Approximate Closed-Form Solution of Diffusion Equation

by

Saba Zehra

Department of Mathematics,

School of Natural Sciences (SNS),

National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

2015

Approximate Closed-Form Solution of Diffusion Equation

by

Saba Zehra

A dissertation submitted in partial fulfillment of the requirements
for the degree of Master of Philosophy in Mathematics

Supervised by

Dr. Azad Akhter Siddiqui

Department of Mathematics,
School of Natural Sciences (SNS),
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

2015

FORM TH-4

National University of Sciences & Technology

M.Phil THESIS WORK

We hereby recommend that the dissertation prepared under our supervision by: SABA ZEHRA, Regn No. NUST201361950MSNS78013F Titled: Approximate Closed - Form Solution of Diffusion Equation be accepted in partial fulfillment of the requirements for the award of **M.Phil** degree.

Examination Committee Members

1. Name: Dr. Tooba Feroze Signature: ________

2. Name: Dr. Muhammad Asif Farooq Signature: ________

3. Name: ________ Signature: ________

4. Name: Dr. Muhammad Amer Qureshi Signature: ________

Supervisor's Name: Prof. Azad. A. Siddiqui Signature: ________

Head of Department

28/08/15
Date

COUNTERSINGED

Date: 28/8/15

Dean/Principal

Dedicated to

Hazrat Muhammad (PBUH)

Who Guided Me Towards the Path of "Knowledge"

And

My

Beloved Parents

Acknowledgements

Only those fear Allah, from among His servants, who have knowledge. Indeed, Allah is Exalted in Might and Forgiving (Quran 35:28).

In the name of Allah, the Most Gracious and the Most Merciful. Alhamdulillah, all praises to Allah almighty for the strengths and his blessing in completing this thesis. All the respects are for the last Holy prophet Muhammad (S. A. W), the creator of peace and love who's moral and spiritual teachings always enlightens the heart of humanity.

I would like to express my sincere gratitude to my supervisor Prof. Azad. Akhtar. Siddiqui for his assistance, motivation and immense knowledge. His guidance helped me in all time of research and writing of this thesis.

Further, I am thankful for the Guidance of Examination Committee members Dr. Muhammad Asif Farooq and Dr. Tooba Feroze for their help and constant support. In the end, I would also like to thank my family especially my parents for their unconditional support and encouragement.

Saba Zehra

Abstract

Symmetry method is a very useful tool to solve a variety of linear and nonlinear differential equations. This method implies to obtain invarient transformations, and by using these transformations one can reduce a partial differential equations (PDEs) into an ordinary differential equations (ODEs). If the reduced ODE is solved analytically, then one uses the inverse transformations to get the solution of the original PDE. In case, the reduced ODE can not be solved analytically, one generally finds numerical solution. In such cases, a procedure was developed that uses the numerical solution of the reduced ODE to find an approximate closed-form solution of the original PDE [1]. However, it was not evident that the procedure works well. In this thesis, we validate the procedure developed in reference [1]. We solve the diffusion equation numerically and compare the approximate closed-form solution with it. It is found that the procedure works well and gives good approximation at least for the diffusion equation considered.

Contents

List of Figures

Symbols

v Velocity (ms^{-1}).

a Acceleration (ms^{-2}).

x Distance (m).

t Time (sec).

p Pressure (Pa).

∇ Divergence.

ρ Density (kgm^-3).

k Permeability (m^2).

Chapter 1

Introduction

Mathematical problems can be solved with arithmetic and logical operations by using numerical methods, also known as computer mathematics. We study numerical methods for solving large systems of equations, non-linearities and complicated geometries that are often impossible to solve analytically with standard calculus. Moreover, it provides an efficient way of solving the problems on a computer. By implementing the numerical methods on a computer, we can access the computers powers and limitations as well as control the errors of approximations of large calculation [2]. By using numerical methods and computer programming techniques we can design programs to solve problems which cannot be solved using canned programs.

Mathematical modeling explains more theoretical aspects of the scientific approach, and is a fundamental part of applied mathematics. A mathematical model usually takes the relationship between variables and configures problems and laws of nature. The purpose of modeling is to gain new knowledge, but it should also be in agreement with the known facts. A model is a simple idealization [3] of complex reality. It is created with the aim characterized by the investigation to obtain new knowledge about the real world and the implications of the model. Figure 1.1 shows a flowchart of the modeling process [4].

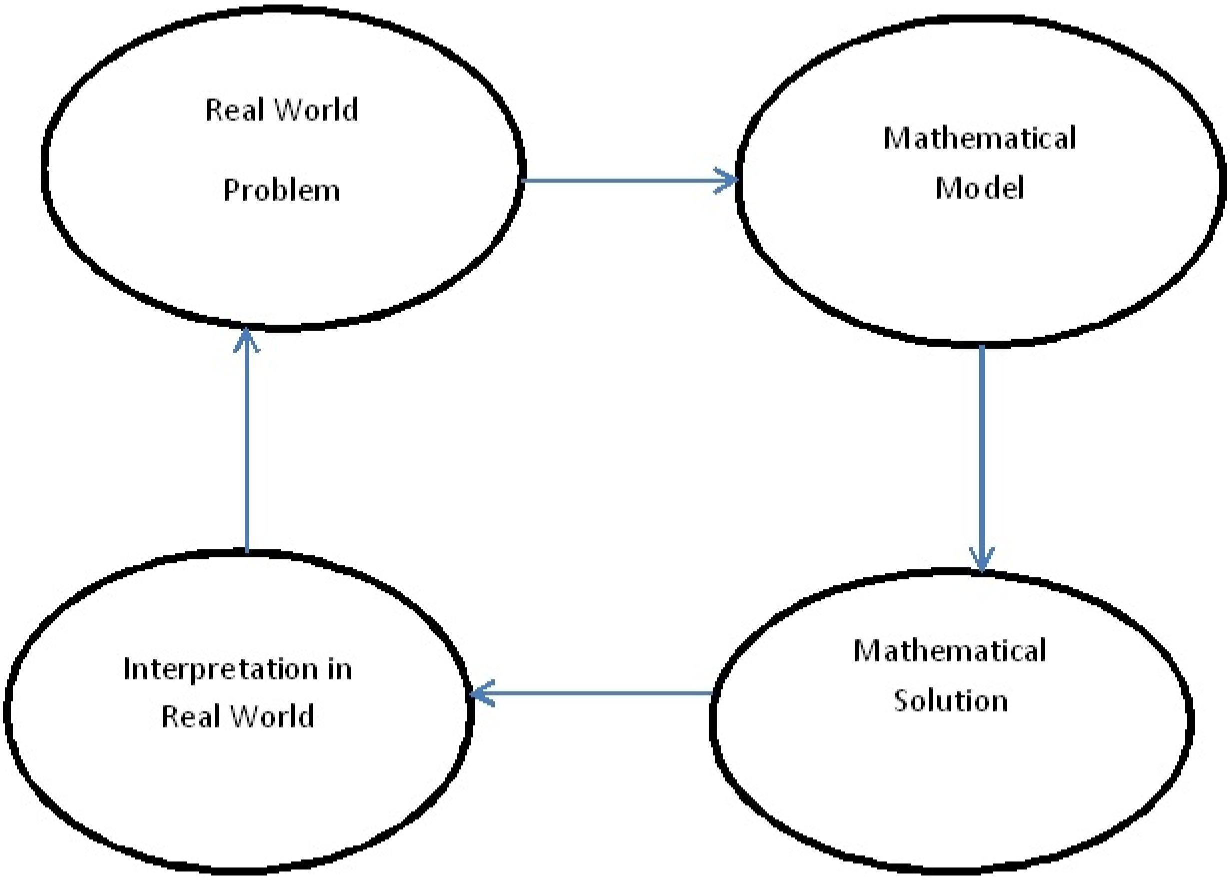

Figure 1.1: Flow chart of the modeling process.

The top arrow in the diagram shows the initial part of the modeling process. Tools such as keen observation, communication skills and human senses are used to solve a problem. It replaces the real world phenomena for employment by an abstract model of mathematical analysis. Right-hand arrow represents the mathematical problem. It is usually a result of the solution. The resulting solution is in mathematical form and it must be reinterpreted back in the original real-world setting. The bottom arrow is deductive part of modeling process. It is mainly concerned with the mathematical solution and it must be related back into the original real world problem. Finally, in the left-hand arrow, interpretation must be checked against reality. It represents the validation part of modeling. Here we check whether the model is appropriate and it agrees with the known aspects of original problem. If this process does not result in the desired knowledge, then repeat the full cycle again to improve the model with alternate assumptions and tools until the satisfactory solution of the problem is obtained.

1.1 Mathematical Modeling of ODEs

In mathematics, a body falling vertically under the gravity leads to a second-order differential equation. The solution of this equation is used to determine the position of body relative to ground. The acceleration of a freely falling body, called gravitational acceleration, is generally denoted by g. On the surface of the earth, its value is approximately 10 ms^{-2}. For bodies falling down freely g is positive and is negative for bodies moving up. Acceleration is the derivative of velocity and velocity is the derivative of distance. Suppose a rock which is tossed from the roof of the building in the upward direction. We assume that the upward direction is positive, so $v_0 > 0$ and then this equation explains the vertical distance which is covered by the body. The negative sign indicates the force which acts on the weight of the body and is directed opposite to the upward acting positive directional force. The mathematical statement is given below as

$$\frac{d^2s}{dt^2} = -g. \tag{1.1}$$

If we assume the height of the building is s_0 and initial velocity of the rock is v_0, then the equation (1.1) satisfies the initial conditions $s(0) = s_0$ and $s'(0) = v_0$. Here, $t = 0$ indicates the initial time when the rock leaves the building and $t = t_1$ is the elapsed time when the rock hits the ground.

1.2 Mathematical Modeling of PDEs

In 1855, Adolf Fick explained the physical phenomena of the diffusion process. The word diffusion is derived from Latin which means "spreading out". In diffusion process, the molecules are moved from an area of high concentration to an area of lower concentration. Consider the following example of diffusion in which some particles are dissolved in a glass of water. It is observed that at the start, all the particles are present at one corner of the glass. After some time particles diffuse randomly and all the particles are uniformly distributed throughout the glass of water as shown in Figure 1.2. A simple example of diffusion process is to place

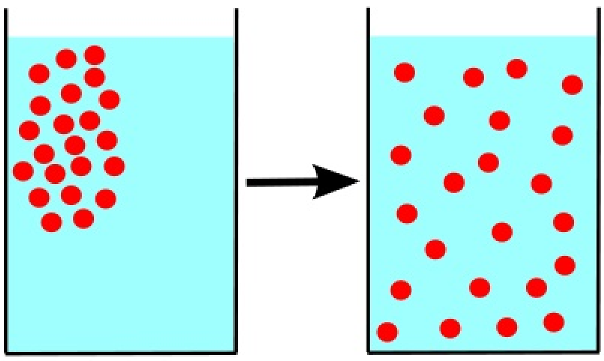

Figure 1.2: Diffusion process.

a drop of dark blue ink into a glass of water. We can observe that at the first, ink is present in a localized region of space. After sometimes, ink diffuses into the water molecules and the water gets bluer throughout the glass. The color of water shows the concentration of ink molecules. Concentration of ink molecules can be represented by using the function c of space and time. When $c = 0$, water contains no ink molecules. When $c(x, y, z, t) = 1$ ink molecules having a small volume surrounds the (x, y, z) spatial points at time t. Initially (i.e., when $t = 0$, $c = 1$) indicates that the droplet of ink is present in a small region. The significant jump from $c = 0$ in the water to $c = 1$ in the ink is smoothed out in time. After some time, we observe the blue color which is evenly distributed throughout the glass of water. This indicates that c has some value between 0 and 1, throughout the glass. This phenomenon is referred as diffusive transport of a substance.

1.3 Derivation of Diffusion Equation

In this section we derive the general form of diffusion equation. We begin with the continuity equation

$$\rho_t + \nabla.(\rho \mathbf{v}) = 0, \tag{1.2}$$

where

- ρ is fluid density,
- $\mathbf{v}$ is the velocity,
- ∇ represents the divergence.

The principle that explains the underground movement of fluid mechanism is called Darcy's law. This law was proposed by the French engineer Henri Darcy in 1856. The ability of fluid to flow through a porous media such as rock is defined by Darcy's law equation. The amount of flow between two points is directly related to the difference in pressure, distance, and interconnectivity of flow pathways between the points.

The mathematical form of Darcy's law is:

$$\mathbf{v} = \frac{-k}{\mu} \nabla p, \tag{1.3}$$

where

- k is permeability,
- μ is viscosity,
- p is pressure.

Using equation (1.3) in equation (1.2), we get

$$\rho_t = -\nabla\big(\rho\frac{-k}{\mu}\nabla p\big). \tag{1.4}$$

In equation (1.3) we see that velocity is related to pressure and negative sign shows that the flow is moving from higher to the lower pressure. In equation (1.2) we see that density is related to velocity. To close the system in a single equation, we use the equation of state as

$$\rho = \rho_0 p^{\gamma}, \tag{1.5}$$

where ρ_0 and γ are constants with $0 < \gamma \leq 1$.
Using equation (1.5) in equation (1.4), we get

$$\rho_t = \nabla\big[\rho(\frac{k}{\mu})\nabla(\frac{\rho}{\rho_0})^{\frac{1}{\gamma}}\big], \tag{1.6}$$

or

$$\rho_t = \frac{k}{\mu}\frac{1}{\rho_0^{\frac{1}{\gamma}}}\frac{1}{\gamma}\nabla(\rho^{\frac{1}{\gamma}}\nabla\rho). \tag{1.7}$$

Notice that, $(\rho^{\frac{1}{\gamma}}\nabla\rho)$ can be written as:

$$\rho^{\frac{1}{\gamma}}\nabla\rho = \frac{\nabla(\rho^{\frac{1}{\gamma}+1})}{(\frac{1}{\gamma}+1)}, \tag{1.8}$$

equation (1.7) becomes

$$\rho_t = \frac{k}{\mu}\frac{1}{\rho_0^{\frac{1}{\gamma}}}\frac{1}{\gamma+1}\nabla^2(\rho^{\frac{1}{\gamma}+1}). \tag{1.9}$$

Equation (1.9) is the general diffusion equation.

1.4 Application of Diffusion Equation

In this section we present several applications of diffusion equation. Many authors have considered this equation for solving the problems in various fields of science. Some of the applications are:

- There are many industrial reactions in which we use catalysts containing small amount of noble metals dispersed in a porous inert material like silica, to speed up the reaction [5]. We can describe the diffusion process and the reaction in the pores in the catalyst by using the diffusion coefficients.

- A carrier is a mobile and reactive species that reacts with diffusing solutes, present in membranes to facilitate the transport of substances across the membrane. Such membranes are used to separate the copper ions from industrial waste and carbon dioxide from coal gas. The diffusion process across these membranes is highly selective and does not change linearly with the concentration difference across them. Fundamental models of diffusion coefficients can be used to describe the facilitated transport across the membranes.

- When food products are spray-dried, the flavor of the food is lost. Spray-drying is a complex industrial-scale process. By using this process, we can inhibit the diffusion of the flavor compounds by forming a tight gel like skin on the surface of drying food. We can describe the flavor retention in food products by seizing the sizes of the pores in the flavor compounds using mass transfer coefficients.

Chapter 2

MATLAB Tools: *bvp4c* and PDE Toolbox

Ordinary differential equations (ODEs) play major role in applications of science and engineering. They arise in wide variety of engineering applications for example electromagnetic theory, computational fluid dynamics, etc. These equations can be typically solved using either analytical or numerical methods. Many of the ODEs arising in real life application can not be solved analytically or we can say that their analytic solution does not exist. For such types of problems certain numerical methods exists.

Partial differential equations (PDEs) occur in many disciplines like Physics, Chemistry, Biology and Mathematics. In PDEs the distinction between linear and nonlinear equations is extremely important. Linear partial differential equations can be solved easily by using particular methods such as Fourier series, separation of variable, Laplace transform etc. However, obtaining exact solutions of nonlinear PDEs is generally difficult and often not possible. To find the exact solution of nonlinear PDEs there is no organized theory. Therefore, one generally looks for the numerical solutions.

There is a lot of work on finding numerical solution of PDEs and a number of methods have been developed to obtain numerical solution of PDEs, for example finite difference method (FDM), finite element method (FEM), finite volume method (FVM). Based on different numerical techniques a number of computer packages have been developed, for example **MATLAB**, **MATHEMATICA**, **MAPLE** etc. There are a lot of tools in these packages, for different types of numerical calculations, including tools for solving ODEs and PDEs. In the following subsections of this chapter we discuss the **MATLAB** tools *bvp4c* and PDE Toolbox, which have been used in our work to solve ODEs and PDEs.

2.1 Solving Boundary Value Problems in MATLAB with *bvp4c*

The boundary value problems (BVPs) for ODEs can be solved using **MATLAB** built-in-function *bvp4c*. Following are the steps to solve BVP on *bvp4c*.

- Convert the higher order ODE into a system of first order ODE in the form

$$\frac{dz}{dx} = z' = h(x, z). \tag{2.1}$$

- Assume that the given BVP is defined on the interval [a, b], then write the boundary conditions as:

  $z(a) = z_a, \qquad z(b) = z_b.$

- Choose an initial guess for the solution.

- *bvp4c* starts finding solution with an initial guess supplied at initial mesh points. For more detail see reference [6].

2.2 PDE Toolbox

The PDE Toolbox is used as a powerful tool for the study and solution of PDEs in two dimensions, one space and one time. The Finite Element Method (FEM) is used to discretize the equations. The main objectives of the PDE Toolbox are:

- Define a PDE problem with a 2-D regions, PDE coefficients and boundary conditions.
- Solve the PDE problem numerically by discretizing the equations, generating un-structured meshes, and approximating the solution.
- Interpreting the results.

Using PDE Toolbox, one can solve the problem in following steps:

2.2.1 Defining a PDE in PDE Toolbox

After defining a PDE problem by using the general form of the PDE equation, we solve the problem. We must also know the geometry of the problem that needs to be solved. For simple geometries that can be drawn manually in the Graphical User Interface (GUI) include any intersection, union or difference of shapes that include squares, rectangle, ellipses and irregular shaped polygons. Next, boundary conditions need to be specified for edges of the object and the edges of the subdomains. The boundary conditions are:

Dirichlet Boundary Condition

It is defined on the boundary of the domain by specifying the value of the solution.

Neumann Boundary Condition

It is defined on the boundary of the domain by specifying the value of the derivative of the solution.

2.2.2 Solution of PDE

We can solve the PDE using PDE **MATLAB** Toolbox in order to solve the problem, a triangular mesh must be initialized and refined carefully. If the mesh is initialized but not refined or improved, some areas of the geometry may have less accurate estimations of the solution than desired. Computational effort will increase if the mesh is heavily refined. Once the triangular mesh is generated in the correct orientation, then **MATLAB** solves the PDE instantly and we get the accurate solution of the problem.

2.2.3 Visualizing the Result on a Graph

The solution of PDE can be interpreted in different ways. If there is any default option, then we see a color grid with a colorbar axis. We can also interpret the results by the use of contour lines, animation mode and a 3D plot of the given problem.

2.2.4 pdepe

In this subsection, we discuss the **MATLAB** command pdepe, which is used to solve PDEs.

Syntax

$$sol = pdepe\big(m, pdex1pde, pdex1ic, pdex1bc, xmesh, tspan\big). \tag{2.2}$$

Here,

$\boldsymbol{m}$ **:** It represents a parameter that correspond to the co-ordinate system of the given problem.

- value of m in Cartesian $= 0$.

- value of m in Cylindrical $= 1$.
- value of m in Spherical $= 2$.

pdex1pde **:** It represents a function which defines the components of the PDE.

pdex1ic **:** It represents a function which defines the initial conditions.

pdex1bc **:** It represents a function which defines the boundary conditions.

xmesh **:** It represents a set of vectors $[x_0, x_1, ..., x_n]$ which specifies the points upon which numerical solution is calculated for every value in *tspan*. The elements of *xmesh* should meet the condition as $x_0 < x_1 < ... < x_n$, and its length should be greater than equal to 3.

tspan **:** It represents a set of vectors $[t_0, t_1, ..., t_f]$ which specifies the points upon which numerical solution is calculated for every value in *xmesh*. The elements of *tspan* should specify the following condition such as $t_0 < t_1 < ... < t_f$, and its length should be greater than equal to 3.

In PDE Toolbox the following general form is used:

$$c\left(x, t, u, \frac{\partial u}{\partial x}\right)\frac{\partial u}{\partial t} = x^{-m}\frac{\partial}{\partial x}\left(x^m f\left(x, t, u, \frac{\partial u}{\partial x}\right)\right) + s\left(x, t, u, \frac{\partial u}{\partial x}\right). \tag{2.3}$$

In the PDE Toolbox the PDE holds for $t_0 \leq t \leq t_f$ and $a \leq x \leq b$. The interval of a and b should be finite. In equation (2.3) $m > 0$, then $a \geq 0$.

The coupling of the partial derivatives with respect to time is restricted to multiplication by a diagonal matrix $c\left(x, t, u, \dfrac{\partial u}{\partial x}\right)$. The diagonal elements of this matrix are either identically zero or positive [7].

The solution components must satisfy the initial condition of the form

$$u(x, t_0) = u_0(x), \tag{2.4}$$

for all x and $t = t_0$ and the solution components must satisfy a boundary condition of the form

$$p(x, t, u) + q(x, t) f\left(x, t, u, \frac{\partial u}{\partial x}\right) = 0, \tag{2.5}$$

for all t and $x = a$ or b.

Elements of q are either identically zero or never zero. The boundary conditions must be expressed in terms of f rather than $\dfrac{\partial u}{\partial x}$. Although, two coefficients are available, but p depends only on u .

When we call the equation (2.2) then **MATLAB** read the script as follows

- m corresponds to 0, 1 or 2.
- $xmesh(1)$ and $xmesh(\text{end})$ corresponds to a and b, respectively.
- $tspan(1)$ and $tspan(\text{end})$ corresponds to t_0 and t_f, respectively.
- $pdex1pde$ computes the terms c, f, and s in equation (2.3). It has the form $[c, f, s] = pdex1pde(x,\ t,\ u,\ dudx)$.

By the use of scalars x and t, vectors u and $dudx$ as an input arguments, we can approximate the solution u as well as its partial derivative with respect to x. The column vectors are c, f and s. The diagonal elements of the matrix c are stored by c as shown in equation (2.3).

- $pdex1ic$ calculates the initial conditions. It has the form

$$u = pdex1ic(x).$$

$pdex1ic$ can evaluate and return to the initial values of the solution components of x by the use of argument x in the column vector u.

- $pdex1bc$ calculates the terms p and q of the boundary conditions in equation (2.5). It has the form

$$[pl, ql, pr, qr] = pdex1bc(xl, ul, xr, ur, t), \tag{2.6}$$

where ul represents the approximate solution at the left boundary $xl = a$ and ur represents the approximate solution at the right boundary $xr = b$. pl and ql are column vectors corresponding to p and q evaluated at xl; similarly pr and qr correspond to xr. pdepe imposes this boundary condition automatically and it ignores values returned in pl and ql.

Chapter 3

Approximate Closed-Form Solution of the Reduced ODE

Mathematical modeling of the physical processes in most fields like diffusion, chemical kinetics, fluid mechanics, wave mechanics and general transport problems is governed by nonlinear PDEs whose exact solutions are difficult to find. So the approach of reduction of PDEs to ODEs is quite important and helps us in the study of various physical processes [8]. A powerful technique for analyzing such nonlinear PDEs and their reduction to ODEs is given by Lie symmetry method. In this method, depending upon the available symmetries of the PDEs under consideration one can:

- reduce the number of independent variables of a PDE.
- reduce the order of an ODE.
- even find the exact solution.

There are very important contributions in the application of symmetry method to initial boundary value problems (IBVPs) [9, 10]. In many cases, a PDE is converted to an ODE by using symmetry method. However, the exact solution of the reduced ODE is not possible to obtain in all cases. In such cases, one solves the reduced ODE numerically. A method to utilize the numerical solution of the reduced ODE to obtain approximate solution of the original PDE is developed in [1], which is discussed in the following section.

3.1 A Method for Generating Approximate Solution of PDEs

In reference [1], a method is presented to obtain approximate solution of an IBVP, using the following steps:

Step 1: Reduce IBVP of PDE to BVP of ODE by applying symmetry transformations.

Step 2: Find numerical solution, v_{num}, of BVP of resulting ODE.

Step 3: Obtain an initial guess, $v_{initial}$, for the approximate solution, $v_{approx}(z)$, of BVP of ODE.

Step 4: Improve the initial approximation, $v_{initial}$, to get the approximate solution, $v_{approx}(z)$, up to the desired level of accuracy.

Step 5: Use the inverse symmetry transformations on $v_{approx}(z)$ to get the approximate solution, $u(x,t)$, of the original PDE.

3.1.1 Example

Consider a transient flow of gas through semi-infinite porous medium [11, 12, 13, 14] initially filled with gas at a uniform pressure $u_0 > 0$, at time $t = 0$, the outflow pressure is suddenly reduced from u_0 to $u_1 > 0$ ($u_1 = 0$ is the case diffusion in to a vacuum) and is then maintained at the lower pressure. The unsteady transient flow of gas is described by a nonlinear partial differential equation. The mathematical equation [15] of the unsteady transient flow of gas through a semi-infinite porous medium can be written as:

$$\nabla^2(u^2) = 2c\frac{\partial u}{\partial t}. \tag{3.1}$$

Here ∇ is divergence, u is pressure, and $c = \frac{\phi\mu}{k}$, where ϕ is porosity, μ is viscosity, and k is permeability. Consider one dimensional case, for $x = 0$ to $x \rightarrow \infty$, then the above equation (3.1) takes the form

$$\frac{\partial}{\partial x}(u\frac{\partial u}{\partial x}) = c\frac{\partial u}{\partial t}, \tag{3.2}$$

subject to the following boundary conditions

$$u(x, 0) = u_0, \tag{3.3}$$

$$u(0, t) = u_1(< u_0), \tag{3.4}$$

$$u(\infty, t) = u_0. \tag{3.5}$$

In equations (3.3)-(3.5) we assume $u_0 = 1$, then we have

$$u(x, 0) = 1, \qquad 0 < x < \infty, \tag{3.6}$$

$$u(0, t) = u_1(< 1), \qquad 0 \leq t < \infty, \tag{3.7}$$

$$u(\infty, t) = 1, \qquad 0 < t < \infty. \tag{3.8}$$

Introducing the following symmetry transformations [11, 12]

$$z = \frac{x}{\sqrt{t}}\left(\frac{c}{4}\right)^{\frac{1}{2}}, \tag{3.9}$$

and

$$v = \alpha^{-1}(1 - u^2), \tag{3.10}$$

with $\alpha = 1 - u_1^2$.

Equation (3.10) can be written as

$$u = \sqrt{1 - \alpha v}. \tag{3.11}$$

Taking partial derivatives of equation (3.9) with respect to x and t respectively, we get

$$\frac{\partial z}{\partial x} = \frac{1}{\sqrt{t}} \left(\frac{c}{4}\right)^{1/2}. \tag{3.12}$$

and

$$\frac{\partial z}{\partial t} = \frac{-x}{2t^{3/2}} \left(\frac{c}{4}\right)^{1/2}, \tag{3.13}$$

Again differentiating equation (3.11) with respect to t, we have

$$\frac{\partial u}{\partial t} = \frac{1}{2\sqrt{1 - \alpha v}} \frac{\partial v}{\partial t},$$

or

$$\frac{\partial u}{\partial t} = \frac{1}{2\sqrt{1 - \alpha v}} \frac{dv}{dz} \frac{\partial z}{\partial t}. \tag{3.14}$$

Substituting equation (3.13) in equation (3.14), we get

$$\frac{\partial u}{\partial t} = \frac{1}{2\sqrt{1 - \alpha v}} \left(\frac{-x}{2t^{2/3}} \left(\frac{c}{2}\right)^{1/2}\right) \frac{dv}{dz}. \tag{3.15}$$

Taking partial derivative of equation (3.11) with respect to x, we get

$$\frac{\partial u}{\partial x} = \frac{1}{2}(1-\alpha v)^{-1/2}\frac{\partial v}{\partial x},$$

or

$$\frac{\partial u}{\partial x} = \frac{1}{2\sqrt{1-\alpha v}}\frac{dv}{dz}\frac{\partial z}{\partial x}. \tag{3.16}$$

By multiplying equations (3.11) and (3.16), we obtain

$$u\frac{\partial u}{\partial x} = \frac{1}{2}\frac{dv}{dz}\frac{\partial z}{\partial x}. \tag{3.17}$$

Substituting equation (3.12) in equation (3.17), we get

$$u\frac{\partial u}{\partial x} = \frac{1}{2\sqrt{t}}\left(\frac{c}{4}\right)^{1/2}\frac{dv}{dz}. \tag{3.18}$$

Partially differentiating equation (3.18) with respect to x, we obtain

$$\frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right) = \frac{1}{2\sqrt{t}}\left(\frac{c}{4}\right)^{1/2}\frac{\partial}{\partial x}\left(\frac{dv}{dz}\right),$$

or

$$\frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right) = \frac{1}{2\sqrt{t}}\sqrt{\frac{c}{4}}\frac{d^2v}{dz^2}\frac{\partial z}{\partial x}. \tag{3.19}$$

Using equation (3.12) in equation (3.19), we have

$$\frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right) = \frac{1}{2\sqrt{t}}\sqrt{\frac{c}{4}}\frac{d^2v}{dz^2}\left(\frac{1}{\sqrt{t}}\left(\frac{c}{4}\right)^{1/2}\right),$$

or

$$\frac{\partial}{\partial x}\left(u\frac{\partial u}{\partial x}\right) = \frac{c}{8t}\left(\frac{d^2v}{dz^2}\right). \tag{3.20}$$

Substituting equations (3.15) and (3.20) in equation (3.2), we obtain

$$\frac{d^2v}{dz^2} = \frac{-8ctx}{4c\sqrt{1-\alpha v}t^{3/2}}\sqrt{\frac{c}{4}}\frac{dv}{dz},$$

or

$$\frac{d^2v}{dz^2} = \frac{-2}{\sqrt{1-\alpha v}}\left(\frac{x}{\sqrt{t}}\sqrt{\frac{c}{4}}\right)\frac{dv}{dz},$$

or

$$v'' + \frac{2z}{\sqrt{1-\alpha v}}v' = 0. \tag{3.21}$$

The transformed boundary conditions are as follows:

$$v(z=0) = 1, \tag{3.22}$$

and

$$v(z\to\infty) = 0. \tag{3.23}$$

3.1.2 Numerical Solution of the Reduced ODE

The equation (3.21) is a nonlinear ODE and its exact solution is difficult to find. Now we find the numerical solution of reduced ODE (3.21) by using **MATLAB** built-in solver *bvp4c*. To apply *bvp4c*, we convert the ODE (3.21) into a system of first order ODEs.

First we define the variables $w_1 = v$, and $w_2 = v'$, then the equation (3.21), in new variables w_1 and w_2, becomes the system of first order ODEs as

$$w_1' = w_2,$$

and

$$w_2' + \frac{2z}{\sqrt{1 - \alpha w_1}} w_2 = 0.$$

The relevant boundary conditions in equations (3.22) and (3.23), now take the form

$$w_1(z = 0) = 1, \tag{3.24}$$

and

$$w_1(z \to \infty) = 0. \tag{3.25}$$

The graph of the numerical solution of the reduced ODE is shown in Figure 3.1.

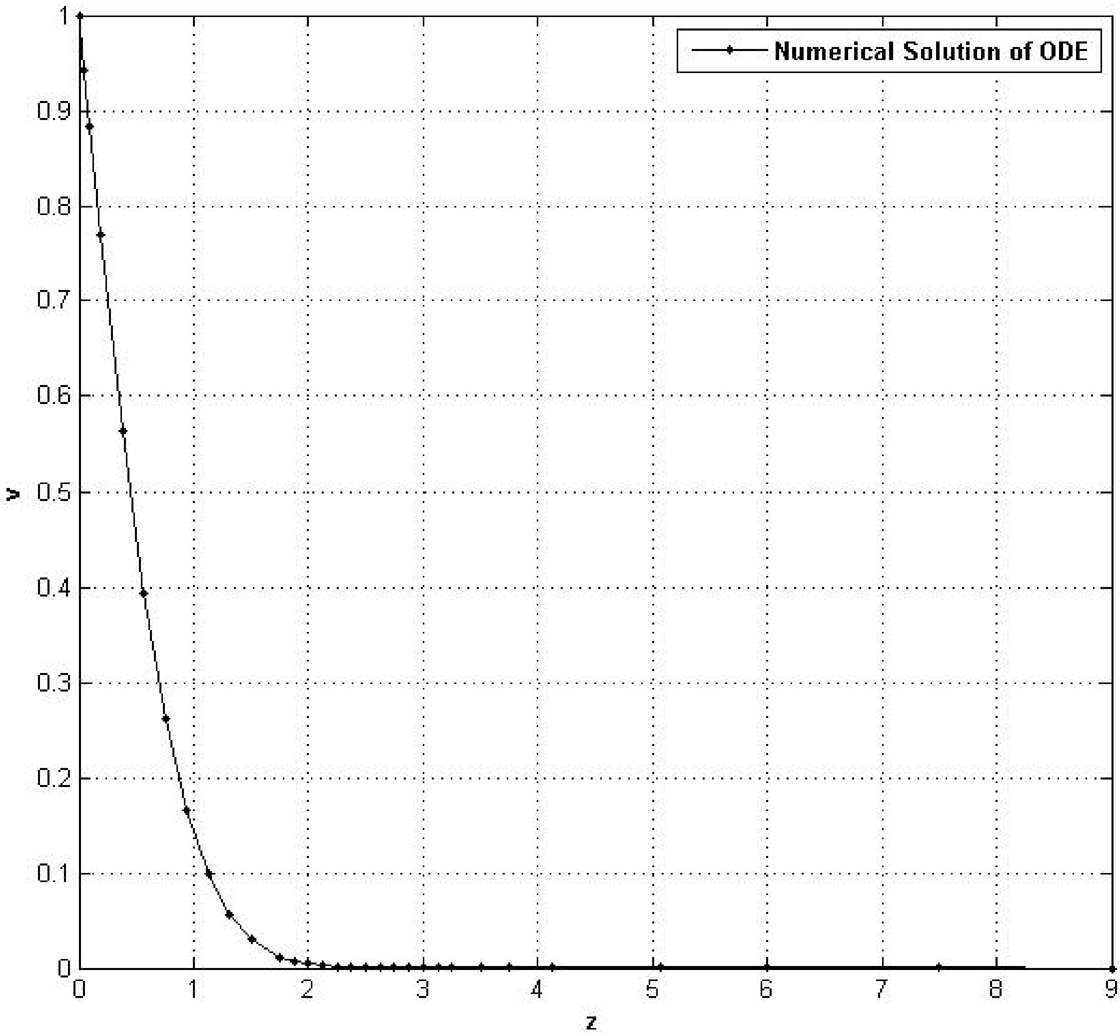

Figure 3.1: Numerical solution of the ODE.

3.1.3 Obtaining Initial Approximation for the Approximate Solution

To obtain an initial guess, v_{approx}, the lower solution [12] of equation (3.21) is used as an initial guess. Lower solution leads to an accurate approximate solution in few iterations, because it provides a good initial guess. For approximate solution of equation (3.21), the initial guess for all the cases is considered as

$$\text{initial approximation} = v_{lower} = 1 - \text{erf}\Big(\frac{z}{\sqrt{u_1}}\Big), \tag{3.26}$$

where erf denotes the error function.

3.1.4 Refining the Initial Approximation

To get v_{approx}, we improve the initial approximation to the desired level of accuracy. In order to improve the initial approximation following strategy is adopted. Assume the lower solution as

$$1 - \text{erf}(hz), \tag{3.27}$$

with $h = h_0 = 1/\sqrt{u_1}$ gives an initial approximation v_{h_0} that is, the lower solution. Numerical simulations suggest that as the value of h decrease from h_0 by a small decrement, the lower solution moves uniformly towards the numerical solution. Given a function *N(x)* and a number $\varepsilon > 0$, we say that *f(x)* lies within ε-band of *N(x)* on an interval I if

$$\mid f(x) - N(x) \mid < \varepsilon, \quad \forall x \in I. \tag{3.28}$$

For a suitable value n and numbers $\epsilon > 0$, $\delta_i > 0$, using the sequences of values

$$h = h_i = h_0 - \delta_i, \ \ (i = 1, 2, ..., n). \tag{3.29}$$

The equation (3.27) generates a sequence of curves v_{h_i} that uniformly approaches the numerical solution, finally resulting in the curve

$$v_{approx} = v_{h_n},$$

which lies in an ε-band around the graph of numerical solution v_{num}. The number ε is chosen according to the desired level of accuracy and the value of h_n is approximated by numerical simulations.

Using the transformations given in equations (3.9) and (3.10), one gets the approximate solution, $u(x,t)$, of the IBVP in equation (3.2). In the following subsection we describe the implementations of the above procedure and calculate approximate solutions for some cases of the values of u_1.

3.1.5 Approximate Closed-Form Solution for $u_1 = 0.9$

To find approximate solution of ODE in equation (3.21) we first solve the ODE numerically, with the help of **MATLAB** function *bvp4c*. We call numerical solution as v_{num}. For $u_1 = 0.9$, $h_0 = 1/\sqrt{0.9}$, and the lower solution of the reduced ODE becomes

$$v_{lower} = 1 - \text{erf}\Big(\frac{z}{\sqrt{0.9}}\Big),$$

or

$$v_{lower} \approx 1 - \text{erf}(1.05409z). \tag{3.30}$$

The graph of v_{num} and v_{lower} is given in Figure 3.2.

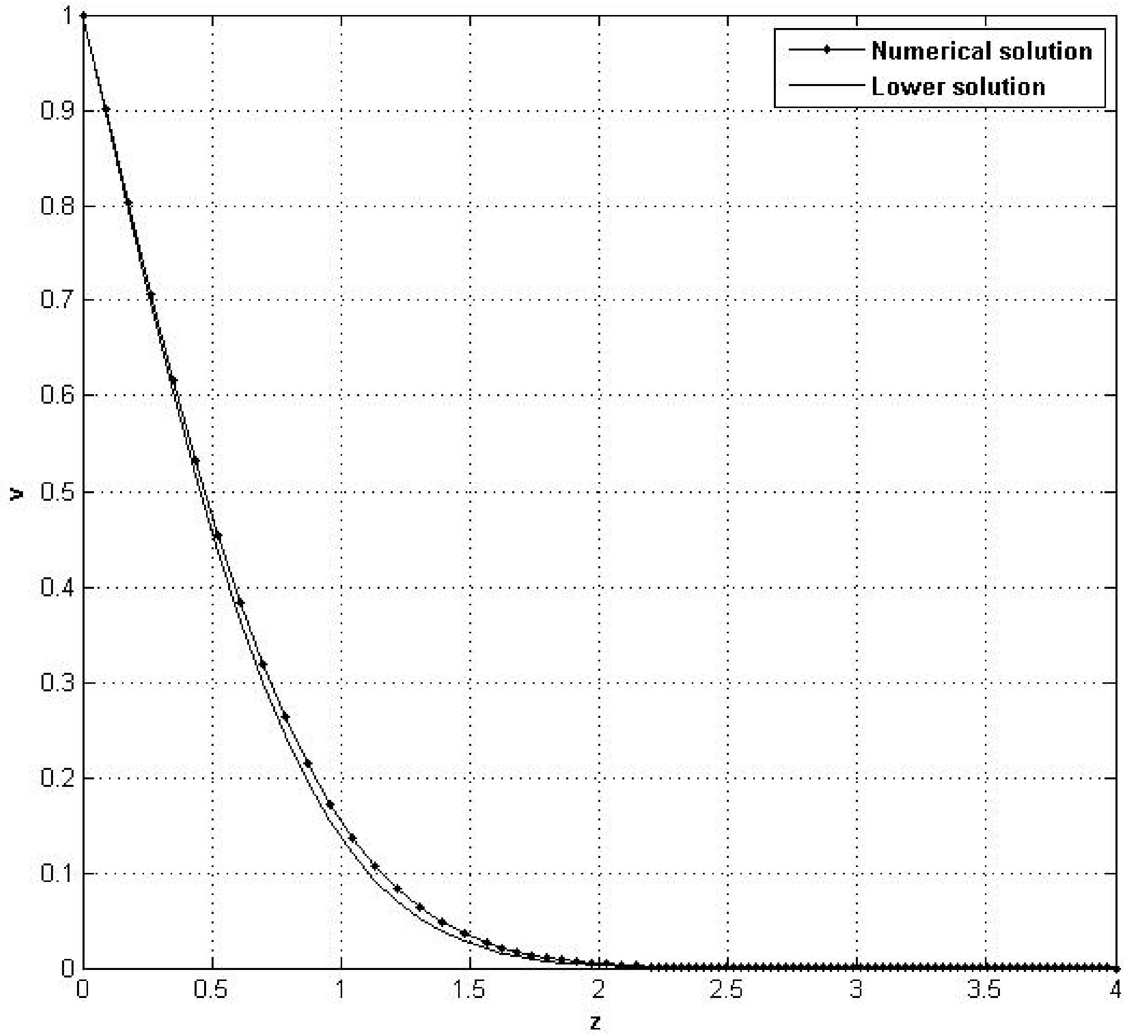

Figure 3.2: Comparisons of v_{num} and v_{lower} for $u_1 = 0.9$.

The maximum absolute difference between numerical solution, v_{num}, and lower solution, v_{lower}, is

$$\text{Max} \mid v_{num} - v_{lower} \mid = 0.0195. \tag{3.31}$$

To plot the difference at each point of the solutions, we define

$$Error(z) = v_{num}(z) - v_{lower}(z), \tag{3.32}$$

whose plot is given in Figure 3.3.

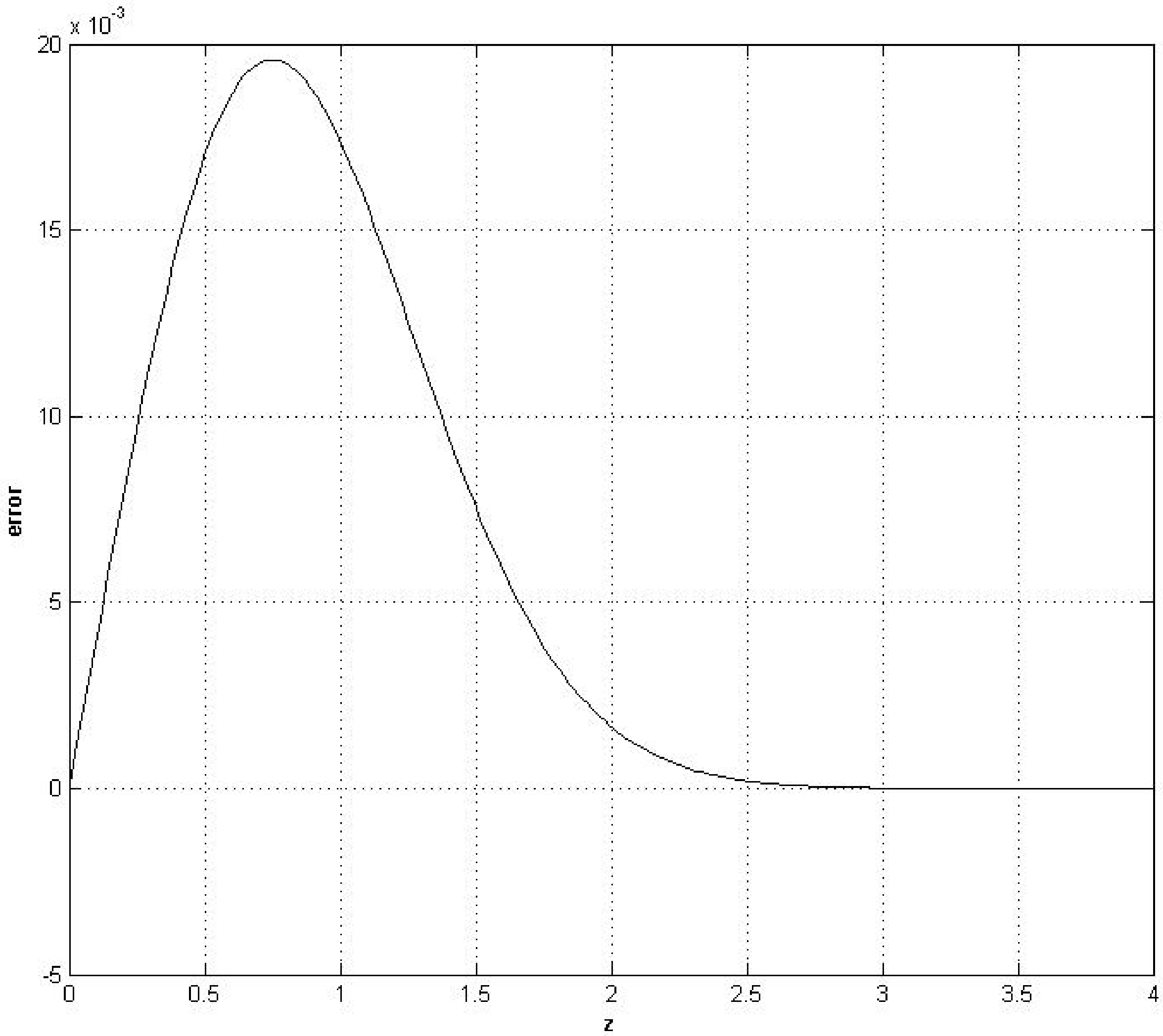

Figure 3.3: Difference between v_{num} and v_{lower}.

The procedure explained in subsection (3.1.4) with decreasing value of $h_0 = 1.05409$ to h_n= 1.0111, we get v_{approx} as

$$v_{approx} = 1 - \text{erf}(1.0111z). \tag{3.33}$$

From Figure 3.4 it is seen that the graph of v_{approx} is more closer to v_{num} than v_{lower}. Here we have

$$\text{Max} \mid v_{num} - v_{approx} \mid = 0.0011. \quad (3.34)$$

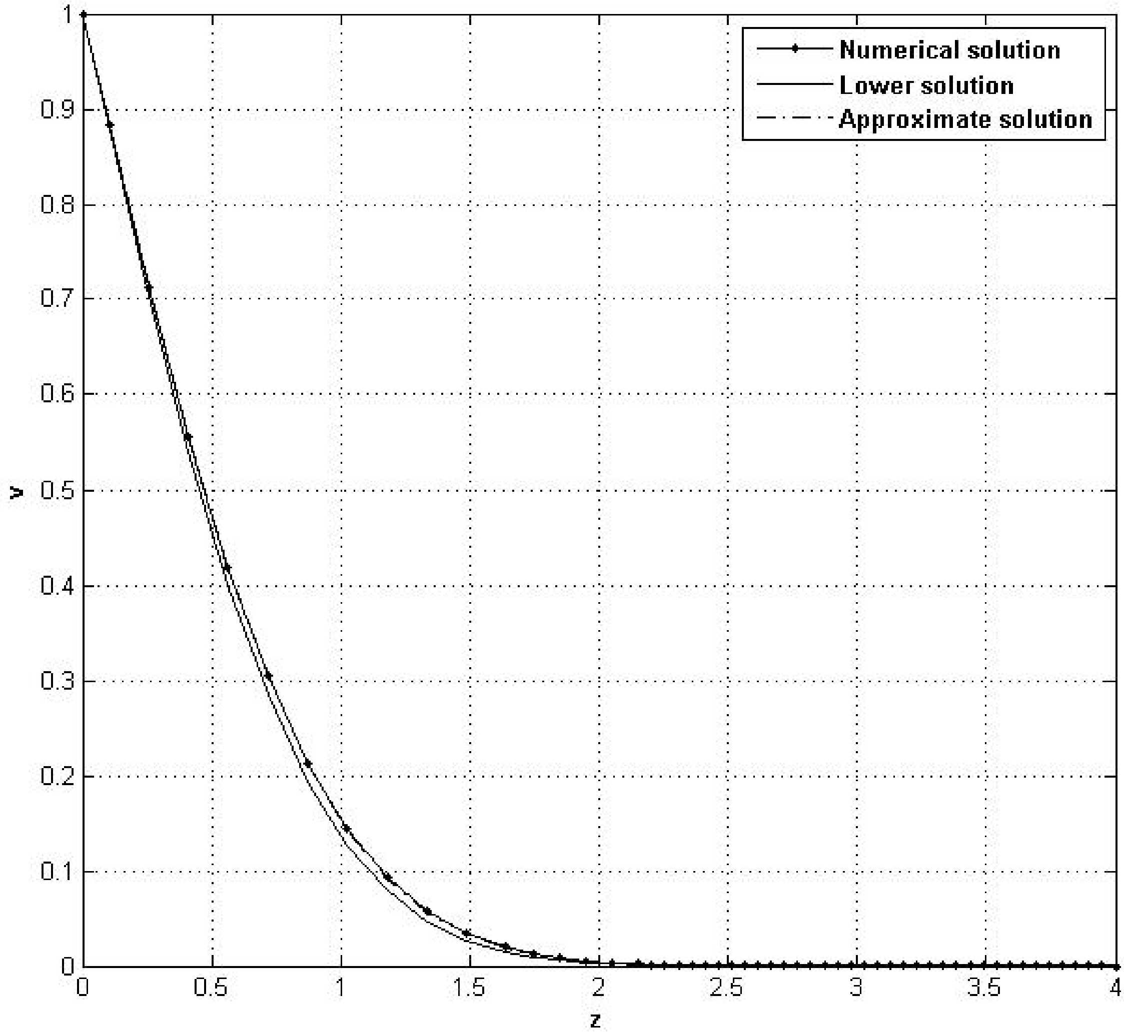

Figure 3.4: Comparisons of v_{lower}, v_{num}, and v_{approx}.

To compare pointwise error between v_{num} and v_{approx}, we define

$$Error(z) = v_{num}(z) - v_{approx}(z).$$

The graph of error between v_{num}, v_{lower} and v_{num}, v_{approx} is given in Figure 3.5.

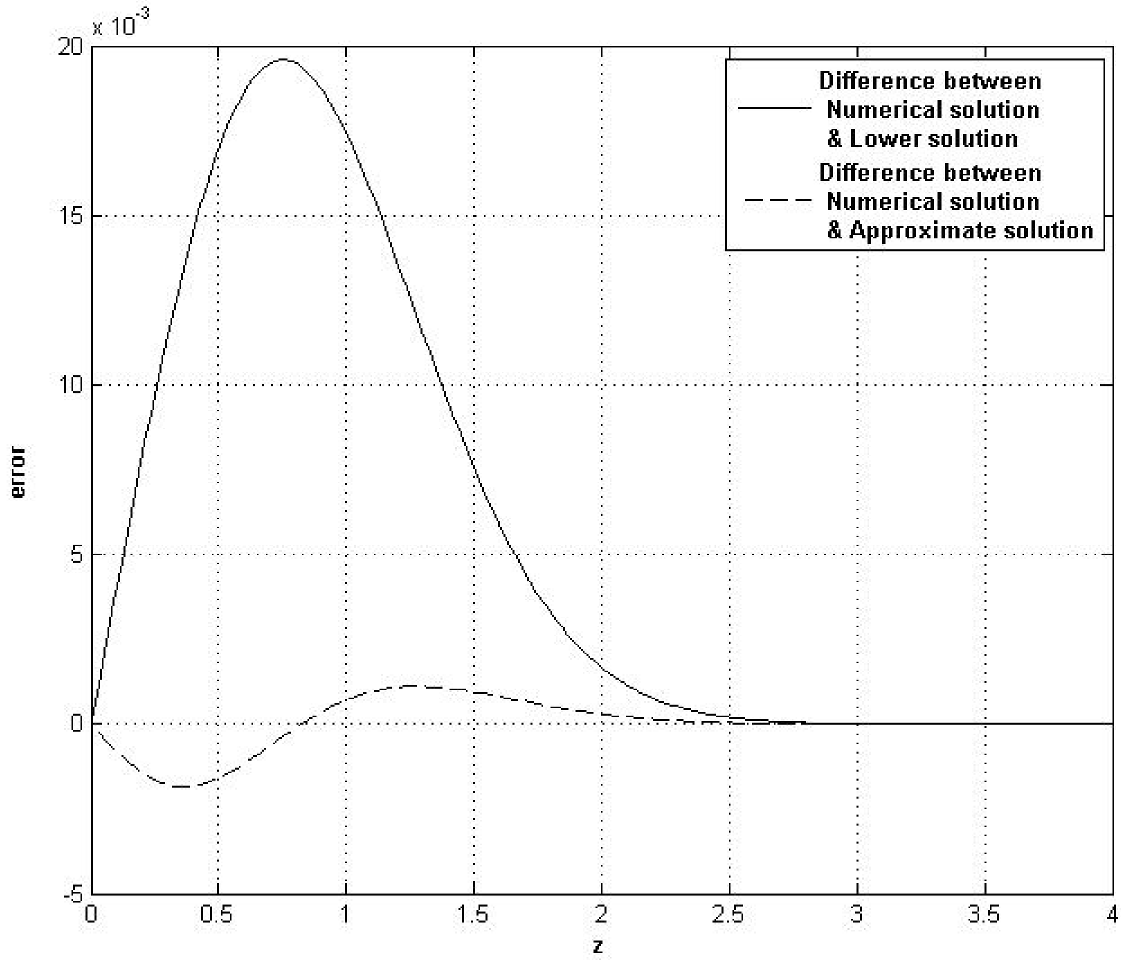

Figure 3.5: Errors in initial and in final approximation.

3.1.6 Approximate Closed-Form Solution for $u_1 = 0.3$

For $u_1 = 0.3$, $h_0 = 1/\sqrt{0.3}$, and the lower solution of the reduced ODE becomes

$$v_{lower} \approx 1 - \text{erf}(1.82574z). \tag{3.35}$$

The graph of v_{num} and v_{lower} is shown in Figure 3.6.

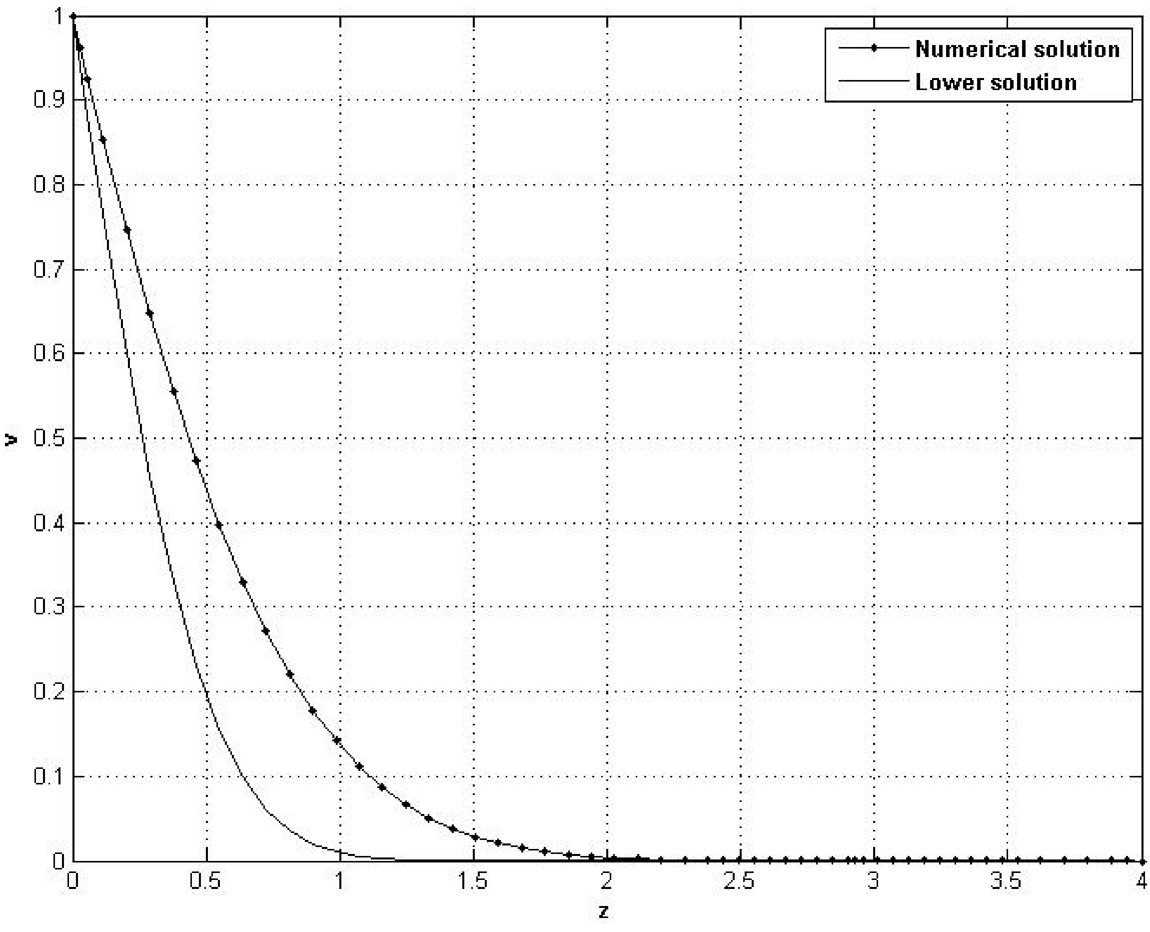

Figure 3.6: Comparsions of the v_{num} and v_{lower} for $u_1 = 0.3$.

The maximum absolute difference between numerical solution, v_{num}, and lower solution, v_{lower}, is

$$\text{Max} \mid v_{num} - v_{lower} \mid = 0.2435. \qquad (3.36)$$

Plot of the difference between v_{num} and v_{lower} is given in Figure 3.7.

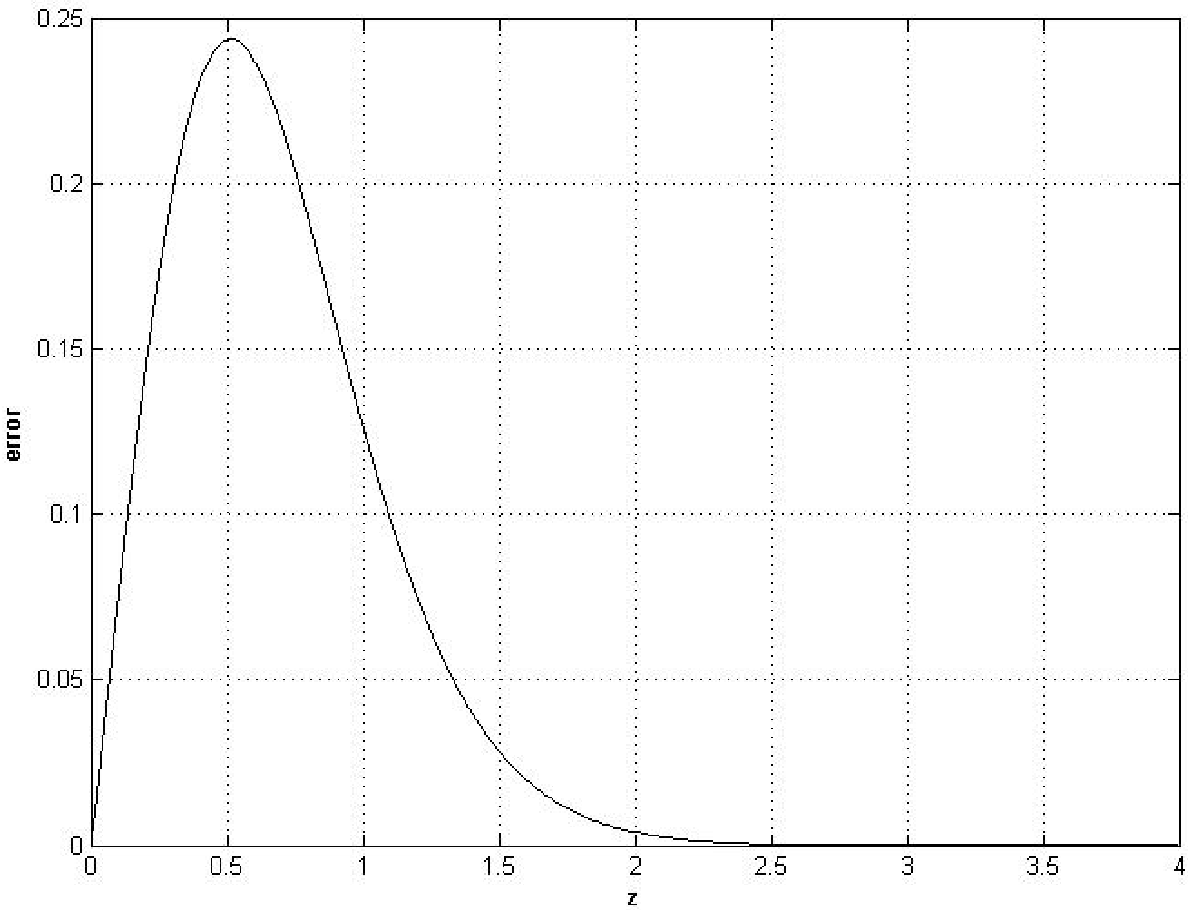

Figure 3.7: Difference between v_{num} and v_{lower}.

By the procedure explained in subsection (3.1.4), decreasing value of $h_0 = 1.82574$ to h_n= 1.0657, we get v_{approx} as

$$v_{approx} = 1 - \text{erf}(1.0657z). \tag{3.37}$$

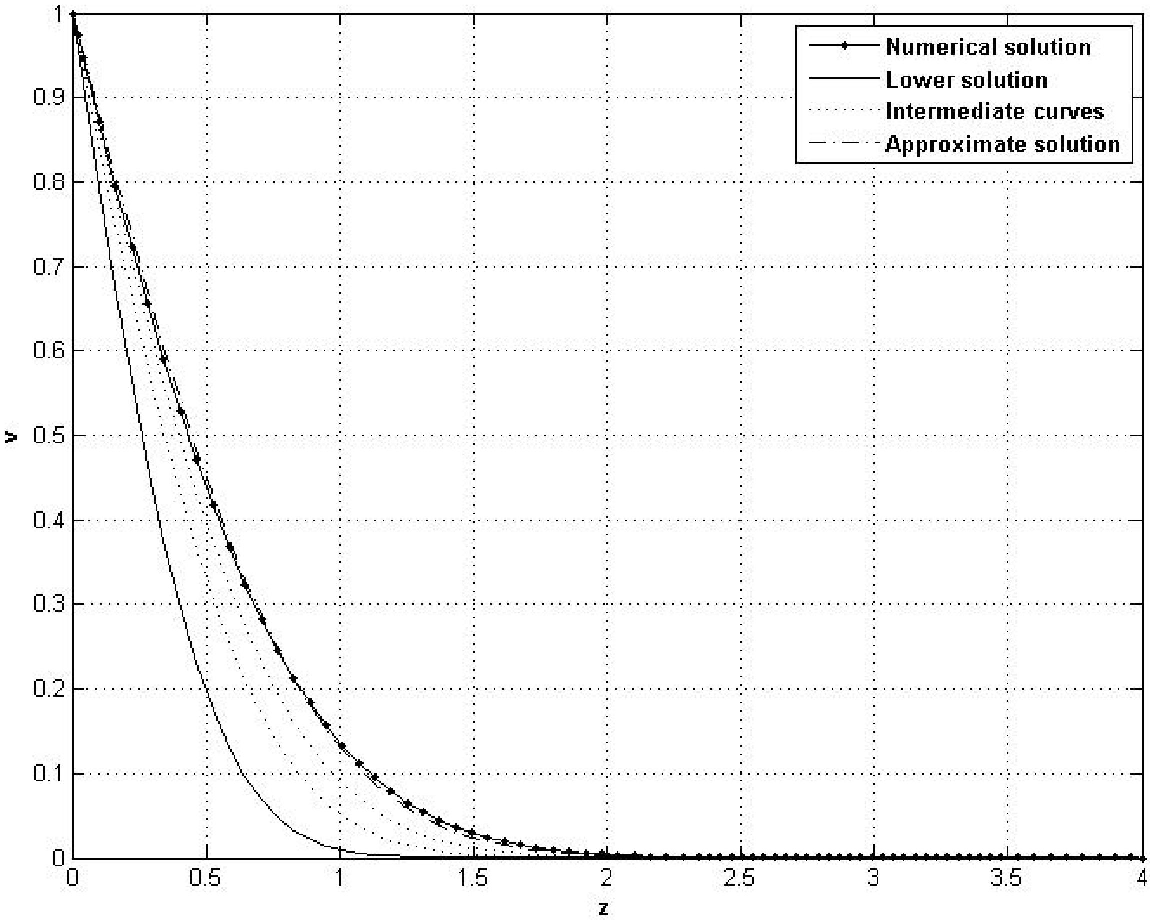

Figure 3.8: The solutions of v_{lower}, v_{num}, v_{approx} and intermediate curves.

From Figure 3.8 it is seen that the graph of v_{approx} is more closer to v_{num} than v_{lower}. The dotted curve is taken from a sequence of curves v_{h_i} approaching uniformly towards the numerical solution. Here we have

$$\text{Max} \mid v_{num} - v_{approx} \mid = 0.0066. \tag{3.38}$$

Plot of the difference between v_{num} and v_{approx} is given in Figure 3.9.

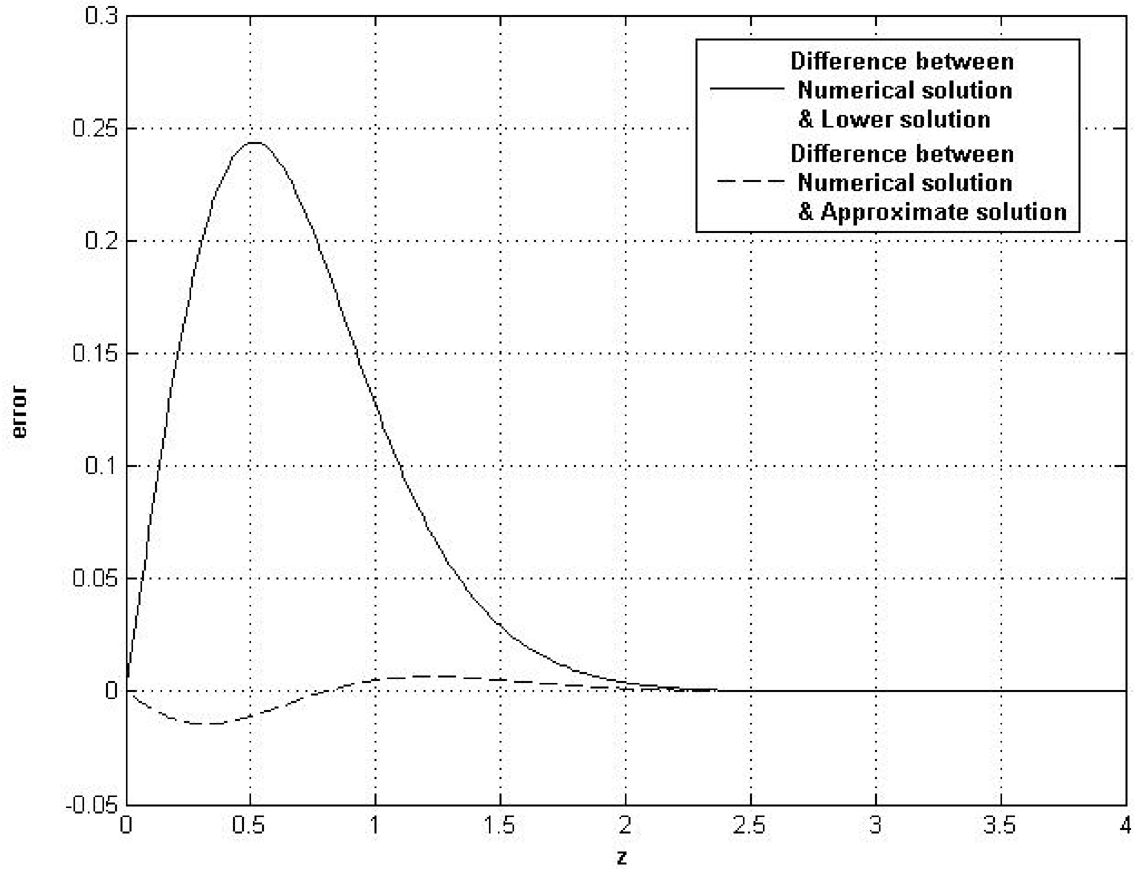

Figure 3.9: Errors in initial and in final approximation.

3.1.7 Approximate Closed-Form Solution for $u_1 = 0.1$

For $u_1 = 0.1$, $h_0 = 1/\sqrt{0.1}$, and the lower solution of the reduced ODE becomes

$$v_{lower} \approx 1 - \text{erf}(3.162277z). \tag{3.39}$$

The graph of v_{num} and v_{lower} is given in Figure 3.10.

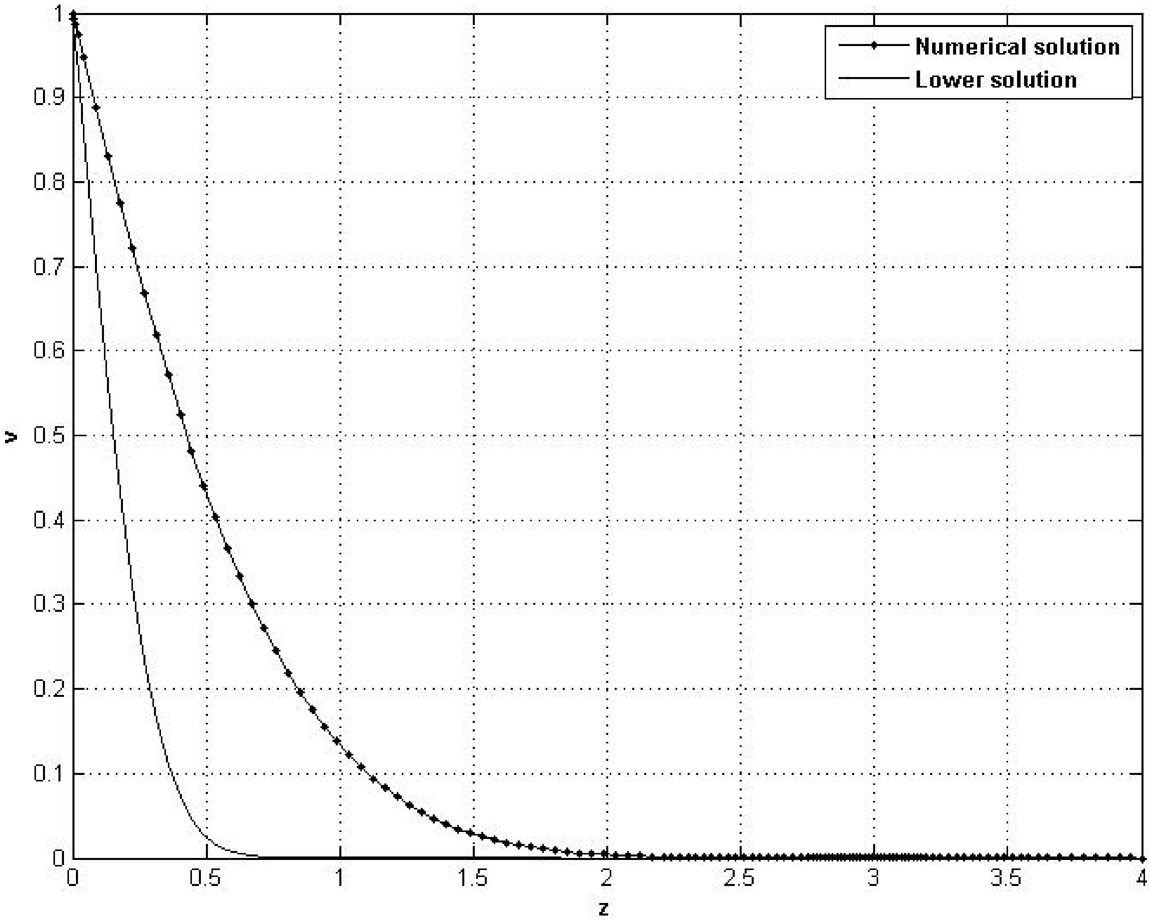

Figure 3.10: Plots of the v_{num} and v_{lower} for $u_1 = 0.1$.

The maximum absolute difference between numerical solution, v_{num}, and lower solution, v_{lower}, is

$$\text{Max} \mid v_{num} - v_{lower} \mid = 0.4598. \tag{3.40}$$

In Figure 3.11 we plot the difference between v_{num} and v_{lower}.

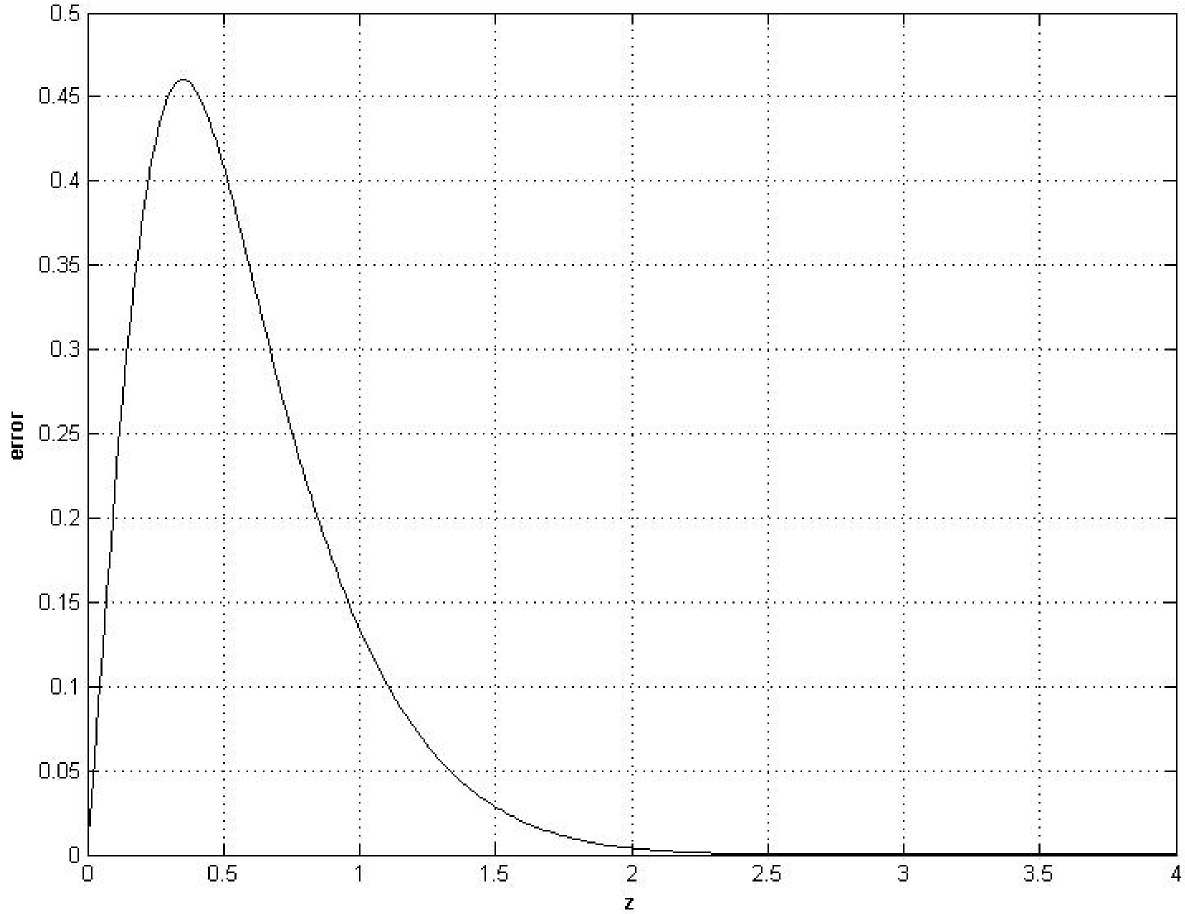

Figure 3.11: Difference between v_{num} and v_{lower}.

By the procedure explained in subsection (3.1.4), decreasing value of $h_0 = 3.162277$ to h_n= 1.0823, we get v_{approx} as

$$v_{approx} = 1 - \text{erf}(1.0823z). \tag{3.41}$$

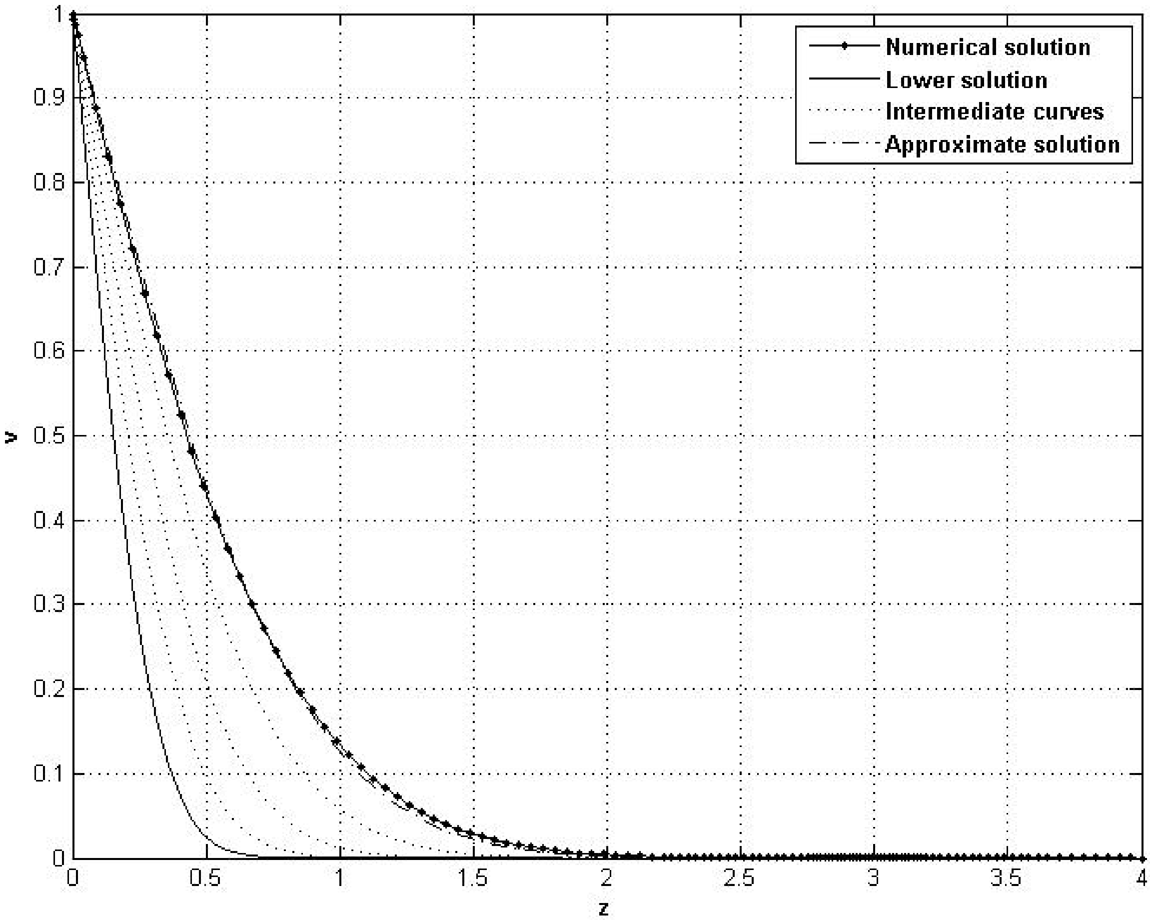

Figure 3.12: Solutions of v_{lower}, v_{num}, v_{approx}, intermediate curves.

From Figure 3.12 it is seen that the graph of v_{approx} is more closer to v_{num} than v_{lower}. The intermediate curves are ploted by taking values between h_0 and h_n. Here we have

$$\text{Max} \mid v_{num} - v_{approx} \mid = 0.0096. \tag{3.42}$$

In Figure 3.13 we plot the difference between v_{num} and v_{lower}.

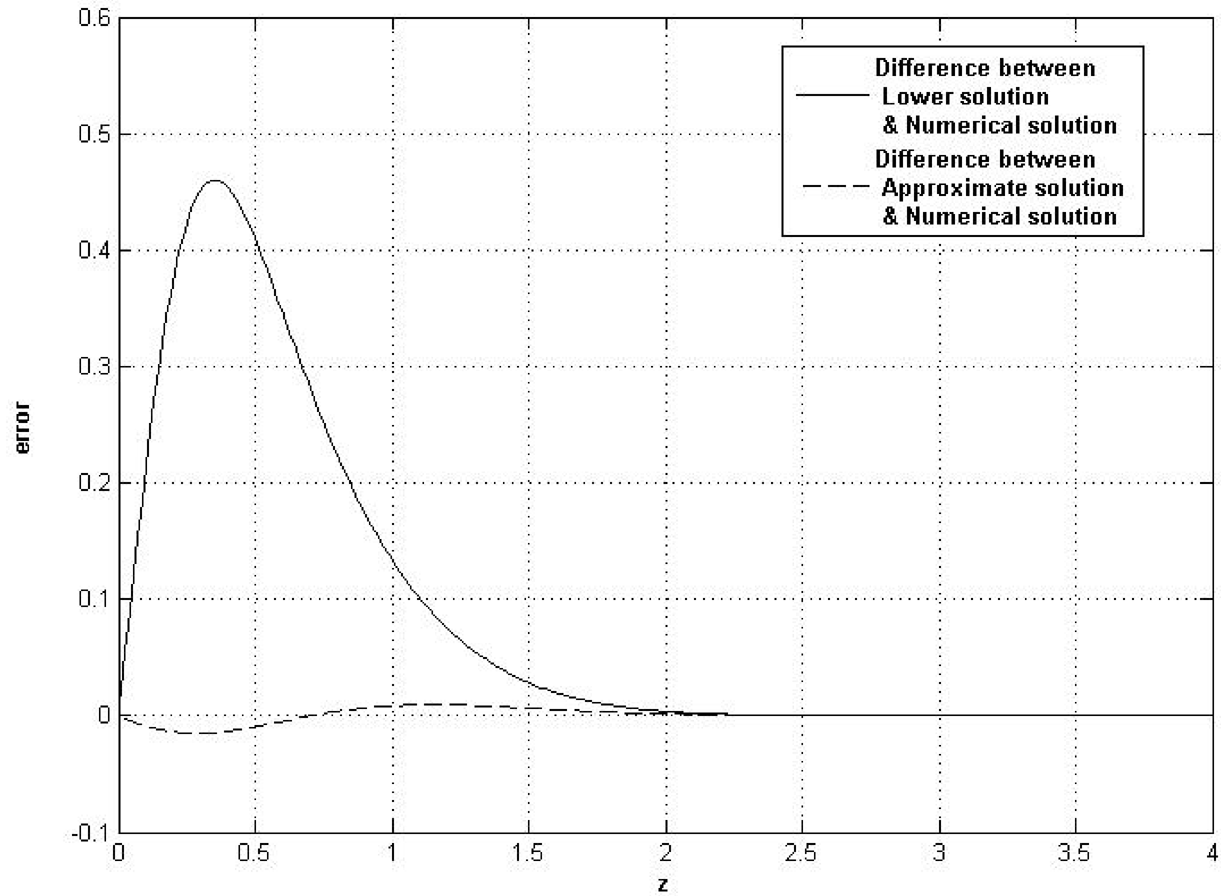

Figure 3.13: Errors in initial and in final approximation.

Chapter 4

Approximate Closed-Form Solution of Diffusion Equation

In chapter 3, the diffusion equation is reduced to an ODE by applying symmetry transformations. The solution of the reduced ODE is found numerically by using **MATLAB**. Then using the numerical solution, an approximate closed-form solution of the reduced ODE is obtained. In this chapter we obtain an approximate closed-form solution of the diffusion equation by applying the inverse symmetry transformations to the approximate solution of the reduced ODE. After that numerical solution of the diffusion equation is obtained using **MATLAB**. At the end a comparison between numerical and approximate closed-form solutions of the diffusion equation is presented.

4.1 Approximate Solution of Diffusion Equation for $u_1 = 0.9$

Using the approximate solution given by equation (3.33) and the transformations given by equations (3.9) and (3.10), we get the approximate closed-form solution of the IBVP (3.2) - (3.5), given by

$$u(x,t) = \sqrt{0.81 + 0.19\text{erf}\Big(1.0111\frac{x}{\sqrt{t}}\Big(\frac{c}{4}\Big)^{1/2}\Big)}. \tag{4.1}$$

Note that $u(x,t)$ is undetermined at $(x,t) = (0,0)$. However, taking the limit as $(x,t) \to (0,0)$, we have

$$\lim_{(x,t)\to(0,0)} u(x,t) = \lim_{(x,t)\to(0,0)} \sqrt{0.81 + 0.19\text{erf}\Big(1.0111\frac{x}{\sqrt{t}}\Big(\frac{c}{4}\Big)^{1/2}\Big)} = 0.9, \tag{4.2}$$

which agrees with the boundary condition. The approximate solution $u(x,t)$ given by equation (4.1) is shown in Figure 4.1.

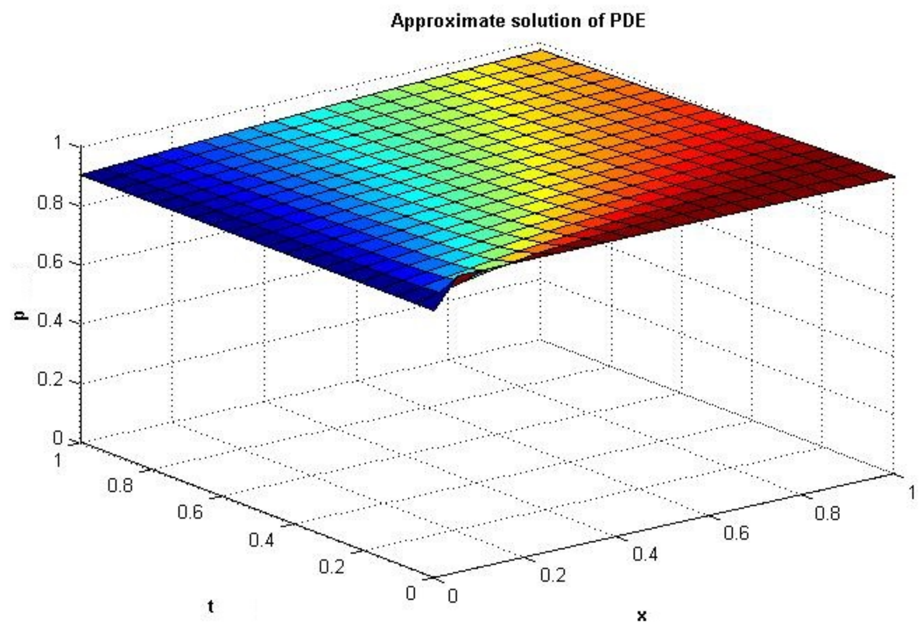

Figure 4.1: Approximate solution of the diffusion equation for $u_1 = 0.9$.

4.1.1 Numerical solution of Diffusion Equation for $u_1 = 0.9$

In this subsection, we solve the diffusion equation by applying **MATLAB** function PDE Toolbox described in section (2.2). The 3D plot of numerical solution of diffusion equation is presented in Figure 4.2.

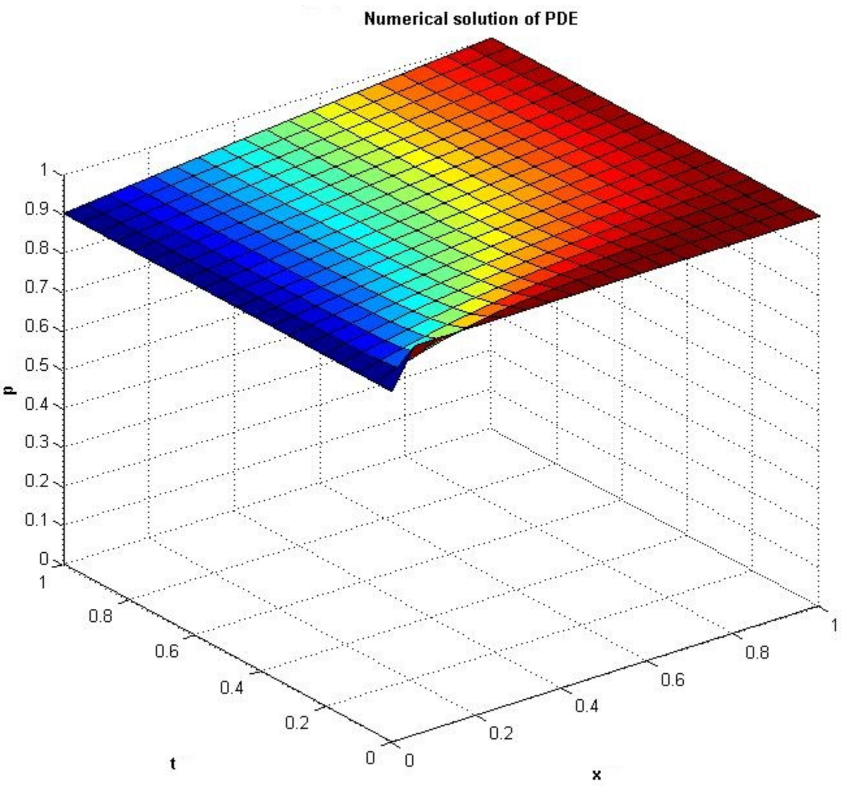

Figure 4.2: Numerical solution of the diffusion equation for $u_1 = 0.9$.

4.1.2 Comparison between Numerical and Approximate Solution for $u_1 = 0.9$

In this subsection, we compare both the numerical and the approximate solutions of the diffusion equation. The Figure 4.3 shows the difference between the numerical and approximate solutions. The maximum absolute difference between the numerical solution and the closed-form solution given by equation (4.1) is 0.03.

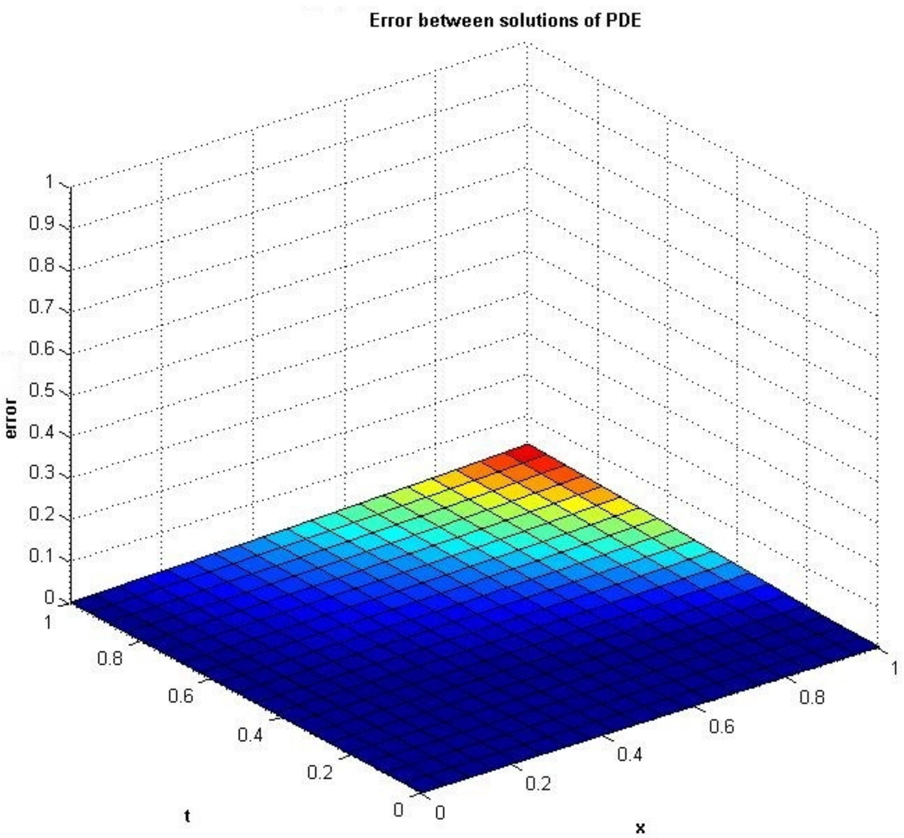

Figure 4.3: The error of the diffusion equation for $u_1 = 0.9$.

4.2 Approximate Solution of Diffusion Equation for $u_1 = 0.3$

Using the approximate solution given by equation (3.37) and the transformations given by equations (3.9) and (3.10), we get the approximate closed-form solution of the IBVP (3.2) - (3.5), given by

$$u(x,t) = \sqrt{0.09 + 0.91\text{erf}\Big(1.0657\frac{x}{\sqrt{t}}\Big(\frac{c}{4}\Big)^{1/2}\Big)}. \tag{4.3}$$

Note that $u(x,t)$ is undetermined at $(x,t) = (0,0)$. However, taking the limit as $(x,t) \to (0,0)$, we have

$$\lim_{(x,t)\to(0,0)} u(x,t) = \lim_{(x,t)\to(0,0)} \sqrt{0.09 + 0.91\text{erf}\Big(1.0657\frac{x}{\sqrt{t}}\Big(\frac{c}{4}\Big)^{1/2}\Big)} = 0.3, \tag{4.4}$$

which agrees with the boundary condition. The approximate solution $u(x,t)$ given by equation (4.3) is shown in Figure 4.4.

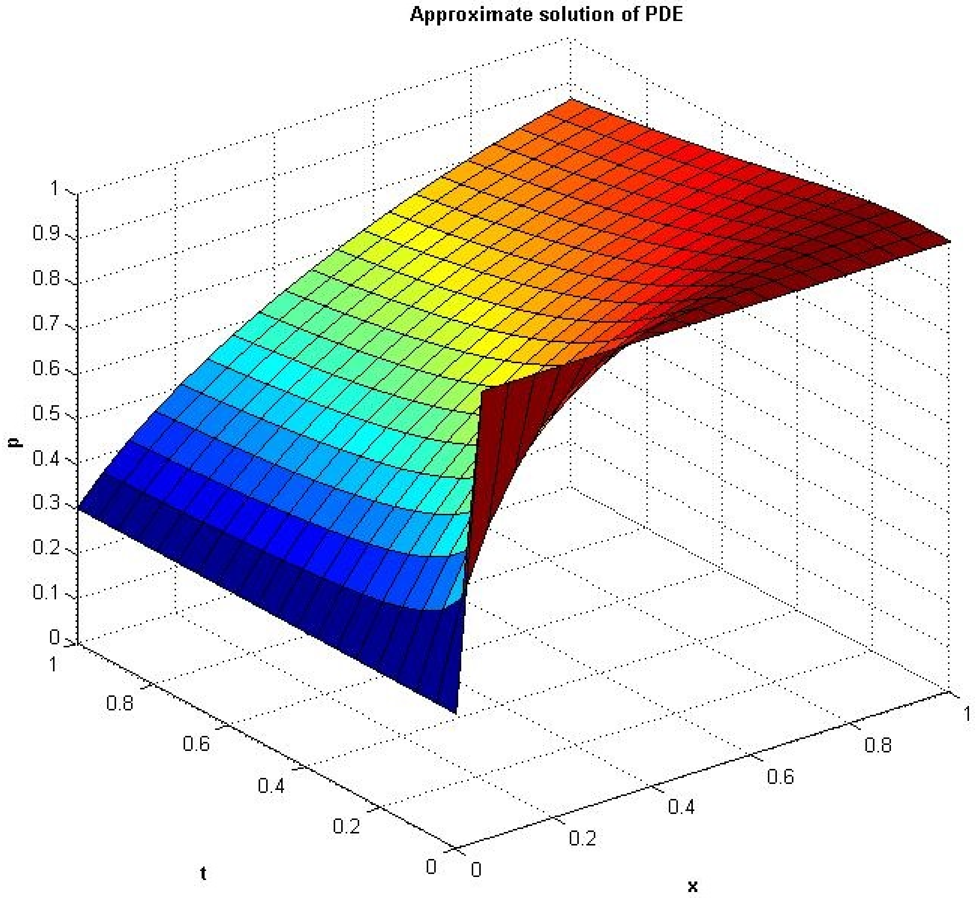

Figure 4.4: Approximate solution of diffusion equation for $u_1 = 0.3$.

4.2.1 Numerical Solution of Diffusion Equation for $u_1 = 0.3$

In this subsection, we solve the diffusion equation by applying **MATLAB** function PDE Toolbox decribed in section (2.2). The 3D plot of numerical solution of diffusion equation is presented in Figure 4.5.

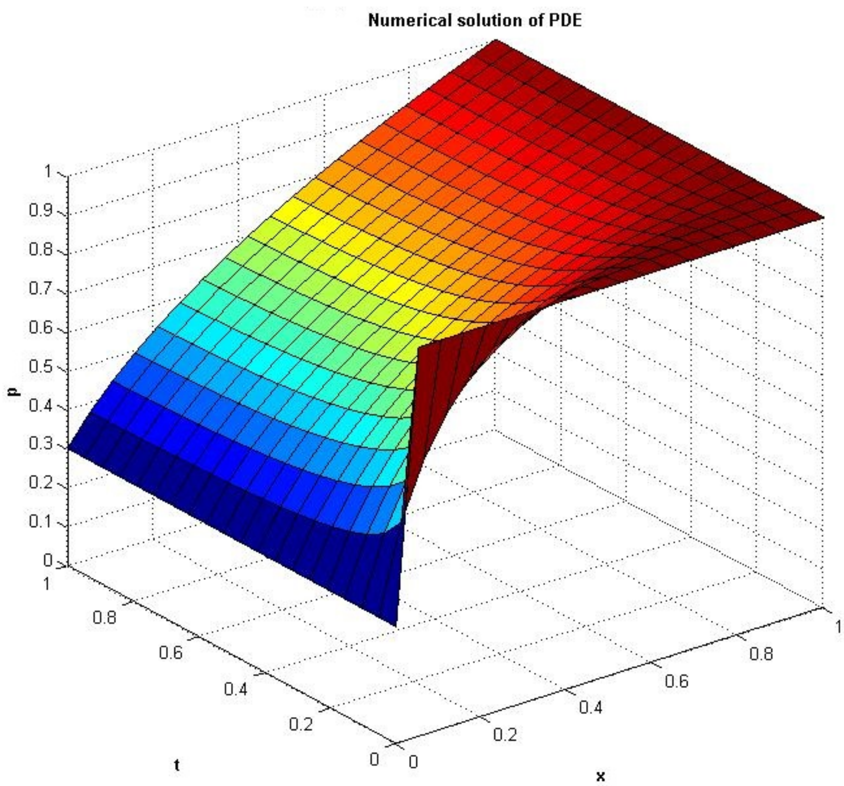

Figure 4.5: Numerical solution of the diffusion equation for $u_1 = 0.3$.

4.2.2 Comparison between Numerical and Approximate Solution for $u_1 = 0.3$

In this subsection, we compare both the numerical and the approximate solutions of the diffusion equation. The Figure 4.6 shows the difference between the numerical and approximate solutions.

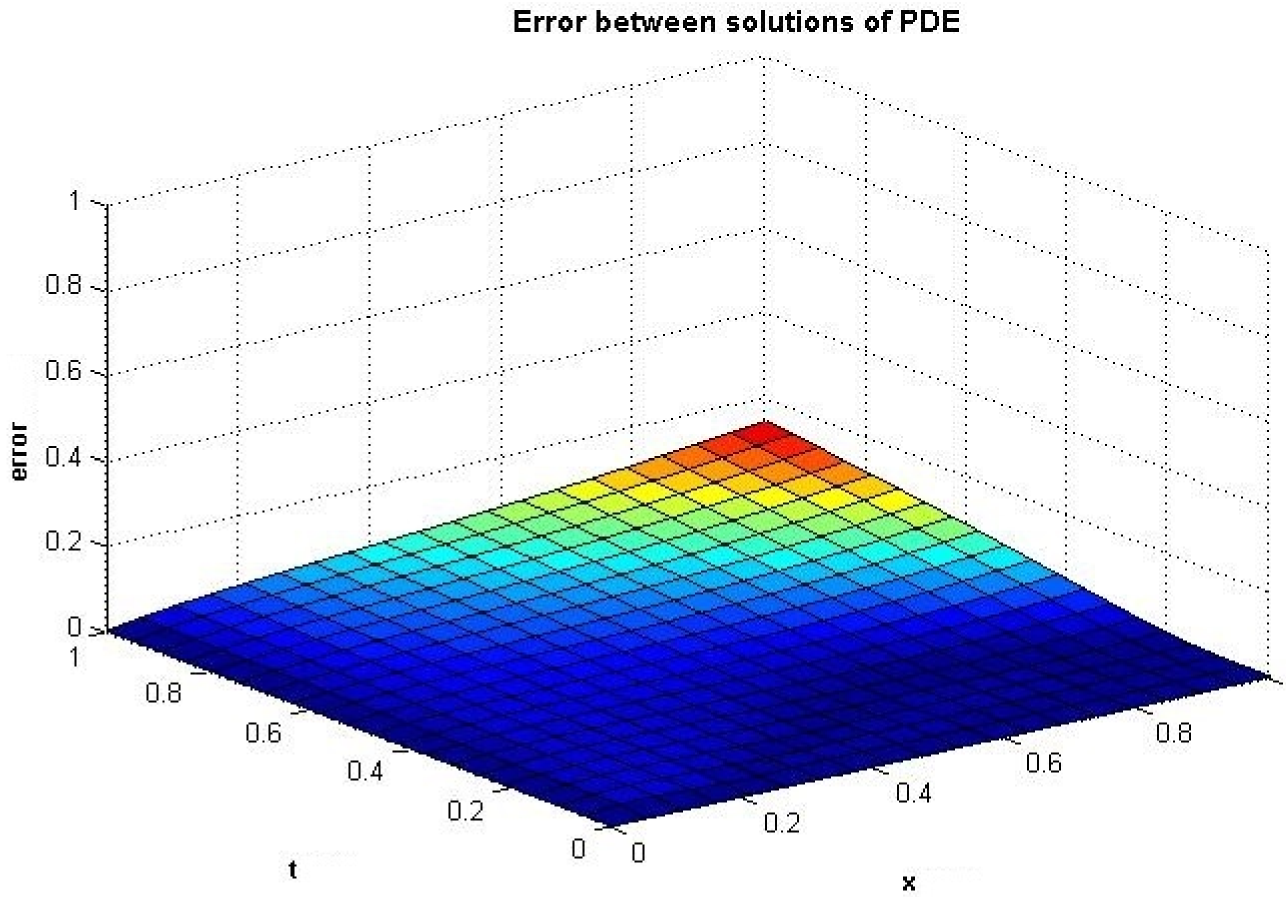

Figure 4.6: The error of the diffusion equation for $u_1 = 0.3$.

The maximum absolute difference between the numerical solution and the closed-form solution given by equation (4.3) is 0.14.

4.3 Approximate Solution of Diffusion Equation for $u_1 = 0.1$

Using the approximate solution given by equation (3.41) and the transformations given by equations (3.9) and (3.10), we get the approximate closed-form solution of the IBVP (3.2) - (3.5), given by

$$u(x,t) = \sqrt{0.01 + 0.99\text{erf}\Big(1.0823\frac{x}{\sqrt{t}}\Big(\frac{c}{4}\Big)^{1/2}\Big)}. \tag{4.5}$$

Note that $u(x,t)$ is undetermined at $(x,t) = (0,0)$. However, taking the limit as $(x,t) \to (0,0)$, we have

$$\lim_{(x,t)\to(0,0)} u(x,t) = \lim_{(x,t)\to(0,0)} \sqrt{0.01 + 0.99\text{erf}\Big(1.0823\frac{x}{\sqrt{t}}\Big(\frac{c}{4}\Big)^{1/2}\Big)} = 0.1. \tag{4.6}$$

which agrees with the boundary condition. The approximate solution $u(x,t)$ given by equation (4.5) is shown in Figure 4.7.

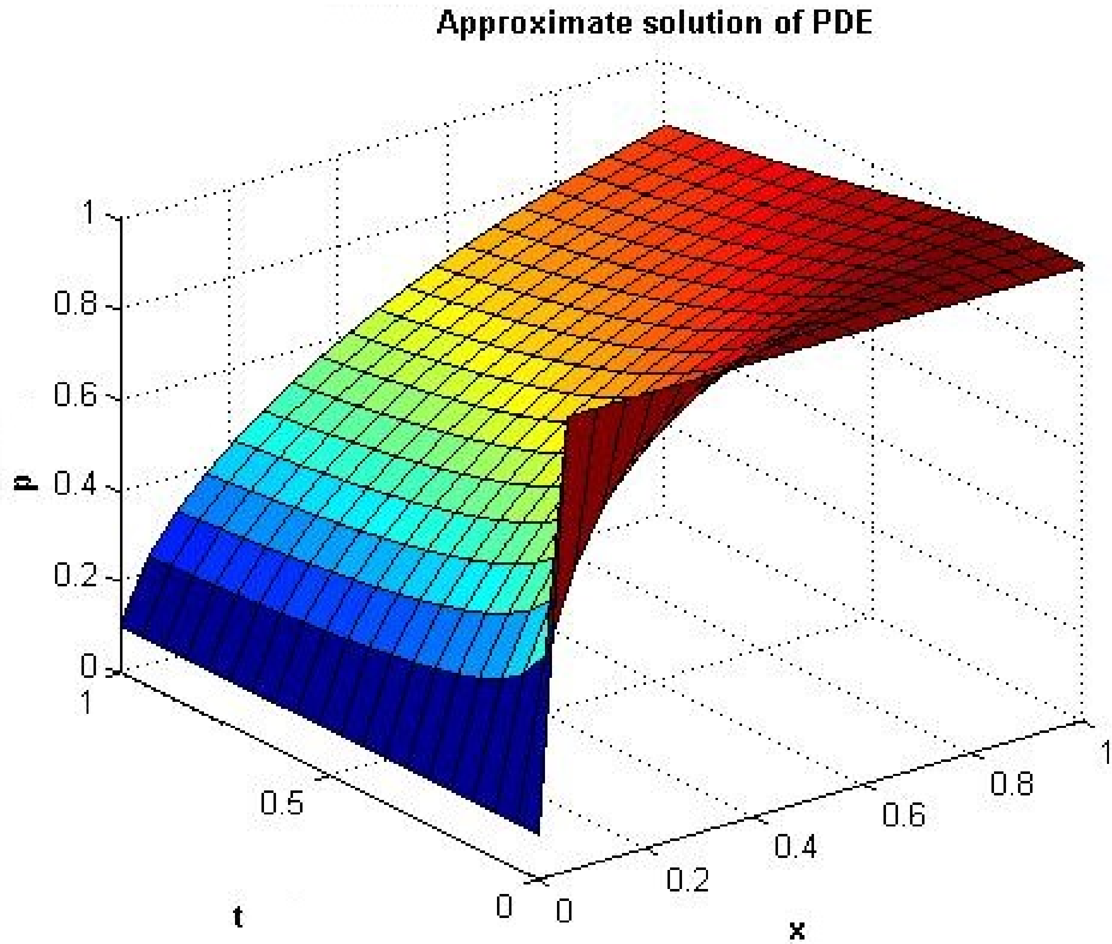

Figure 4.7: Approximate solution of the diffusion equation for $u_1 = 0.1$

4.3.1 Numerical Solution of the Diffusion Equation for $u_1 = 0.1$

In this subsection, we solve the diffusion equation by applying **MATLAB** function PDE Toolbox described in section (2.2). The 3D plot of numerical solution of diffusion equation is presented in Figure 4.8.

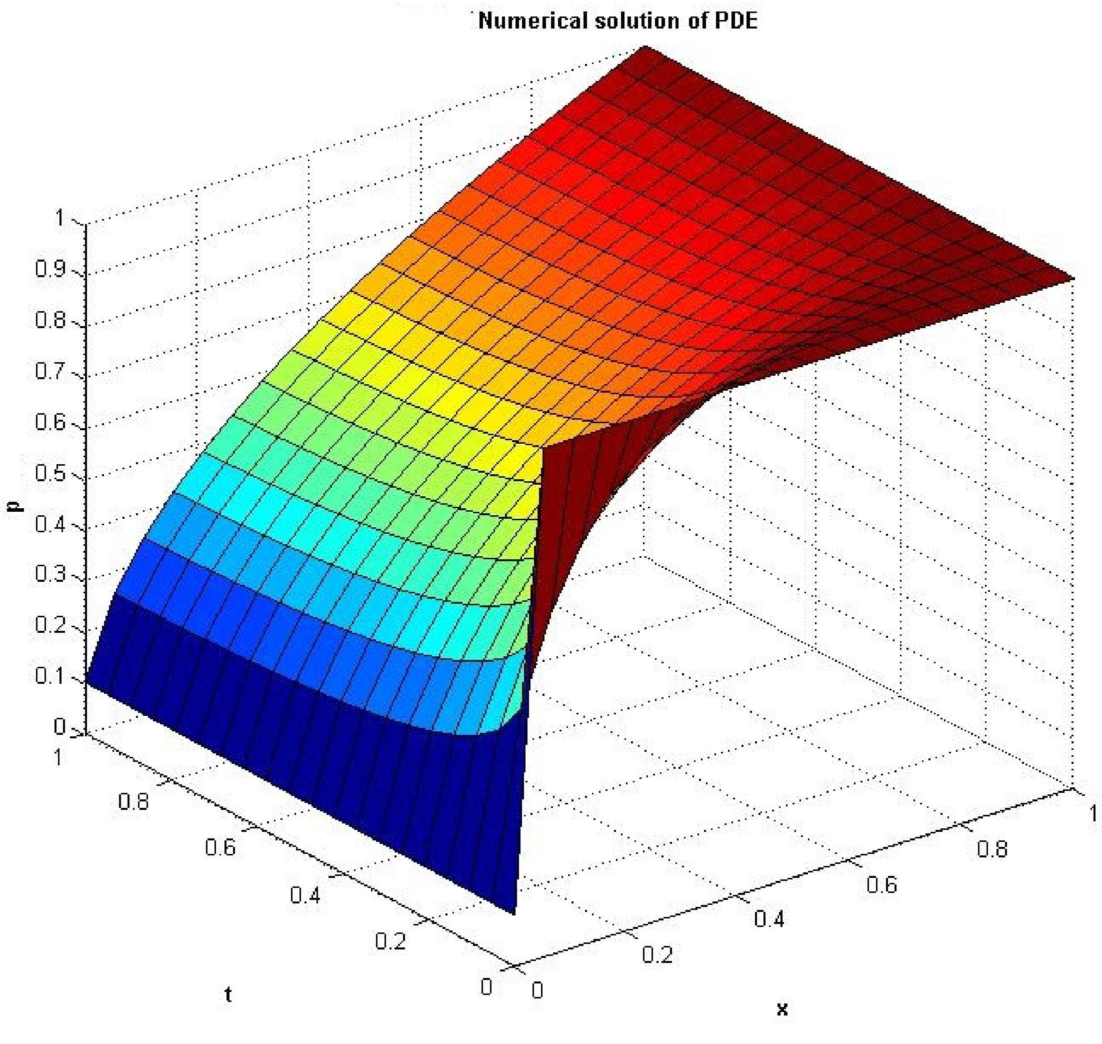

Figure 4.8: Numerical solution of the diffusion equation for $u_1 = 0.1$.

4.3.2 Comparison between Numerical and Approximate Solution for $u_1 = 0.1$

In this subsection, we compare both the numerical and the approximate solutions of diffusion equation. The Figure 4.9 shows the difference between the numerical and approximate solutions. The maximum absolute difference between the numerical solution and the closed-form solution given by equation (4.5) is 0.15.

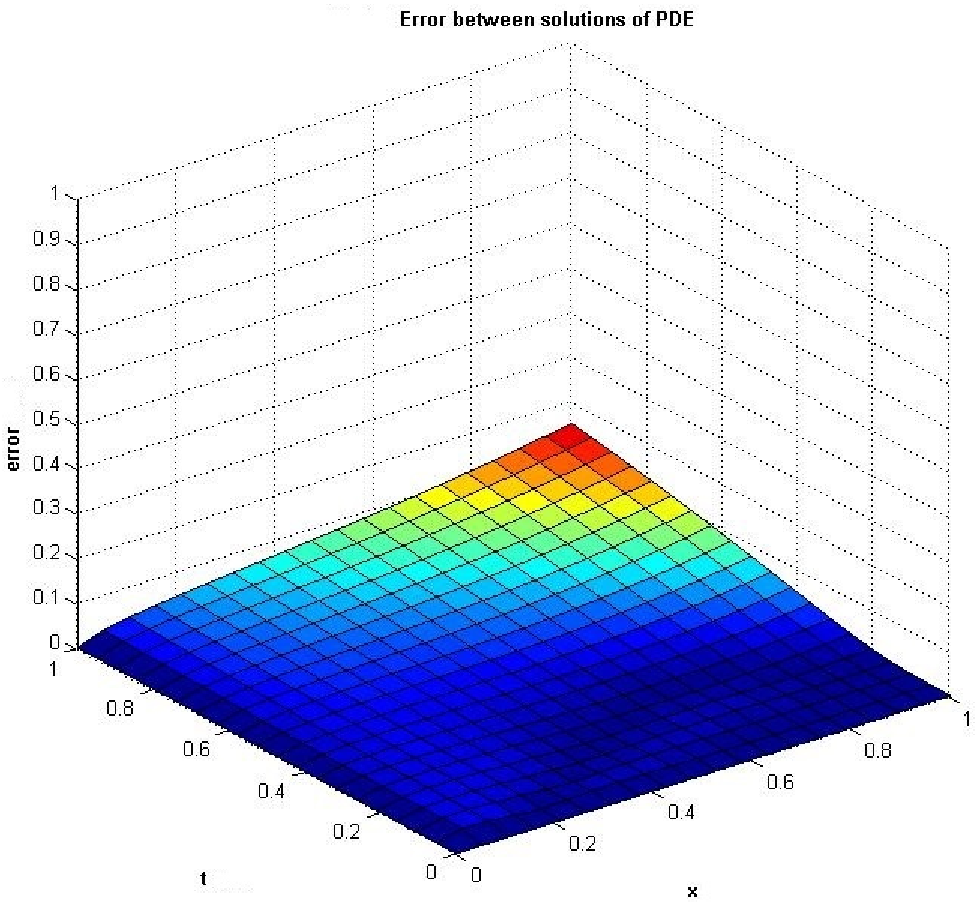

Figure 4.9: The error of the diffusion equation for $u_1 = 0.1$.

Chapter 5

Conclusions

In the application of symmetry method to solve a PDE, one often ends up in a situation, where the PDE is reduced to an ODE whose analytical solution is not possible or difficult to find, and one obtains numerical solution. A method to use the numerical solution of the reduced ODE to get an approximate closed-form solution of the original PDE was developed in reference [1]. As the transformations, used to write the approximate solution of PDE from the approximate solution of the reduced ODE are generally nonlinear, therefore, it is not clear whether the approximate solution of the PDE obtained remains reasonable in the sense that the errors involved in the approximation of the reduced ODE are not increased significantly by the use of transformations to obtain approximate solution of PDE. For this purpose, in this thesis numerical solutions of the original IBVP of PDE are obtained and the approximate solutions are compared with these numerical solutions. The comparison is made for three different values of parameter u_1 involved in the boundary conditions, which are $u_1 = 0.9$, 0.3, and 0.1. It is observed that approximate solution is reasonably closed to the numerical solution. In all cases, the maximum absolute error between the approximate and numerical solution of PDE for $u_1 = 0.9$ is 0.03, for $u_1 = 0.3$ is 0.14 and for $u_1 = 0.1$ is 0.15. Notice that the maximum error increases with the decrease in the values of u_1. This is expected as the error in the approximation of the reduced ODE also has the similar behaviour.

Bibliography

[1] M. Iqbal, M. T. Mustafa and A. A. Siddiqui, *A Method for Generating Approximate Similarity Solutions of Nonlinear Partial Differential Equations*, Abstract and Applied Analysis, Hindawi Publishing Corporation, 2014.

[2] C. C. Steven, *Applied Numerical Methods with MATLAB for Engineers and Scientists*, 2nd edition, 2008.

[3] D. Mooney and R. Swift, *A Course in Mathematical Modeling*, Washington, DC: Mathematical Association of America, 1999.

[4] D. G. Zill, D. W. Carol and S. W. Wright, *Differential Equations with Boundary Value Problems*, 4th edition, 1997.

[5] L. C. Edward, *Diffusion Mass Transfer in Fluid Systems*, Cambridge University Press, 1987.

[6] L. F. Shampine, J. kierzenka and M. W. Reichelt, *Solving Boundary Value Problems for Ordinary Differential Equations in MATLAB with bvp4c* Tutorial Notes. Available at http://www. mathworks. com, 2003.

[7] *Solving Partial Differential Equations in MATLAB with PDE Toolbox* Tutorial Notes. Available at http://www.mathworks.com/help/matlab/ref/pdepe.html.

[8] W. F. Ames, *Nonlinear Partial Differential Equations in Engineering*, vol. I, Academic Press, New York, NY, USA, 1965.

[9] L. V. Ovsiannikov, *Group Analysis of Differential Equations*, Academic Press, New York, NY, USA, 1982.

[10] G. W. Bluman, *Applications of the General Similarity Solution of the Heat Equation to Boundary Value Problems*, Quarterly of Applied Mathematics, vol. 31, pp. 403–415, 1974.

[11] R. P. Agarwal and D. O'Regan, *Infinite Interval Problems Modeling the Flow of a Gas through a Semi-Infinite Porous Medium*, Studies in Applied Mathematics, vol. 108, no. 3, pp. 245–257, 2002.

[12] R. P. Agarwal and D. O'Regan, *Non-linear Boundary Value Problems on the Semi-Infinite Interval: an Upper and Lower Solution Approach* , Mathematika, vol. 49, no. 1–2, pp. 129–140, 2002.

[13] R. E. Kidder, *Unsteady Flow of Gas through a Semi-Infinite Porous Medium*, Journal of Applied Mechanics, vol. 27, pp. 329 – 332, 1957.

[14] T. Y. Na, *Computational Methods in Engineering Boundary value Problems of Mathematics in Science and Engineering*, Academic Press, vol. 145, New York, NY. USA, 1979.

[15] M. Muskat, J. Edwards and Ann Arbor, *Flow of Homogeneous Fluids*, USA, 1946.