zero with time is because this problem is essentially in 1D and the errors in the computations do not accumulate to significant values as has been evident with other 3D simulations in this work.

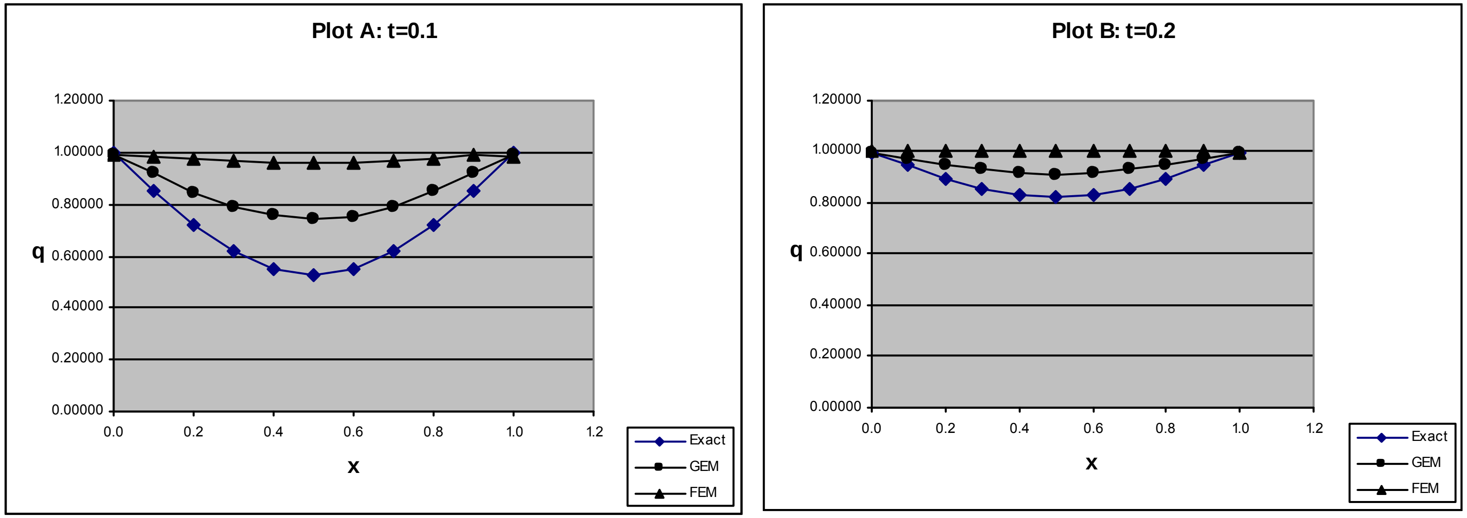


Figure 5.16 Computed heat fluxes at selected time values: $t = 0.1$ and $t = 0.2$, Neumann conditions.

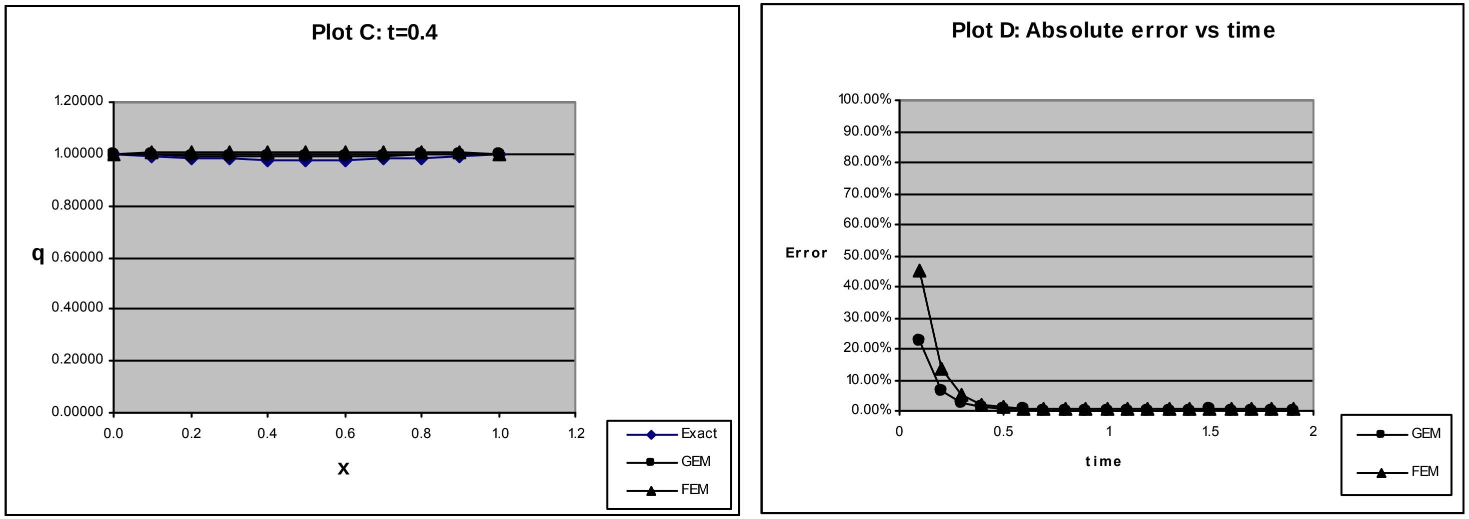


Figure 5.17 Computed heat fluxes at selected time values $t = 0.4$ and absolute error vs. time, Neumann conditions.

6. CONCLUSION

Although the study of partial differential equations began in the 18th Century, the development of methods for their solution in practically based problems continues to be a subject of major interest in the research today. Two such partial differential equations are the Laplace equation and its variant the Poisson equation, and the time-dependent diffusion equation. The fact that these equations continue to arouse great interest today is understandable, considering the fact that the equations describe phenomena in various branches of the physical science and engineering.

A number of techniques and methods, as reviewed in this project, are under development for the solution of these differential equations, and numerical techniques have proven to be particularly helpful in producing solutions to practically based potential flows governed by such equations One such technique is the Green element method, which is based on the theory of the boundary element method but is implemented in an element by element method in the fashion of the Finite element method.

The objective of the research project was to develop a numerical formulation based on the Green element method for time-dependent potential flow problems that extends the work done to date for the one dimensional and two dimensional cases to three dimensions. The hypothesis was raised that it is possible to develop a three-dimensional formulation for the Green element method and in the event that the three dimensional formulation is developed, it was also the objective of the project to test if the computational gains that have been reported for the one dimensional and two-dimensional cases do extend to three dimensions.

In this project, extension of the Green element method to three-dimensional domains has been accomplished. The performances of GEM and FEMWATER were compared on a number of simple potential flow problems for which analytical solutions are available. In all cases GEM performed better than FEMWATER. It was observed that with similar discretization, the Green element method generally produces more accurate solutions than the Finite element method. Therefore, the computational gain in terms of accuracy that has been reported for the one dimensional and the two-dimensional cases do indeed extend to three dimensions. In real world potential flow problems where the solution is often not known, the ability to generate an accurate solution becomes particularly important. Given the second order accuracy of boundary element theory that is utilised by GEM, and the flexibility offered by the element by element implementation of the theory in the manner of FEM, GEM is able to handle heterogeneities arising in real world groundwater aquifers.

It is also worth noting that the computational gain observed comes at the expense of a large flux coefficient matrix, which results due to the utilisation of all the fluxes normal to the computation planes of the elements for all planes that meet at an internal node. This leads to the closure problem whereby more unknowns are generated than the available number of equations as elaborated in this report. For 1D and 2D GEM analysis, this closure problem has been effectively resolved by previous researchers (Taigbenu, 2008), but this is not the case with 3D analysis. The closure problem has been circumvented in this project by adopting cuboidal hexahedra elements and approximating the fluxes by their potential difference expressions. Another advantage of adopting cuboidal hexahedral elements is that it becomes possible to avoid the integrals which generate results with imaginary components simply by taking advantage of the symmetry of the elements. This allows the replacement of those integrals which have complex (imaginary) results with the simpler versions. There is thus need to resolve the closure problem either along the lines of what Taigbenu accomplished for 2D cases, or along the lines of expressing the fluxes at each node only in terms of three flux variables namely q_x, q_y, and q_z (Pecher et al., 2001), and to implement better integration methods which can handle the imaginary components. Thus, we note that although the Green element method is a very promising new method for the solution of potential flow problems governed by the Laplace equation and its variants, two major challenges still need to be resolved, namely the need to resolve the closure problem and the domains integration which result in imaginary components. The resolution of these challenges will make it possible to utilise the more general tetrahedra and hexahedra elements without the need to approximate the fluxes by their potential difference expressions and without the need to rely on the symmetry of the elements. These will be subject of future research to achieve a more robust 3D GEM.

A novel system to implement surface compatibility issues has been developed and implemented. This involves associating each surface with a unique surface identity code, thereby precluding duplication of surfaces. This means that where two elements interact, they do so across one surface shared between the two elements. With this system, it becomes possible to identify the fluxes using the surface to which they are perpendicular, and the node at which they are taken. Ramsak and Skerget (2009) have observed that the procedures for resolving these compatibility issues have not hitherto been developed, and that this has been one of the issues that has led to the non development of boundary element techniques for multi-domain systems in 3D problems. This report represents the resolution of this problem that has slowed down the development of the Green element method for 3D domains.

REFERENCES

Abashar, M. E. E. (2004). "Application of the Green Element Method to Chemical Engineering Problems." *Journal of King Saud University.*, Vol. 17 (No. 1), pp 47-59.

Abdel-Fattah, A., Langford, R., Schulze-Makuch, D. (2007). "Applications of Particle-Tracking Techniques to Bank Infiltration: A Case Study from El Paso, Texas, USA." *Environmental Geology,* Vol. 55 (No. 3), pp 505-515.

Archer, R. (2005). "C1 Continuous Solutions from the Green Element Method using Overhauser Elements." *Applied Numerical Mathematics,* Vol. 56, pp 222-229.

Archer, R., and Horne, R. N. (2002). "Green Element Method and Singularity Programming for Numerical Well Test Analysis." *Engineering Analysis with Boundary Elements,* Vol. 26 (No. 6), pp 537-546.

Archer, R., and Horne, R. N. (2000). "The Green Element Method for Numerical Well Test Analysis." *Proceedings - SPE Annual Technical Conference and Exhibition,* Vol. OMEGA, pp 137-145.

Archer, R., Horne, R. N., Onyejekwe, O. O. (1999). "Petroleum Reservoir Engineering Applications of the Dual Reciprocity Boundary Element Method and the Green Element Method." *Proc., World Conference on the*

Boundary Element Method (21st Conference), Wessex Institute, Oxford University, London, pp 525-536.

Arnold, D. N. (1990). "Mixed Finite Element Methods for Elliptic Problems." *Proc., Proceedings of the Workshop on Reliability in Computational Mechanics,* Elsevier, Austin, TX, USA, pp 281-300.

Azoury, P. H. (1992). *Engineering Applications of Unsteady Fluid Flow,* 1st Ed., John Wiley & Sons, New York.

Bear, J., and Verrijt, A. (1988). *Modeling Groundwater Flow and Pollution: With Computer Programs for Sample Cases.* 2nd Ed., D. Reidel Publishing Company, Boston.

Bear, J. (1972). *Dynamics of Fluids in Porous Media,* American Elsevier, New York.

Bergamaschi, L., Mantica, S., Saleri, F., Reservoir, R., Agip, S. P. A., Via, E., Milanese, S. D. (1994). *Mixed Finite Element Approximation of Darcy's Law in Porous Media,* Technical Report CRS4, Cagliari.

Bozkaya, N., and Tezer-Sezgin, M. (2008). "Time-Domain BEM Solution of Convection-Diffusion-Type MHD Equations." *Int. J. Numer. Methods Fluids,* Vol. 56 (No. 11), pp 1969-1991.

Brebbia, C. A., and Dominguez, J. (1992). *Boundary Elements: An Introductory Course,* McGraw Hill, New York.

Brebbia, C. A., and Nardini, D. (1983). "Dynamic Analysis in Solid Mechanics by an Alternative Boundary Element Procedure." *International Journal of Soil Dynamics and Earthquake Engineering,* Vol. 2 (No. 4), pp 228-233.

Brebbia, C. A., and Walker, S. (1980). *Boundary Element Techniques in Engineering,* Butterworth, London.

Brebbia, C. A. (1978). *The Boundary Element Method for Engineers,* Pentech Press, London.

Brezis, H., and Browder, F. (1998). "Partial Differential Equations in the 20th Century." *Advances in Mathematics,* Vol. 135, pp 76-144.

Burden, R. L., and Faires, J. D. (2001). *Numerical Analysis,* Seventh Ed., Brooks/Cole, Calofornia.

Calderon, A. P., and Zygmund, A. (1952). "On the Existence of Certain Singular Integrals." *Acta Mathematica,* Vol. 88 (No. 1), pp 85-139.

Carrer, J. A. M., and Mansur, W. J. (2002). "Time-Dependent Fundamental Solution Generated by a Not Impulsive Source in the Boundary Element Method Analysis of the 2D Scalar Wave Equation." *Communications in Numerical Methods in Engineering,* Vol. 18 (No. 4), pp 277-285.

Carslaw, H. S., and Jaeger, J. C. (1959). *Conduction of Heat in Solids,* 2nd Ed., Oxford University Press, London.

Chen, C. S., Golberg, M. A., Schaback, R. A. (2003). "Recent Developments in the Dual Reciprocity Method using Compactly Supported Radial Basis Functions." *Transformation of Domain Effects to the Boundary,* 1 st Ed., Wit Press, London, pp 1-41.

Csoma, R. (2005). "Modelling the Local Variation of Aquifer Parameters with the Help of the Analytic Element Method." *Periodica Polytechnica: Civil Engineering,* Vol. 49 (No. 2), pp 137-156.

Dai, K. Y., Liu, G. R., Lim, K. M., Han, X., Du, S. Y. (2004). "A Meshfree Radial Point Interpolation Method for Analysis of Functionally Graded Material (FGM) Plates." *Comput. Mech.,* Vol. 34 (No. 3), pp 213-223.

Dake, L. P. (1978). *Fundamentals of Reservoir Engineering,* 1st Ed., Elsevier Science, Amsterdam, The Netherlands.

Davies, A. J., and Crann, D. (2004). "The Laplace Transform Boundary Element Methods for Diffusion Problems with Periodic Boundary Conditions." *Proc., 26th World Conference on Boundary Elements and Other Mesh Reduction Methods, BEM XXVI,* WIT Press, Bologna - Italy, pp 393-402.

Davies, A. J., and Honnor, M. E. (2002). "Time-Domain Laplace Transform Boundary Element Methods for Diffusion Problems." *Proc., Seventh International Conference on Application of High-Performance Computing in Engineering, HPC VII,* WITPress, Bologna - Italy, pp 65-74.

Dolbow, J., and Belytschko, T. (1999). "Numerical Integration of the Galerkin Weak Form in Meshfree Methods." *Comput. Mech.,* Vol. 23 (No. 3), pp 219-230.

Drebushchak, V. A. (2009). "Heat Capacity Increases with Pressure." *Journal of Thermal Analysis and Calorimetry,* Vol. 95 (No. 1), pp 313-317.

Ewing, D. J. F., Fawkes, A. J., Griffiths, J. R. (1970). "Rules Governing the Numbers of Nodes and Elements in a Finite Element Mesh." *International Journal for Numerical Methods in Engineering,* Vol. 2 (No. 4), pp 597-600.

Fredrick, K. C., Becker, M. W., Flewelling, D. M., Silavisesrith, W., Hart, E. R. (2004). "Enhancement of Aquifer Vulnerability Indexing using the Analytic-Element Method." *Environ. Geol.,* Vol. 45 (No. 8), pp 1054-1061.

Ganis, B., Klie, H., Wheeler, M. F., Wildey, T., Yotov, I., Zhang, D. (2008). "Stochastic Collocation and Mixed Finite Elements for Flow in Porous Media." *Computer Methods in Applied Mechanics and Engineering,* Vol. 197 (No. 43-44), pp 3547-3559.

Gao, X. W., and Davies, T. G. (2000). "3D Multi-Region BEM with Corners and Edges." *International Journal of Solids and Structures*, Vol. 37 (No. 11), pp 1549-1560.

Gao, X. (2004). "A Promising Boundary Element Formulation for Three-Dimensional Viscous Flow." *International Journal for Numerical Methods in Fluids*, Vol. 47 (No. 1), pp 19-43.

Gatmiri, B., and Van Nguyen, K. (2005). "Time 2D Fundamental Solution for Saturated Porous Media with Incompressible Fluid." *Communications in Numerical Methods in Engineering*, Vol. 21 (No. 3), pp 119-132.

Gil, S., Saleba, M. E., Tobia, D. (2002). "Experimental Study of the Neumann and Dirichlet Boundary Conditions in Two-Dimensional Electrostatic Problems." *Am. J. Phy.*, Vol. 70 (No. 12), pp 1208-1213.

Golub, G. H., and Van Leon, C. F. (1989). *Matrix Computations,* 2nd Ed., John Hopkins University Press, Baltimore.

Gray, L. J., and Kaplan, T. (2001). "3D Galerkin Integration without Stokes' Theorem." *Engineering Analysis with Boundary Elements*, Vol. 25 (No. 4-5), pp 289-295.

Hemker, P. W. (2004). "Discretisation of PDEs, finite element method." <http://homepages.cwi.nl/~pieth/LECTURES/lectures04.pdf> (08 June 2009, 2009).

Hsieh, C. K., Choi, C. -., Kassab, A. J. (1992). "Solution of Stefan Problems by a Boundary Element Method." *Proc., Seventh International Conference on Boundary Element Technology - BETECH 92, June 1, 1992 - June 1,* Publ by Computational Mechanics Publ, Albuquerque (NM, USA), pp 473-490.

Hu, W., Guang Yao, L., Zhi Hua, Z. (2007). "Parallel Point Interpolation Method for Three-Dimensional Metal Forming Simulations." *Eng. Anal. Boundary Elements*, Vol. 31 (No. 4), pp 326-342.

Huebner, K. H., Dewhirst, D. L., Smith, D. E., Byrom, T. G. (2001). "Meet the Finite Element Method." *The Finite Element Method for Engineers,* 4th Ed., John Wiley & Sons, New York, pp 3-16.

Hunt, R. J. (2006). "Ground Water Modeling Applications using the Analytic Element Method." *Ground Water,* Vol. 44 (No. 1), pp 5-15.

Incropera, F. P., DeWitt, D. P., Bergman, T. L., Lavine, A. S. (2006). *Introduction to Heat Transfer,* 5th Ed., John Wiley & Sons, New York.

Jankovic, I., Fiori, A., Dagan, G. (2003). "Flow and Transport in Highly Heterogeneous Formations: 3. Numerical Simulations and Comparison with Theoretical Results." *Water Resour. Res.,* Vol. 39 (No. 9), pp SBH161-SBH1613.

Jones, N. L., Lemon, A. M., Kennard, M. J. (2005). "Efficient Data Management Strategies for Large MODFLOW Models." *Proc., 2005 World Water and Environmental Resources Congress, may 15, 2005 - may 19,* American Society of Civil Engineers, Anchorage (AK, United states), pp 365-371.

Ketkar, S. P. (1993). "Iterative and Direct Solution Methods in Thermal Network Solvers." *Int. Commun. Heat Mass Transfer,* Vol. 20 (No. 4), pp 527-534.

Khosravifard, A., and Hematiyan, M. R. (2009). "A New Method for Meshless Integration in 2D and 3D Galerkin Meshfree Methods." *Engineering Analysis with Boundary Elements,* Vol. 34 (No. 1), pp 30-40.

Knochenmus, L. A., and Robinson, J. L. (1996). *Descriptions of Anisotropy and Heterogeneity and their Effect on Ground-Water Flow and Areas of Contribution to Public Supply Wells in a Karst Carbonate Aquifer System.* Unites States Government Printing Office, Washington.

Kraemer, S. R. (2003). "The Maturity of Analytic Element Ground-Water Modeling as a Research Program (1980-2003)." *Proc., 4Th International Conference on the Analytic Element Method,* American Society of Civil Engineers, Saint-Etienne-France, pp 1-29.

Kreysig, E. (1988). *Advanced Engineering Mathematics,* John Wiley and Sons, Oxford, England.

Liao, S. (1998). "On the General Boundary Element Method." *Eng. Anal. Boundary Elements,* Vol. 21 (No. 1), pp 39-51.

Liggett, J. A., and Liu, L. F. (1983). *The Boundary Integral Equation Method for Porous Media Flow,* 1st Ed., George Allen & Unwin Ltd, London.

Liu, H., and Zhu, S. (2002). "The Dual Reciprocity Boundary Element Method for Magnetohydrodynamic Channel Flows." *Anziam,* Vol. 44, pp 305.

Livesley, R. K. (1983). *Finite Elements: An Introduction for Engineers,* Cambridge University Press, Cambridge United Kingdom.

Lorinczi, P., Harris, S. D., Elliot, L. (2009). "Modified Flux-Vector-Based Green Element Method for Problems in Steady-State Anisotrpic Media." *Engineering Analysis with Boundary Elements,* Vol. 33 (No. 3), pp 368-387.

Luther, K. H., and Haitjema, H. M. (1998). "Numerical Experiments on the Residence Time Distributions of Heterogeneous Groundwatersheds." *Journal of Hydrology,* Vol. 207 (No. 1-2), pp 1-17.

Maduramuthu, P., and Fenner, R. T. (2004). "Three-Dimensional Shape Design Optimization of Holes and Cavities using the Boundary Element Method." *J. Strain Anal. Eng. Des.,* Vol. 39 (No. 1), pp 87-98.

Maryska, J., Rozloznik, M., Tuma, M. (1995). "Mixed-Hybrid Finite Element Approximation of the Potential Fluid Flow Problem." *Journal of Computational and Applied Mathematics,* Vol. 63 (No. 1-3), pp 383-392.

Matott, L. S., Rabideau, A. J., Craig, J. R. (2006). "Pump-and-Treat Optimization using Analytic Element Method Flow Models." *Adv. Water Resour.,* Vol. 29 (No. 5), pp 760-775.

McCarthy, J. F. (1994). "Analytical Solutions of 2D Cresting Models using the Hodograph Method." *Transport in Porous Media,* Vol. 15 (No. 3), pp 251-269.

Moridis, G. J. (1992). "Alternative Formulations of the Laplace Transform Boundary Element (LTBE) Numerical Method for the Solution of Diffusion-Type Equations." *Proc., Seventh International Conference on Boundary Element Technology - BETECH 92,* Publ by Computational Mechanics Publ, Albuquerque (NM, USA), pp 815-833.

Narayanan, R., Indrasenan, N., Elango, K. (1992). "Extended Boundary Element Method for Inhomogeneous and Unconfined Aquifers." *Engineering Analysis with Boundary Elements,* Vol. 10 (No. 3), pp 241-247.

Natalini, B., and Popov, V. (2007). "Tests of Performance of the Boundary Element Dual Reciprocity Method: Multi-Domain Approach for 3D Problems." *Mecanica Comptacional,* Vol. XXVI, pp 1174.

Niku, S. M., and Adey, R. A. (1996). "Computational Aspect of the Dual Reciprocity Method for Dynamics." *Engineering Analysis with Boundary Elements,* Vol. 18 (No. 1), pp 43-61.

Nowak, A. J., and Brebbia, C. A. (1992). "Numerical Verification of the Multiple Reciprocity Method for Linear Potential Problems with Body Forces." *Eng. Anal. Boundary Elements,* Vol. 10 (No. 3), pp 259-266.

Onate, E., Cervera, M., Zienkiewicz, O. C. (1994). "Finite Volume Format for Structural Mechanics." *International Journal for Numerical Methods in Engineering,* Vol. 37 (No. 2), pp 181-201.

Onyejekwe, O. O. (2006). "A Note on Green Element Method Discretization for Poisson's Equation in Polar Coordinates." *Applied Mathematics Letters,* Vol. 19 (No. 8), pp 785-788.

Park, E. (2005). "Mixed Finite Element Methods for Generalized Forchheimer Flow in Porous Media." *Numerical Methods for Partial Differential Equations,* Vol. 21 (No. 2), pp 213-228.

Pecher, R., Harris, S. D., Knipe, R. J., Elliot, L., Ingham, D. B. (2001). "New Formulation of the Green Element Method to Maintain its Second-Order Accuracy in 2D/3D." *Engineering Analysis with Boundary Elements,* Vol. 25 (No. 3), pp 211-219.

Peratta, A., and Popov, V. (2006). "A New Scheme for Numerical Modelling of Flow and Transport Processes in 3D Fractured Porous Media." *Adv. Water Resour.,* Vol. 29 (No. 1), pp 42-61.

Poincaré, H. (1890). "Sur Les Equations Aux Dérivées Partielles De La Physique Mathématique." *American Journal of Mathematics,* Vol. 12 (No. 3), pp 211-294.

Popov, V., and Power, H. (1999). "DRM-MD Approach for the Numerical Solution of Gas Flow in Porous Media, with Application to Landfill." *Eng. Anal. Boundary Elements,* Vol. 23 (No. 2), pp 175-188.

Popov, V., and Power, H. (1998). "DRM-MD Integral Equation Method for the Numerical Solution of Convection-Diffusion Equation." *Proc., Proceedings of the 1998 2nd European Boundary Element Method Symposium,* Computational Mechanics Publ, Southampton, UK, pp 67-80.

Popov, V., and Power, H. (1996). "Domain Decomposition on the Dual Reciprocity Approach." *Boundary Elements Communications,* Vol. 7 (No. 1), pp 1-5.

Portapila, M. I., and Power, H. (2007). "Iterative Schemes for the Solution of Systems of Equations Arising from the DRM in Multidomains." *Domain Decomposition Techniques for Boundary Elements (Application to Fluid Flow),* First Ed., WIT Press, London, pp 236-298.

Pozrikidis, C. (2006). "A Note on the Relation between the Boundary- and Finite-Element Method with Application to Laplace's Equation in Two Dimensions." *Eng. Anal. Boundary Elements,* Vol. 30 (No. 2), pp 143-147.

Ramšak, M., and Škerget, L. (2009). "3D Multidomain BEM for a Poisson Equation." *Engineering Analysis with Boundary Elements,* Vol. 33 (No. 5), pp 689-694.

Ramšak, M., and Škerget, L. (2007). "3D Multidomain BEM for Solving the Laplace Equation." *Engineering Analysis with Boundary Elements,* Vol. 31 (No. 6), pp 528-538.

Rosca, V. E., and Leitao, V. M. A. (2008). "Quasi-Monte Carlo Mesh-Free Integration for Meshless Weak Formulations." *Eng. Anal. Boundary Elements,* Vol. 32 (No. 6), pp 471-479.

Rühaak, W., Rath, V., Wolf, A., Clauser, C. (2008). "3D Finite Volume Groundwater and Heat Transport Modeling with Non-Orthogonal Grids, using a Coordinate Transformation Method." *Advances in Water Resources,* Vol. 31 (No. 3), pp 513-524.

Schclar, N. A., and Partridge, P. W. (1993). "3D Anisotropic Elasticity with BEM using the Isotropic Fundamental Solution." *Engineering Analysis with Boundary Elements,* Vol. 11 (No. 2), pp 137-144.

Seibert, J., Bishop, K., Rodhe, A., Mcdonnell, J. J. (2003). "Groundwater Dynamics Along a Hillslope: A Test of the Steady State Hypothesis." *Water Resources Research,* Vol. 39 (No. 1), pp SWC21-SWC29.

Shiah, Y. C., Tan, C. L., Lee, V. G. (2008). "Evaluation of Explicit-Form Fundamental Solutions for Displacements and Stresses in 3D Anisotropic Elastic Solids." *CMES - Computer Modeling in Engineering and Sciences,* Vol. 34 (No. 3), pp 205-226.

Strack, O. D. L. (1989). *Groundwater Mechanics,* 1st Ed., Prentice Hall, New Jersey, USA.

Sun, M., and Zheng, C. (1999). "Long-Term Groundwater Management by a Modflow Based Dynamic Optimization Tool." *Journal of the American Water Resources Association,* Vol. 35 (No. 1), pp 99-111.

Taigbenu, A. E. (2008). "The Flux-Correct Green Element Formulation for Linear, Nonlinear Heat Transport in Heterogeneous Media." *Engineering Analysis with Boundary Elements,* Vol. 32 (No. 1), pp 52-63.

Taigbenu, A. E. (2006). "Simulation of Coupled Nonlinear Electromagnetic Heating with the Green Element Method." *Proc., 9th International Conference on Advanced Computational Methods and Experimental*

Measurements in Heat and Mass Transfer 2006, HT06, WITPress, The New Forest (United Kingdom), pp 77-86.

Taigbenu, A. E. (2004). "Green Element Calculations of Nonlinear Heat Conduction with a Time-Dependent Fundamental Solution." *Eng. Anal. Boundary Elements,* Vol. 28 (No. 1), pp 53-60.

Taigbenu, A. E. (2003). "A Time-Dependent Green's Function-Based Model for Stream-Unconfined Aquifer Flows." *Water SA,* Vol. 29 (No. 3), pp 257-265.

Taigbenu, A. E. (2001). "Time-Dependent Fundamental Solution in Green Element Calculations of Nonlinear Unconfined Flow." *Proc., Twenty-Third International Conference on the Boundary Element Method, BEM XXIII,* WITPress, Lemnos, Greece, pp 223-232.

Taigbenu, A. E. (1999). *The Green Element Method,* Kluwer Academic Publishers, Boston.

Taigbenu, A. E., and Onyejekwe, O. O. (1999). "Green's Function-Based Integral Approaches to Nonlinear Transient Boundary-Value Problems (II) " *Applied Mathematical Modelling,* Vol. 23, pp 241.

Taigbenu, A. E. (1995). "The Green Element Method." *International Journal for Numerical Methods in Engineering,* Vol. 38 (No. 13), pp 2241-2263.

Taigbenu, A. E., and Onyejekwe, O. O. (1995). "Green Element Simulations of the Transient Nonlinear Unsaturated Flow Equation." *Appl. Math. Model.,* Vol. 19 (No. 11), pp 675-684.

Taigbenu, A. E. (1990). "A More Efficient Implementation of the Boundary Element Theory." *Proc., 5th International Conference on Boundary Element Technology,* BETECH 90, Newark, Delaware, pp 355-366.

Tanaka, M., Nakamura, M., Aoki, K. (1991). "Study on Computational Accuracy of the Laplace Transform Boundary Element Method for Unsteady Elastodynamic Problems. (Two-Dimensional Cases)." *Nippon Kikai Gakkai Ronbunshu, C Hen/Transactions of the Japan Society of Mechanical Engineers, Part C,* Vol. 57 (No. 542), pp 3158-3163.

Tsilingiris, P. T. (2006). "Parametric Space Distribution Effects of Wall Heat Capacity and Thermal Resistance on the Dynamic Thermal Behavior of Walls and Structures." *Energy Build.,* Vol. 38 (No. 10), pp 1200-1211.

Van Criekingen, S., and Beauwens, R. (2007). "Mixed-Hybrid Discretization Methods for the P1 Equations." *Applied Numerical Mathematics,* Vol. 57 (No. 2), pp 117-130.

Walsum, P. E. V., and Groenendijk, P. (2008). "Quasi Steady-State Simulation of the Unsaturated Zone in Groundwater Modeling of Lowland Regions." *Vadose Zone Journal*, Vol. 7 (No. 1), pp 769-781.

Wang, J., and Schweizerhof, K. (1995). "Computation of Fundamental Solutions for Laminated Anisotropic Shallow Shells." *Mech. Res. Commun.,* Vol. 22 (No. 4), pp 393-400.

Wang, S., Shao, J., Song, X., Zhang, Y., Huo, Z., Zhou, X. (2008). "Application of MODFLOW and Geographic Information System to Groundwater Flow Simulation in North China Plain, China." *Environ. Geol.,* Vol. 55 (No. 7), pp 1449-1462.

Watt, S. M. (2006). "Making Computer Algebra More Symbolic." *Proc., Transgressive Computing 2006: A Conference in Honor of Jean Della Tora,* Transgressive Computing, Granada Spain, pp 43-49.

Wolfram Research. (2010). "Symbolic computation." <http://reference.wolfram.com/mathematica/tutorial/SymbolicComputation.html> (03/03, 2010).

Wuolo, R. W., Dahlstrom, D. J., Fairbrother, M. D. (1995). "Wellhead Protection Area Delineation using the Analytic Element Method of Ground-Water Modeling." *Ground Water,* Vol. 33 (No. 1), pp 71-83.

Xia, G. H., Zhao, Y., Yeo, J. H., Lv, X. (2007). "A 3D Implicit Unstructured-Grid Finite Volume Method for Structural Dynamics." *Computational Mechanics,* Vol. 40 (No. 2), pp 299-312.

Young, D. L., Tsai, C. C., Murugesan, K., Fan, C. M., Chen, C. W. (2004). "Time-Dependent Fundamental Solutions for Homogeneous Diffusion Problems." *Eng. Anal. Boundary Elements,* Vol. 28 (No. 12), pp 1463-1473.

Zhao, X. Y., and Liao, S. J. (2003). "A Short Note on the General Boundary Element Method for Viscous Flows with High Reynolds Number." *Int. J. Numer. Methods Fluids*, Vol. 42 (No. 4), pp 349-359.

APPENDICES

Appendix A. Information on some grid generation software available on the market.

Software:	***Description:***	***Author :***	***Availability:***	***Platforms:***
ADMesh	Processes triangulated solid meshes in STL file format	Anthony D. Martin (amartin@engr.csulb.edu)	Via http from http://www.varlog.com/products/admesh/	Has been compiled on SunOS 4.1.3 and IRIX 5.2 but should compile and run on any Unix-like system, and perhaps even under DOS or Windows.
AMR1D	Adaptive Mesh Refinement for a fluid element code. Solves the 1D Euler equations on an unstructured grid. Primarily intended as a learning tool.		Download from http://sdcd.gsfc.nasa.gov/ESS/exchange/contrib/de-fainchtein/adaptive_mesh_refinement.html	
BAMG	Two-dimensional mesh generation and refinement.	Frederic Hecht. English documentation is available	The C++ source may be downloaded from the BAMG website	
Casca	A 2D mesh generator and preprocessor for FRANC2D (qv) and FRANC2DL (qv). Casca can be used as a mesh generator for other FE programs. Source code may be available on request	Paul Wawrzynek and Luiz Martha at Cornell.	Binaries may be downloaded through the FRANC2DL homepage. See also the CFG Web page. or the Cornell ftp site in the appropriate directory (SUN, HP, IBM etc).	Include Windows95/NT PC's and Sun, HP, DEC Alpha, SGI and IBM workstations.
Chalmesh	A 3D overlapping grid generator.	Anders Peterson at the Chalmers University of Technology.	Source code compilation requires the HDF library and OpenGL or Mesa 3D. Compiled versions for Sun/Solaris and PC/Linux as well as other platforms (SGI, HP, DEC).	See the Chalmesh homepage.

Software:	*Description*:	*Author :*	*Availability*:	*Platforms*:
COG	COGeometry is a 2D/3D Delaunay grid generation package including local mesh refinement. COG is intended as an eventual replacement for IBG (qv).	Ilja Schmelzer	The package may be downloaded via the COG Webpage.	Source code (C++) available. Suitable for Unix/Linux OS.
CUBIT	CUBIT is a mesh generating program for 2D and 3D grids. Both a graphical and command line interface are available. Grids can be exported to a number of finite element packages, including SEACAS (qv), ABAQUS (qv), LS-DYNA (qv) and Patran (qv).	Sandia National Laboratories	Linux, Solaris, HP-UX, IRIX, MicroSoft Windows, MacOS X	While the software can be used freely by approved government and academic sites, there is a $300.00 distribution fee, and a license must be obtained. See the licensing page for more information.
EasyMesh	Delaunay-based mesh generator. Unstructured high quality 2D mesh generation in general domains	Bojan Niceno	From the web site http://www-dinma.univ.trieste.it/~nirftc/research/easymesh/	
Genie++	Genie++ multiblock, 3D, structured grid generation package. Genie2D is a PC version for MS-Windows, which will not generate 3D grids	Primary author Bharat K. Soni. Others include Nadesan Narenthiran, Ming-Hsin Shih, Hugh Thornburg, and Tzu-Yi Yu.	Outside the U.S., only executables are available. Contact Dr. Soni for more information.	Silicon Graphics (SGI) workstations with GL, or Sun workstations with X11 (X-Genie++). Genie2D for Windows 3.0 or higher.

***Software*:**	***Description*:**	***Author* :**	***Availability*:**	***Platforms*:**
GEOMPACK	Widely used 2D triangular / 3D tetrahedral finite element mesh generation programs. Postscript user's guide and other documentation available. The newer version does not include source and is licensed as freeware	Barry Joe (bjoe@allstream.net)	See the homepage for the new version Geompack++ (supersedes Geompack90).	The older version ran under Unix. The newer one runs under Windows NT/95/98.
GiD	GiD is a pre/post processing environment (including mesh generation) which can be adapted to the requirements of specific finite element/difference/volume/ point analysis programs. GiD can be configured by the user to write data in any format, hence it is compatible with all FE software provided the user defines the connection by the means of a problem type. GiD generates all the information (2D/3D structured/unstructured meshes, boundary and loading conditions, material types, result visualisation etc) required for the analysis of any problem using numerical methods.	International Centre for Numerical Methods in Engineering.	UNIX workstations (Silicon Graphics, DEC Alpha, Sun, HP, AIX) and PC's running Windows 95/NT or Linux.	GiD is a fully-featured but low-cost commercial package. For details on pricing etc see the GiD homepage. A trial of the professional version of GiD is available for download from this website. A free academic version (limited number of elements) may also be downloaded.
GrAL - Grid Algorithms Library	GrAL is a generic library for grid data structures and algorithms operating on them. In this context, "generic" means that algorithms are decoupled from grid data structures, and can be used with any grid data structure. GrAL has been inspired by the C++ STL.	Guntram Berti	Available via the web at http://www.math.tu-cottbus.de/~berti/gral. This is released under an Open Source license, and includes C++ source code and documentation. The current version is 0.2.	

Software:	***Description***:	***Author*** :	***Availability***:	***Platforms***:
GRUMMP	Generation and Refinement of Unstructured Mixed-element Meshes in Parallel. GRUMMP features simplicial mesh generation in 2D,3D; mesh quality assessment; mesh manipulation; mesh improvement etc. A User's Guide is available	Carl Ollivier-Gooch at UBC	GRUMMP is distributed in source (C/C++) form. It has been successfully built on systems running Linux 2.0.x, SunOS 4.1.4 and IRIX 6.2.	By ftp from ftp://tetra.mech.ubc.ca/pub/GRUMMP/ or via the GRUMMP homepage
gmsh	Gmsh is a 3D meshing program. Line, surface and volume meshing are available, using lines, triangles and tetrahedral. Requires the OpenGL library. Also does post processing of scalar or vector fields. Released under the GPL Complete reference manual is available the code can be compiled natively on Mac OS X (in addition to Windows and UNIX/Linux) Gmsh provides its own scriptable CAD engine and post-processor; meshes can now also be generated by extrusion (i.e. generate quadrilaterals, hexahedra and prisms). See also GetDP.	Christophe Geuzaine and Jean-Francois Remacle	Several UNIX systems, including Linux, Solaris, Tru64, AIX, HPUX, Mac OS X and IRIX. Also available on Windows 95/98/NT.	Via the web page at http://www.geuz.org/gmsh/.

Appendix B. Script listing of the utility to output grid information from GID software for further processing by a utility in FORTRAN.

```
Elements
Total:  *nelem
ElemID    Face1    Face2    Face3    Face4    Face5    Face6
*set elems(all)
*loop elems
*Set var face1=((ElemsNum-1)*6+1)
*Set var face2=((ElemsNum-1)*6+2)
*Set var face3=((ElemsNum-1)*6+3)
*Set var face4=((ElemsNum-1)*6+4)
*Set var face5=((ElemsNum-1)*6+5)
*Set var face6=((ElemsNum-1)*6+6)
*format "%6i%8i%8i%8i%8i%8i%8i"
*ElemsNum *Face1 *Face2 *Face3 *Face4 *Face5 *Face6
*end elems

Faces
Total:  *Operation(6*nelem)
Face |Elem |Node Stat       Value |Node Stat       Value |Node
Stat       Value |Node Stat       Value |
*set elems(hexahedra)
*set var face=0
*set var Stat1=0
*set var Val1=0.0
*set Cond Flux
*loop elems
*set var face=face+1
*set var Nod1=elemsconec(1)
*set var Nod2=elemsconec(4)
*set var Nod3=elemsconec(3)
*set var Nod4=elemsconec(2)
*format "%4i%5i%5i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e"
*face *ElemsNum *Nod1 *Stat1 *val1 *Nod2 *Stat1 *Val1 *Nod3 *Stat1
*Val1 *Nod4 *Stat1 *Val1
*set var face=face+1
*set var Nod1=elemsconec(1)
*set var Nod2=elemsconec(5)
*set var Nod3=elemsconec(8)
*set var Nod4=elemsconec(4)
*format "%4i%5i%5i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e"
*face *ElemsNum *Nod1 *Stat1 *val1 *Nod2 *Stat1 *Val1 *Nod3 *Stat1
*Val1 *Nod4 *Stat1 *Val1
*set var face=face+1
*set var Nod1=elemsconec(1)
*set var Nod2=elemsconec(2)
*set var Nod3=elemsconec(6)
*set var Nod4=elemsconec(5)
*format "%4i%5i%5i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e"
*face *ElemsNum *Nod1 *Stat1 *val1 *Nod2 *Stat1 *Val1 *Nod3 *Stat1
*Val1 *Nod4 *Stat1 *Val1
*set var face=face+1
```

```
*set var Nod1=elemsconec(2)
*set var Nod2=elemsconec(3)
*set var Nod3=elemsconec(7)
*set var Nod4=elemsconec(6)
*format "%4i%5i%5i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e"
*face *ElemsNum *Nod1 *Stat1 *val1 *Nod2 *Stat1 *Val1 *Nod3 *Stat1
*Val1 *Nod4 *Stat1 *Val1
*set var face=face+1
*set var Nod1=elemsconec(3)
*set var Nod2=elemsconec(4)
*set var Nod3=elemsconec(8)
*set var Nod4=elemsconec(7)
*format "%4i%5i%5i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e"
*face *ElemsNum *Nod1 *Stat1 *val1 *Nod2 *Stat1 *Val1 *Nod3 *Stat1
*Val1 *Nod4 *Stat1 *Val1
*set var face=face+1
*set var Nod1=elemsconec(5)
*set var Nod2=elemsconec(6)
*set var Nod3=elemsconec(7)
*set var Nod4=elemsconec(8)
*format "%4i%5i%5i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e%4i%4i%12.5e"
*face *ElemsNum *Nod1 *Stat1 *val1 *Nod2 *Stat1 *Val1 *Nod3 *Stat1
*Val1 *Nod4 *Stat1 *Val1
*end elems

Nodes
Total:  *npoin
NodeID             X              Y               Z      Stat
Potential
*Set Cond Potential
*loop nodes
*Set var Status=Cond(1,int)
*Set var Poten=Cond(2,real)
*format "%5i%14.5e%14.5e%14.5e%9i%14.5e"
*NodesNum *NodesCoord(1,real) *NodesCoord(2,real)
*NodesCoord(3,real) *Status *Poten
*end nodes

Known Boundary Fluxes
*Set Var Tot=0
*Set Cond Flux *elems
*loop elems *OnlyInCond
*Set Var Tot=Tot+1
*end elems
Total:  *Tot
Count  Node1   Node2   Node3         Flux
*loop elems *OnlyInCond
*format "%4i%7i%7i%7i%12.5e"
*elemsnum *globalnodes(1) *globalnodes(2) *globalnodes(3) *cond(2)
*end elems

Connectivities
Total:  *nelem
ElemID    Node1    Node2    Node3    Node4    Node5    Node6
Node7    Node8
```

```
*loop elems
*set var Nod1=elemsconec(1)
*set var Nod2=elemsconec(2)
*set var Nod3=elemsconec(3)
*set var Nod4=elemsconec(4)
*set var Nod5=elemsconec(5)
*set var Nod6=elemsconec(6)
*set var Nod7=elemsconec(7)
*set var Nod8=elemsconec(8)
*format "%6i%8i%8i%8i%8i%8i%8i%8i%8i"
*ElemsNum *Nod1 *Nod2 *Nod3 *Nod4 *Nod5 *Nod6 *Nod7 *Nod8
*end elems

Fluxes
Total:  *Operation(4*face)
Count  Face   FromNode     ToNode
*set var face=0
*loop elems
*set var face=face+1
*set var From1=elemsconec(5)
*set var From2=elemsconec(8)
*set var From3=elemsconec(7)
*set var From4=elemsconec(6)
*set var To1=elemsconec(1)
*set var To2=elemsconec(4)
*set var To3=elemsconec(3)
*set var To4=elemsconec(2)
*format "%5i%5i%10i%10i"
*Operation(4*face-3) *face *From1 *To1
*format "%5i%5i%10i%10i"
*Operation(4*face-2) *face *From2 *To2
*format "%5i%5i%10i%10i"
*Operation(4*face-1) *face *From3 *To3
*format "%5i%5i%10i%10i"
*Operation(4*face) *face *From4 *To4
*set var face=face+1
*set var From1=elemsconec(7)
*set var From2=elemsconec(3)
*set var From3=elemsconec(2)
*set var From4=elemsconec(6)
*set var To1=elemsconec(8)
*set var To2=elemsconec(4)
*set var To3=elemsconec(1)
*set var To4=elemsconec(5)
*format "%5i%5i%10i%10i"
*Operation(4*face-3) *face *From1 *To1
*format "%5i%5i%10i%10i"
*Operation(4*face-2) *face *From2 *To2
*format "%5i%5i%10i%10i"
*Operation(4*face-1) *face *From3 *To3
*format "%5i%5i%10i%10i"
*Operation(4*face) *face *From4 *To4
*set var face=face+1
*set var From1=elemsconec(8)
*set var From2=elemsconec(4)
```

```
*set var From3=elemsconec(3)
*set var From4=elemsconec(7)
*set var To1=elemsconec(5)
*set var To2=elemsconec(1)
*set var To3=elemsconec(2)
*set var To4=elemsconec(6)
*format "%5i%5i%10i%10i"
*Operation(4*face-3) *face *From1 *To1
*format "%5i%5i%10i%10i"
*Operation(4*face-2) *face *From2 *To2
*format "%5i%5i%10i%10i"
*Operation(4*face-1) *face *From3 *To3
*format "%5i%5i%10i%10i"
*Operation(4*face) *face *From4 *To4
*set var face=face+1
*set var From1=elemsconec(5)
*set var From2=elemsconec(1)
*set var From3=elemsconec(4)
*set var From4=elemsconec(8)
*set var To1=elemsconec(6)
*set var To2=elemsconec(2)
*set var To3=elemsconec(3)
*set var To4=elemsconec(7)
*format "%5i%5i%10i%10i"
*Operation(4*face-3) *face *From1 *To1
*format "%5i%5i%10i%10i"
*Operation(4*face-2) *face *From2 *To2
*format "%5i%5i%10i%10i"
*Operation(4*face-1) *face *From3 *To3
*format "%5i%5i%10i%10i"
*Operation(4*face) *face *From4 *To4
*set var face=face+1
*set var From1=elemsconec(6)
*set var From2=elemsconec(2)
*set var From3=elemsconec(1)
*set var From4=elemsconec(5)
*set var To1=elemsconec(7)
*set var To2=elemsconec(3)
*set var To3=elemsconec(4)
*set var To4=elemsconec(8)
*format "%5i%5i%10i%10i"
*Operation(4*face-3) *face *From1 *To1
*format "%5i%5i%10i%10i"
*Operation(4*face-2) *face *From2 *To2
*format "%5i%5i%10i%10i"
*Operation(4*face-1) *face *From3 *To3
*format "%5i%5i%10i%10i"
*Operation(4*face) *face *From4 *To4
*set var face=face+1
*set var From1=elemsconec(1)
*set var From2=elemsconec(2)
*set var From3=elemsconec(3)
*set var From4=elemsconec(4)
*set var To1=elemsconec(5)
*set var To2=elemsconec(6)
```

```
*set var To3=elemsconec(7)
*set var To4=elemsconec(8)
*format "%5i%5i%10i%10i"
*Operation(4*face-3) *face *From1 *To1
*format "%5i%5i%10i%10i"
*Operation(4*face-2) *face *From2 *To2
*format "%5i%5i%10i%10i"
*Operation(4*face-1) *face *From3 *To3
*format "%5i%5i%10i%10i"
*Operation(4*face) *face *From4 *To4
*end elems
```

Appendix C. Script listing in FORTRAN of the utility to convert output from GID into a form suitable for the developed Green element method software

```
C
C -------------------------------------------------------------------
C  THIS PROGRAM RE-WRITES TETRAHEDRA OUTPUT FROM GID
C   FOR USE IN POTENTIAL AND FLUX COMPUTATIONS.
C
C  40 Elements, 40*4=160 Faces, 40*4=160 nodes, 40*12=480 Fluxes
C    ALT = Alternative Face Identities

        PROGRAM ReWrite

        USE MSFLIB
        USE MSFNLS

C ****************** DECLARATIONS

c **Element variables
        INTEGER Elems(40), ElemFaces(40,4)
c **Face Variables
        Character Sig*20(160)
        INTEGER Faces2(160),RefElem2(160),FaceNodes2(160,3)
        INTEGER Faces(160),RefElem(160),FaceNodes(160,3)
        INTEGER Alt(160),Renum2(160),Renum(160)
        Integer QStatus2(480)
        Real DFI2(480)
        Integer QStatus(480)
    Real DFI(480)
c **Node Variables
        INTEGER AllNodes(160),UStatus(480)
        Real X(160),Y(160),Z(160),FI(160)


        CHARACTER
FILENAME*12,FILEIN*12,FILEOUT*12,FILEOUT2*12,TString*10
        CHARACTER St1*5,St2*5,St3*5
        Integer Nd1, Nd2, Nd3, P1, P2, P3

        Integer NElem
        Logical Same
        Integer NFaces2,NFaces,JJ


    Integer NNodes
```

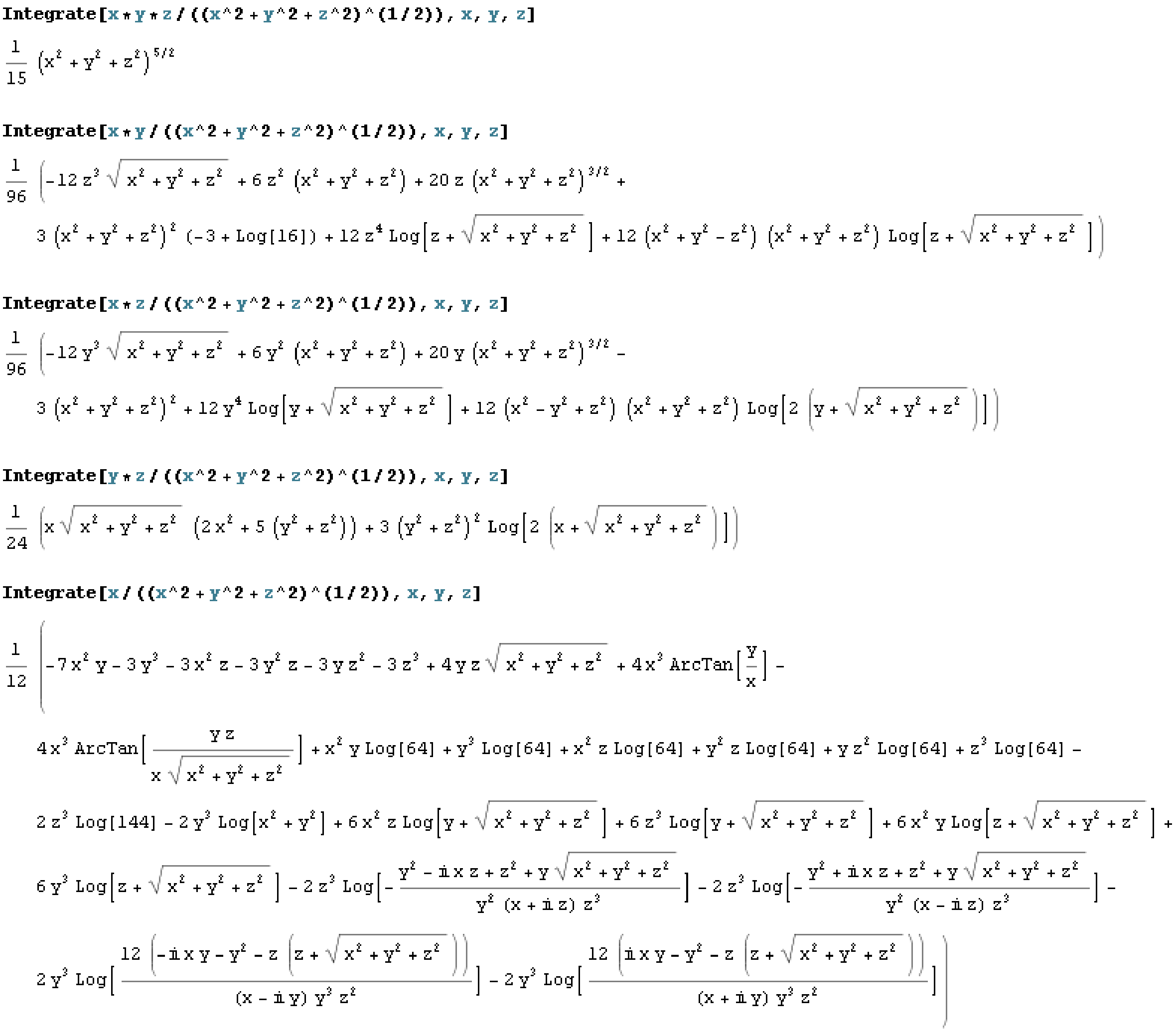
```
C ****************** READ THE DATA
    WRITE(*,100)
 100    FORMAT(2X,'Enter Name of Output File from GID =>',/)
    READ(*,*)FILENAME
        FILEIN=Trim(FILENAME) // '.dat'
        FILEOUT=Trim(FILENAME) // '.txt'
        FILEOUT2=Trim(FILENAME) // '.res'

    OPEN(5,FILE=FILEIN,STATUS='OLD')
        COUNT=DELFILESQQ(FILEOUT)
        COUNT=DELFILESQQ(FILEOUT2)
        OPEN(6,FILE=FILEOUT,STATUS='NEW')
        OPEN(7,FILE=FILEOUT2,STATUS='NEW')

        READ(5,*)TString             !Read 'Elements'
        READ(5,150)TString,NElem   !NElem
 150 FORMAT(A6,I6)
        Write(*,*) TString,NElem
        Read(5,160)TString                         !Read Element labels
 160 Format(A6)
        Write(*,*) TString

    DO 200 I=1,NElem
        Read(5,140) Elems(I)
        1,ElemFaces(I, 1),ElemFaces(I, 2)
   2,ElemFaces(I, 3),ElemFaces(I, 4)
 140    Format(I6,4I9)
        Write(*,140) Elems(I)
        1,ElemFaces(I, 1),ElemFaces(I, 2)
   2,ElemFaces(I, 3),ElemFaces(I, 4)

 200    Continue
        Read(5,160)TString                                !A Blank line

        READ(5,160)TString                                 ! "Faces"
        READ(5,150)TString,NFaces2

        Write(*,*) TString,NFaces2

        Read(5,160)TString                                !Read Face Labels
        Write(*,*) TString

    DO 300 I=1,NFaces2
        Read(5,310) Faces2(I),RefElem2(I)
   1,Nd1,QStatus2(3 * I - 2),DFI2(3 * I - 2)
```

```
    2,Nd2,QStatus2(3 * I - 1),DFI2(3 * I - 1)
    3,Nd3,QStatus2(3 * I),DFI2(3 * I)
     FaceNodes2(I, 1) = Nd1
     FaceNodes2(I, 2) = Nd2
     FaceNodes2(I, 3) = Nd3

         Write(*,310) Faces2(I),RefElem2(I)
    1,Nd1,QStatus2(3 * I - 2),DFI2(3 * I - 2)
    2,Nd2,QStatus2(3 * I - 1),DFI2(3 * I - 1)
    3,Nd3,QStatus2(3 * I),DFI2(3 * I)
  300    Continue
  310 Format(I5,I6,3(I5,I5,ES13.5))

         Read(5,160)TString     !Should read blank
         Read(5,160)TString
         Write(*,*) TString


         READ(5,150)TString,NNodes
         Write(*,*) TString,NNodes

         Read(5,160)TString
         Write(*,*) TString

     DO 350 I=1,NNodes
     Read(5,360) AllNodes(I),X(I),Y(I),Z(I),UStatus(I),FI(I)
     Write(*,360) AllNodes(I),X(I),Y(I),Z(I),UStatus(I),FI(I)

  350    Continue
  360 Format(I5,3E16.5,I8,E15.5)
c      Pause
C ***************CREATE FACE SIGNATURES
     DO 500 I=1,NFaces2
     Nd1 = FaceNodes2(I,1)
     Nd2 = FaceNodes2(I,2)
     Nd3 = FaceNodes2(I,3)
         P1=Nd1
C Find the smallest
     If (P1>Nd2) P1=Nd2
     If (P1>Nd3) P1=Nd3
C Swap
     If (P1 .eq. Nd2) Nd2=Nd1
     If (P1 .eq. Nd3) Nd3=Nd1
     Nd1=P1
```

```
      P2 = Nd2
C Find the smallest
      If (P2>Nd3)  P2=Nd3
C Swap
      If (P2 .eq. Nd3)  Nd3=Nd2
      Nd2=P2

      P3=Nd3

          write(St1, '(I5)') P1
          write(St2, '(I5)') P2
          write(St3, '(I5)') P3
      Sig(I)=(Trim(St1)//Trim(St2)//Trim(St3))
c         Pause
  500 CONTINUE
C ***************FIND ALTERNATIVE NODE IDENTITIES
      Alt(1) = Faces2(1)
      Do 550 I=2,NFaces2
      Do 530 K=1,(I - 1)
          Same = MBLEQ(Sig(I),Sig(K))
          If (.not. Same) goto 520
      Alt(I) = Faces2(K)    !if same
      Exit
  520 Alt(I)=Faces2(I)
  530 Continue
  550 Continue
C ***************RENUMBER
      JJ=0
          do 600 I=1,NFaces2
      If (Alt(I) .ne. Faces2(I)) Goto 580
          JJ=JJ+1
          Renum2(I)=JJ
          goto 590
580   continue
          do 585 K=1,I
           if (Alt(I) .eq. Faces2(K)) Renum2(I)=Renum2(K)
585       Continue
590   Continue
600   continue


C *********** EXTRACT UNIQUE FACES FROM FACES2 TO FACES
      NFaces=0
           DO 630 I=1,NFaces2
          If (Faces2(I) .ne. Alt(I)) goto 620
          NFaces=NFaces+1
          Faces(NFaces)=Renum2(I)
          Renum(NFaces)=Renum2(I)

      RefElem(NFaces) = RefElem2(I)
      FaceNodes(NFaces,1) = FaceNodes2(I,1)
      FaceNodes(NFaces,2) = FaceNodes2(I,2)
      FaceNodes(NFaces,3) = FaceNodes2(I,3)

      QStatus(3 * NFaces - 2) = QStatus2(3 * I - 2)
      QStatus(3 * NFaces - 1) = QStatus2(3 * I - 1)
      QStatus(3 * NFaces) = QStatus2(3 * I)

      DFI(3 * NFaces - 2) = DFI2(3 * I - 2)
      DFI(3 * NFaces - 1) = DFI2(3 * I - 1)
      DFI(3 * NFaces) = DFI2(3 * I)



620      Continue
630   continue
c ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
          do 640 I=1,NFaces
      Write(7,*) I,Faces(I),Renum(I)
640   continue
c ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

C ****************** DELETE EXTRA FACES
c *****On the Elements
      Do 700 I=1,NElem
      do 680 J=1,4          !4 faces for a tetrahedron

      do 650 K = 1,NFaces2
      If (ElemFaces(I,J) .EQ. Faces2(K)) ElemFaces(I,J)=Renum2(K)
 650  Continue
 680  Continue
 700  Continue


C ****************** RE-WRITE THE DATA
      Write(6,805) 'Elements'
 805  Format(A8)
 806  Format(A45)
          Write(6,150) 'Total:',NElem
          Write(6,806) 'ElemID    Face1    Face2    Face3    Face4'
```

```
          Do 830 I=1,Nelem
          Write(6,140) Elems(I)
          1,ElemFaces(I, 1),ElemFaces(I, 2)
   2,ElemFaces(I, 3),ElemFaces(I, 4)
830     Continue
     Write(6,835) ''
     Write(6,835) 'Faces'
835  Format(A5)
          Write(6,150) 'Total:',NFaces
          Write(6,836) 'Face |Elem |Node Stat       Value |Node Stat
          1Value |Node Stat       Value |'
836  Format(A82)
          DO 910 I=1,NFaces
          Write(6,310) Faces(I),RefElem(I)
   1,FaceNodes(I,1),QStatus(3 * I - 2),DFI(3 * I - 2)
   2,FaceNodes(I,2),QStatus(3 * I - 1),DFI(3 * I - 1)
   3,FaceNodes(I,3),QStatus(3 * I),DFI(3 * I)
910  Continue

     Write(6,835) ''
     Write(6,835) 'Nodes'
          Write(6,150) 'Total:',NNodes
          Write(6,936) 'NodeID        X             Y             Z
          1Stat      Potential'
936  Format(A74)
          DO 1010 I=1,NNodes
     Write(6,360) AllNodes(I),X(I),Y(I),Z(I),UStatus(I),FI(I)
1010 Continue



          PAUSE


          END
```

Appendix D. Integrations of the product of the interpolating functions and the fundamental solutions for surface and volume integrals as performed by MATHEMATICA software.



```
Integrate[x*y*z/((x^2+y^2+z^2)^(1/2)), x, y, z]
```

$$\frac{1}{15}\left(x^2+y^2+z^2\right)^{5/2}$$

```
Integrate[x*y/((x^2+y^2+z^2)^(1/2)), x, y, z]
```

$$\frac{1}{96}\Big(-12z^3\sqrt{x^2+y^2+z^2}+6z^2\left(x^2+y^2+z^2\right)+20z\left(x^2+y^2+z^2\right)^{3/2}+ 3\left(x^2+y^2+z^2\right)^2(-3+\mathrm{Log}[16])+12z^4\,\mathrm{Log}\left[z+\sqrt{x^2+y^2+z^2}\right]+12\left(x^2+y^2-z^2\right)\left(x^2+y^2+z^2\right)\mathrm{Log}\left[z+\sqrt{x^2+y^2+z^2}\right]\Big)$$

```
Integrate[x*z/((x^2+y^2+z^2)^(1/2)), x, y, z]
```

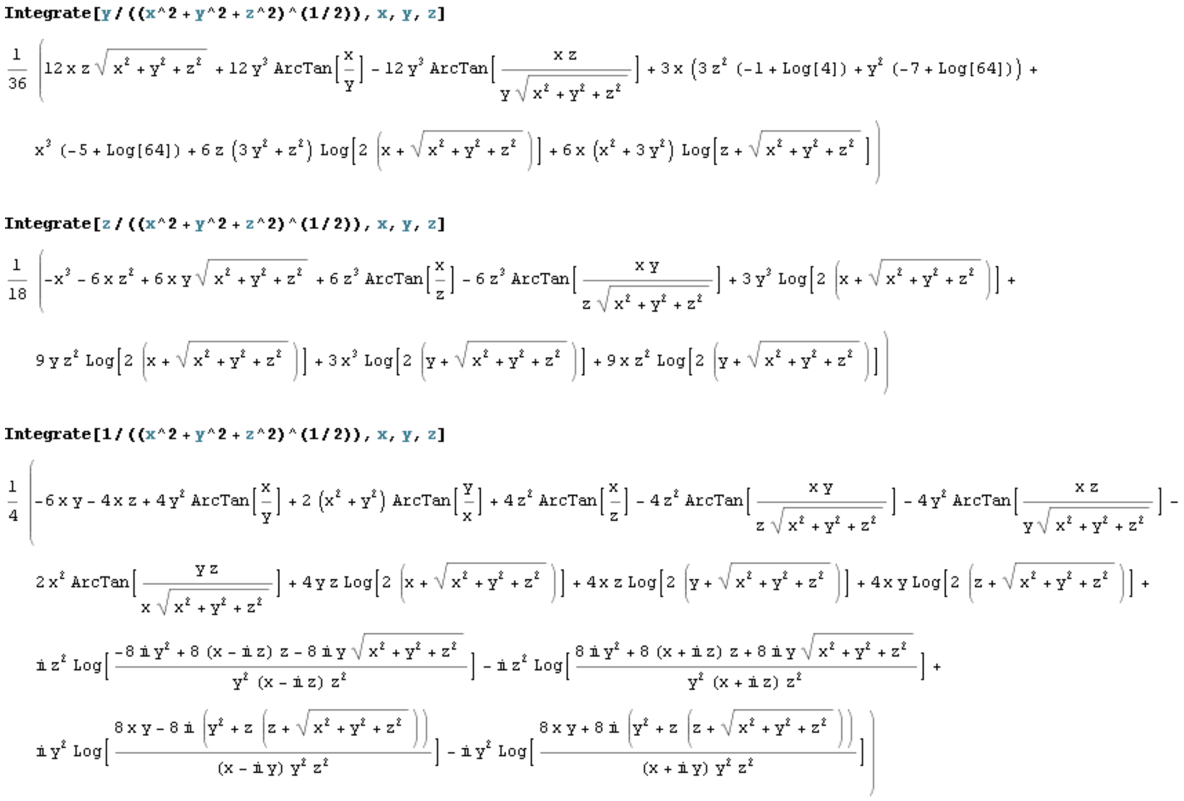
$$\frac{1}{96}\Big(-12y^3\sqrt{x^2+y^2+z^2}+6y^2\left(x^2+y^2+z^2\right)+20y\left(x^2+y^2+z^2\right)^{3/2}- 3\left(x^2+y^2+z^2\right)^2+12y^4\,\mathrm{Log}\left[y+\sqrt{x^2+y^2+z^2}\right]+12\left(x^2-y^2+z^2\right)\left(x^2+y^2+z^2\right)\mathrm{Log}\left[2\left(y+\sqrt{x^2+y^2+z^2}\right)\right]\Big)$$

```
Integrate[y*z/((x^2+y^2+z^2)^(1/2)), x, y, z]
```

$$\frac{1}{24}\left(x\sqrt{x^2+y^2+z^2}\left(2x^2+5\left(y^2+z^2\right)\right)+3\left(y^2+z^2\right)^2\mathrm{Log}\left[2\left(x+\sqrt{x^2+y^2+z^2}\right)\right]\right)$$

```
Integrate[x/((x^2+y^2+z^2)^(1/2)), x, y, z]
```

$$\frac{1}{12}\Bigg(-7x^2y-3y^3-3x^2z-3y^2z-3yz^2-3z^3+4yz\sqrt{x^2+y^2+z^2}+4x^3\,\mathrm{ArcTan}\left[\frac{y}{x}\right]- 4x^3\,\mathrm{ArcTan}\left[\frac{yz}{x\sqrt{x^2+y^2+z^2}}\right]+x^2y\,\mathrm{Log}[64]+y^3\,\mathrm{Log}[64]+x^2z\,\mathrm{Log}[64]+y^2z\,\mathrm{Log}[64]+yz^2\,\mathrm{Log}[64]+z^3\,\mathrm{Log}[64]- 2z^3\,\mathrm{Log}[144]-2y^3\,\mathrm{Log}\left[x^2+y^2\right]+6x^2z\,\mathrm{Log}\left[y+\sqrt{x^2+y^2+z^2}\right]+6z^3\,\mathrm{Log}\left[y+\sqrt{x^2+y^2+z^2}\right]+6x^2y\,\mathrm{Log}\left[z+\sqrt{x^2+y^2+z^2}\right]+ 6y^3\,\mathrm{Log}\left[z+\sqrt{x^2+y^2+z^2}\right]-2z^3\,\mathrm{Log}\left[-\frac{y^2-ixz+z^2+y\sqrt{x^2+y^2+z^2}}{y^2(x+iz)z^3}\right]-2z^3\,\mathrm{Log}\left[-\frac{y^2+ixz+z^2+y\sqrt{x^2+y^2+z^2}}{y^2(x-iz)z^3}\right]- 2y^3\,\mathrm{Log}\left[\frac{12\left(-ixy-y^2-z\left(z+\sqrt{x^2+y^2+z^2}\right)\right)}{(x-iy)y^3z^2}\right]-2y^3\,\mathrm{Log}\left[\frac{12\left(ixy-y^2-z\left(z+\sqrt{x^2+y^2+z^2}\right)\right)}{(x+iy)y^3z^2}\right]\Bigg)$$



```
Integrate[y / ((x^2 + y^2 + z^2)^(1/2)), x, y, z]
```

$$\frac{1}{36}\left(12\,x\,z\sqrt{x^2+y^2+z^2}+12\,y^3\,\text{ArcTan}\left[\frac{x}{y}\right]-12\,y^3\,\text{ArcTan}\left[\frac{x\,z}{y\sqrt{x^2+y^2+z^2}}\right]+3\,x\left(3\,z^2\,(-1+\text{Log}[4])+y^2\,(-7+\text{Log}[64])\right)+x^3\,(-5+\text{Log}[64])+6\,z\left(3\,y^2+z^2\right)\text{Log}\left[2\left(x+\sqrt{x^2+y^2+z^2}\right)\right]+6\,x\left(x^2+3\,y^2\right)\text{Log}\left[z+\sqrt{x^2+y^2+z^2}\right]\right)$$

```
Integrate[z / ((x^2 + y^2 + z^2)^(1/2)), x, y, z]
```

$$\frac{1}{18}\left(-x^3-6\,x\,z^2+6\,x\,y\sqrt{x^2+y^2+z^2}+6\,z^3\,\text{ArcTan}\left[\frac{x}{z}\right]-6\,z^3\,\text{ArcTan}\left[\frac{x\,y}{z\sqrt{x^2+y^2+z^2}}\right]+3\,y^3\,\text{Log}\left[2\left(x+\sqrt{x^2+y^2+z^2}\right)\right]+9\,y\,z^2\,\text{Log}\left[2\left(x+\sqrt{x^2+y^2+z^2}\right)\right]+3\,x^3\,\text{Log}\left[2\left(y+\sqrt{x^2+y^2+z^2}\right)\right]+9\,x\,z^2\,\text{Log}\left[2\left(y+\sqrt{x^2+y^2+z^2}\right)\right]\right)$$

```
Integrate[1 / ((x^2 + y^2 + z^2)^(1/2)), x, y, z]
```

$$\frac{1}{4}\left(-6\,x\,y-4\,x\,z+4\,y^2\,\text{ArcTan}\left[\frac{x}{y}\right]+2\left(x^2+y^2\right)\text{ArcTan}\left[\frac{y}{x}\right]+4\,z^2\,\text{ArcTan}\left[\frac{x}{z}\right]-4\,z^2\,\text{ArcTan}\left[\frac{x\,y}{z\sqrt{x^2+y^2+z^2}}\right]-4\,y^2\,\text{ArcTan}\left[\frac{x\,z}{y\sqrt{x^2+y^2+z^2}}\right]-2\,x^2\,\text{ArcTan}\left[\frac{y\,z}{x\sqrt{x^2+y^2+z^2}}\right]+4\,y\,z\,\text{Log}\left[2\left(x+\sqrt{x^2+y^2+z^2}\right)\right]+4\,x\,z\,\text{Log}\left[2\left(y+\sqrt{x^2+y^2+z^2}\right)\right]+4\,x\,y\,\text{Log}\left[2\left(z+\sqrt{x^2+y^2+z^2}\right)\right]+i\,z^2\,\text{Log}\left[\frac{-8\,i\,y^2+8\,(x-i\,z)\,z-8\,i\,y\sqrt{x^2+y^2+z^2}}{y^2\,(x-i\,z)\,z^2}\right]-i\,z^2\,\text{Log}\left[\frac{8\,i\,y^2+8\,(x+i\,z)\,z+8\,i\,y\sqrt{x^2+y^2+z^2}}{y^2\,(x+i\,z)\,z^2}\right]+i\,y^2\,\text{Log}\left[\frac{8\,x\,y-8\,i\left(y^2+z\left(z+\sqrt{x^2+y^2+z^2}\right)\right)}{(x-i\,y)\,y^2\,z^2}\right]-i\,y^2\,\text{Log}\left[\frac{8\,x\,y+8\,i\left(y^2+z\left(z+\sqrt{x^2+y^2+z^2}\right)\right)}{(x+i\,y)\,y^2\,z^2}\right]\right)$$