*Nigerian Journal of Mathematics and Applications*
*Volume* **27**, (2018), 132 – 148. *Printed by Unilorin press*
©*Nig. J. Math. Appl.* *http://www.njmaman.com*

# Modeling The Fluid Flow for Seepage Problem: A Crank-Nicolson Approach

A. A. Hassan*, I. O. Abiala and J. S. Aroloye

Abstract

The two dimensional Laplace equation (which models the fluid flow of seepage problem) was solved by a finite difference method which is the Crank-Nicolson Method (CNM) subject to some boundary and initial conditions. We compared the numerical results of CNM to the Alternating Direction Explicit Finite Difference Scheme (ADES). First, we derived the finite differential form of the implicit, explicit and Crank-Nicolson methods for the given model and then presented an algorithm for each method. The resulting systems of linear algebraic equations for the CNM was solved using the MATLAB software. The solutions were presented graphically in three dimensions and interpreted. We also analysed the numerical stability of the CNM by matrix method and was found to be unconditionally stable. We observed that the seepage flow decreases with distance from the source (dam) and also the smaller the mesh size, the finer the decrease in the fluid seepage.

## 1. Introduction

The movement of fluid is a crucial behaviour in engineering and physics and this makes the study of its properties imperative so as to determine the seepage. Many

Received: 13/09/2018, Accepted: 24/11/2018, Revised: 29/11/2018.
2015 *Mathematics Subject Classification.* 00A71, & 03C98.
*Key words and phrases.* Crank-Nicolson; Explicit; Finite Difference; Implicit; Seepage
*Abbreviations.* Explicit Finite Difference Method (EFDM); Implicit Finite Difference Method (IFDM)
Department of Mathematics, University of Lagos, Nigeria
E-mail: amisu.hassan91@gmail.com, iabiala@unilag.edu.ng & saroloye@unilag.edu.ng

phenomena in science, engineering and mechanics involve the use of partial differential equation to study its prevailing conditions and behaviour. In fact, many of the advance partial differentials are not analytically tractable and involve the use of numerical method so as to obtain the optimum solutions to those problems. In general, solving these equations by classical analytical methods for arbitrary shapes is almost impossible. Since the seepage cannot be easily solved analytically, we therefore employed numerical method called Crank-Nicolson method in this research work. In numerical analysis, the CrankNicolson method is a finite difference method used for numerically solving the heat equation and similar partial differential equations. It is a second order method in time. It is implicit in time and can be written as an implicit RungeKutta method, and it is numerically stable. The method was developed by John Crank and Phyllis Nicolson in the mid 20th century.

For diffusion equations (and many other equations), it can be shown the CrankNicolson Method (CNM) is unconditionally stable. However, the approximate solutions can still contain (decaying) spurious oscillations if the ratio of time step $\Delta t$ times the thermal diffusivity to the square of space step, $\Delta x_2$, is large (typically larger than 1/2 per Von Neumann stability analysis). For this reason, whenever large time steps or high spatial resolution is necessary, the less accurate backward Euler method is often used, which is both stable and immune to oscillations. From an engineering standpoint, the CNM is a method for solving engineering problems such as stress analysis, heat transfer, fluid flow and electro magnetics by computer simulation. Millions of engineers and scientists worldwide use the CNM to predict the behaviour of structural, mechanical, thermal, electrical and chemical systems for both design and performance analyses. The equation for CrankNicolson method is a combination of the forward Euler method at $n$ and the backward Euler method at $n+1$ (note, however, that the method itself is not simply the average of those two methods, as the equation has an implicit dependence on the solution).

## 2. Mathematical Formulation

In this section, we present some preliminaries on finite difference method. For the purpose of this paper, we discuss the motivation of the Crank-Nicolson's method (CNM) over the explicit and implicit finite difference methods. We shall also present the stability analysis of these approaches.

2.1. **Representative Model Problem.** Let us consider the one-dimensional heat equation

$$\frac{\partial \phi(x,t)}{\partial t} = k\frac{\partial^2 \phi(x,t)}{\partial x^2} \tag{1}$$

In order to obtain a numerical solution to (1) using finite difference methods, we first need to define a set of grid points in the domain $\Omega$ as follows; (Hull, 2006) choose a state step size $\Delta x = \dfrac{b-a}{N}$ ($N$ is an integer) and a time step $\Delta t$.
We draw a set of vertical and horizontal lines across the domain $\Omega$, and identify all the points of intersection $(x_i, t_j)$ or simply $(i, j)$ where $x_i = a + i\Delta x$, $i = 0, 1, \cdots, N$ and $t_j = j\Delta t$, $j = 0, 1, \cdots$.
Suppose $\Omega = [a, b] \times [0, T]$, then choose $\Delta t = \dfrac{T}{M}$ ($M$ is an integer) and $t_j = j\Delta t$, $j = 0, 1, \cdots, M$. The figure below shows some nodes on the domain $\Omega$.

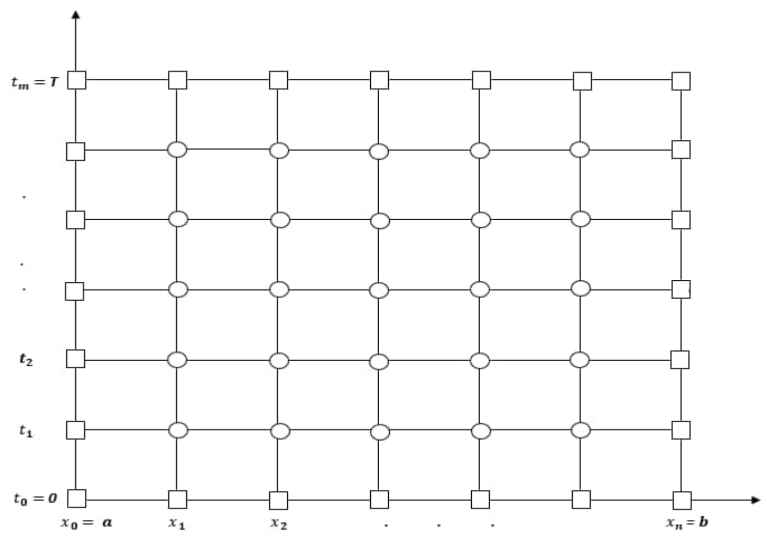



Figure 1: Grid points

2.2. **The Implicit Finite Difference Method (IFDM).** The IFDM makes use of the forward approximation for the time derivatives (Hull, 2006). The approximation for the time derivatives in equation (1) is

$$\frac{\partial \phi}{\partial t} = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta t} \tag{2}$$

where $\phi_{i,j} = \phi(i\Delta x, j\Delta t)$, $i = 1, 2, \ldots, m-1$, $j = N-1, N-2, \ldots, 1, 0$.
For the finite derivative of the R.H.S of (1), we use the central approximation

$$\frac{\partial \phi}{\partial x} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2\Delta x} \tag{3}$$

Similarly, in the second derivative, we use the standard approximation

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta x)^2} \tag{4}$$

Substituting (2) and (4) into (1), we have

$$\frac{\phi_{i,j+1}-\phi_{i,j}}{\Delta t}=k\left[\frac{\phi_{i+1,j}-2\phi_{i,j}+\phi_{i-1,j}}{(\Delta x)^2}\right] \tag{5}$$

Thus, simplifying (5), we have

$$\phi_{i,j+1}=\phi_{i,j}+\lambda(\phi_{i+1,j}-2\phi_{i,j}+\phi_{i-1,j}) \tag{6}$$

where $\lambda = k\dfrac{\Delta t}{(\Delta x)^2}$
From (6), we observe that the unknown first term $\phi_{i,j+1}$ is written in terms of the known values $\phi_{i,j}$, $\phi_{i-1,j}$ and $\phi_{i+1,j}$. Furthermore, it relates the unknown values at $(j+1)\Delta t$ to the function values at $j\Delta t$. Therefore if we are provided with the value of $\phi_{i,j+1}$ for all $i$, at time step $j$, we can compute the value of $\phi_{i,j}$ implicitly.

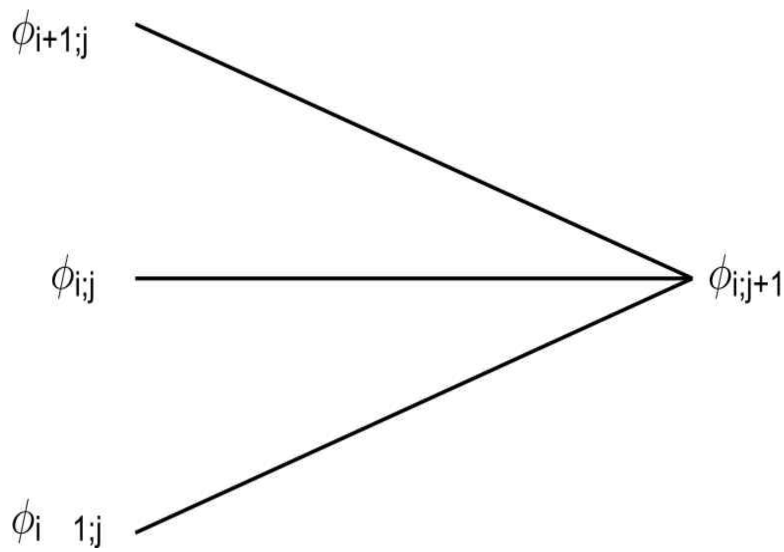


Figure 2: Implicit Finite Difference discretisation (Hull, 2006)

Upon evaluation of equation (6) using the corresponding grid points, we obtain a linear algebraic equation of the form,

$$\phi^{n+1}=A\phi^n+b^n \tag{7}$$

where

$$\phi^n=(\phi_1^n,\phi_2^n,\ldots,\phi_{N-1}^n)^T\ \in\ \mathbb{R}^{N-1} \tag{7a}$$

and

(7b)
$$A = \begin{pmatrix} 1+2\lambda & -\lambda & & \\ -\lambda & 1+2\lambda & -\lambda & \\ \ddots & \ddots & & \\ & -\lambda & 1+2\lambda & \end{pmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}$$

(7c)
$$b^n = \begin{pmatrix} \lambda\phi_0^n \\ 0 \\ \vdots \\ \lambda\phi_N^n \end{pmatrix} \in \mathbb{R}^{N-1}$$

It is easy to now implement a numerical code to solve equation (7).

2.3. **The Explicit Finite Difference Method (EFDM).** The EFDM assumes the values of the derivative at the point $(i, j)$ on the lattice to be the same as $(i, j+1)$ and it makes use of the backward approximation for the time derivative (Hull, 2006). Thus, equation (1) can be approximated as

$$\frac{\partial\phi}{\partial t} = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta t} \text{ as in (2)}$$

Meanwhile, for the first derivative of the R.H.S of (1), we have from the central approximation as

(8)
$$\frac{\partial\phi}{\partial x} = \frac{\phi_{i+1,j+1} - \phi_{i-1,j+1}}{2\Delta x}$$

For the second derivative, we have from the standard approximation

(9)
$$\frac{\partial^2\phi}{\partial x^2} = \frac{\phi_{i+1,j+1} - 2\phi_{i,j+1} + \phi_{i-1,j+1}}{(\Delta x)^2}$$

where $(i, j)$ denotes the nodes on the lattice, substituting (2) and (9) into (1) we have

(10)
$$\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta t} = k\left[\frac{\phi_{i+1,j+1} - 2\phi_{i,j+1} + \phi_{i-1,j+1}}{(\Delta x)^2}\right]$$

We put $\lambda = k\dfrac{\Delta t}{(\Delta x)^2}$ in (10) and simplify to get

(11)
$$-\lambda\phi_{i+1,j+1} + (1+2\lambda)\phi_{i,j+1} - \lambda\phi_{i-1,j+1} = \phi_{i,j}$$

From equation (11), observe that the unknown function $\phi_{i,j}$ is written explicitly in terms of the unknowns. Furthermore, it relates the unknown value at $j\Delta t$ to the known values at $(j+1)\Delta t$. In obtaining the values of $\phi(x, 0)$, we approximate backward until $t = 0$ using the explicit scheme (11).

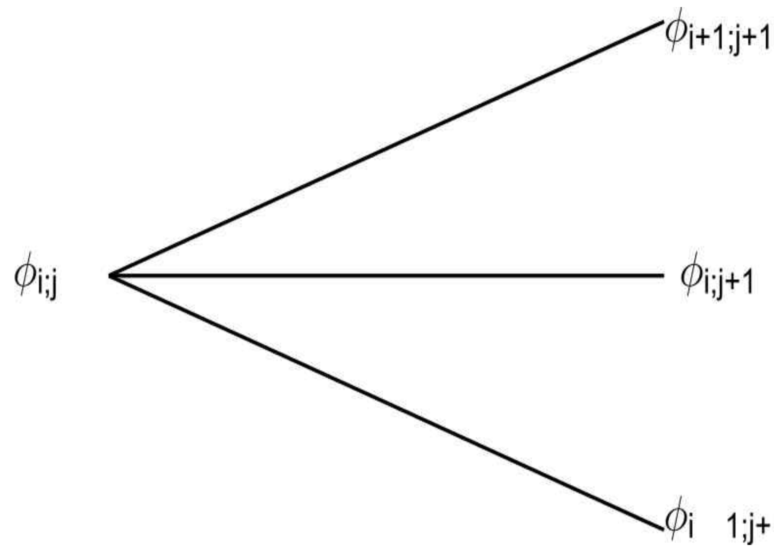



Figure 3: Explicit Finite Difference discretisation (Hull, 2006)

In vector notation, the explicit scheme can be written as:

$$\phi^{n+1} = B\phi^n + d^n \tag{12}$$

where

$$\phi^n = (\phi_1^n, \phi_2^n, \ldots, \phi_{N-1}^n)^T \;\in\; \mathbb{R}^{N-1} \tag{12a}$$

and

$$B = \begin{pmatrix} 1-2\lambda & \lambda & & \\ \lambda & 1-2\lambda & & \\ \ddots & \ddots & & \\ & \lambda & 1-2\lambda \end{pmatrix} \;\in\; \mathbb{R}^{(N-1)\times(N-1)} \tag{12b}$$

$$d^n = \begin{pmatrix} \lambda\phi_0^n \\ 0 \\ \vdots \\ \lambda\phi_N^n \end{pmatrix} \;\in\; \mathbb{R}^{N-1} \tag{12c}$$

2.4. **The Crank-Nicolson's Method (CNM).** The CNM is the average of the explicit scheme at $(i, j)$ and the implicit method at $(i, j+1)$. Thus, from equation (6) and (11), we have

$$\frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta t} = \frac{k}{2}\left[\frac{\phi_{i+1,j+1} - 2\phi_{i,j+1} + \phi_{i-1,j+1}}{(\Delta x)^2} + \frac{\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j}}{(\Delta x)^2}\right] \tag{13}$$

We simplify (13) and obtain

$$(14)\quad -\frac{\lambda}{2}\phi_{i-1,j+1} + (1+\lambda)\phi_{i,j+1} - \frac{\lambda}{2}\phi_{i+1,j+1} = \frac{\lambda}{2}\phi_{i-1,j} + (1-\lambda)\phi_{i,j} + \frac{\lambda}{2}\phi_{i+1,j}$$

where $\lambda = \dfrac{k\Delta t}{(\Delta x)^2}$.
Equation (14) is the CNM for the one dimensional heat equation (1) . The CNM enjoys stronger accuracy over the I.F.D.M and E.F.D.M, and it is unconditionally stable with the local truncation error $O((\Delta t)^2, (\Delta x)^2, (\Delta y)^2)$.

2.5. **Numerical Stability.** In numerical analysis, consistency is necessary but not a sufficient condition for convergence (Mark Davis, 2010) and the references therein. Round off errors incurred during calculations may lead to a blow up of the solution or erode the computation as a whole. Therefore, a scheme is stable if round off errors are not amplified in the calculations. The Fourier approach can be used to check if a scheme is stable. Assume that a numerical scheme admits a solution of the form

$$(15)\qquad V_j^n = a^{(x)}(\omega)\mathrm{e}^{ij\omega\Delta x}$$

where $\omega$ is the wave number and $i = \sqrt{-1}$.
Define

$$(16)\qquad G(\omega) = \frac{a^{(n+1)}(\omega)}{a^{(n)}(\omega)}$$

where $G(\omega)$ is an amplification factor which governs the growth of the Fourier component $a(\omega)$.
The Von Neuman stability condition is given by

$$|G(\omega)| \leq 1 \text{ for } 0 \leq \omega\Delta x \leq \pi$$

We show that the explicit scheme is stable if and only if $\lambda \leq \frac{1}{2}$. Called conditionally stable, and the implicit and Crank-Nicolson schemes are stable for any values of $\lambda$, called unconditionally stable, see (Mark Davis, 2010).

2.6. **Stability Analysis.** For the explicit scheme, see (M. Davies, 2010), we get upon substituting (15) into (16) that

$$a^{(n+1)}(\omega)\mathrm{e}^{ij\omega\Delta x} = \lambda a^{(n)}(\omega)\mathrm{e}^{i(j+1)\omega\Delta x} + (1-2\lambda)a^{(n)}(\omega)\mathrm{e}^{ij\omega\Delta x} + \lambda a^{(n)}(\omega)\mathrm{e}^{i(j-1)\omega\Delta x}$$

$$\Rightarrow G(\omega) = \frac{G^{(n+1)}(\omega)}{G^{(n)}(\omega)} = \lambda\mathrm{e}^{ij\omega\Delta x} + (1-2\lambda) + \lambda\mathrm{e}^{-ij\omega\Delta x}$$

The Von Neuman stability condition then is

$$\begin{array}{llcl} |G(\omega)| \leq 1 & \Leftrightarrow & |\lambda e^{ij\omega\Delta x} + (1-2\lambda) + \lambda e^{-ij\omega\Delta x}| & \leq 1 \\ & \Leftrightarrow & |(1-2\lambda) + 2\lambda\cos(\omega\Delta x)| & \leq 1 \\ & \Leftrightarrow & |1 - 4\lambda\sin^2(\frac{\omega\Delta x}{2})| & \leq 1 \qquad [\cos 2x = 1 - 2\sin^2 x] \\ & \Leftrightarrow & 0 \leq 4\lambda\sin^2(\frac{\omega\Delta x}{2})| & \leq 2 \\ & \Leftrightarrow & 0 \leq \lambda & \leq \frac{1}{2\sin^2(\frac{\omega\Delta x}{2})} \end{array}$$

for all $0 \leq \omega\Delta x \leq \pi$.
This is equivalent to $0 \leq \lambda \leq \frac{1}{2}$

## 3. Numerical Solution of a Seepage Problem

In this section, we develop the seepage Laplace equation using Crank-Nicolson Method (CNM). In what follows, we discuss the stability analysis and implementation of our scheme. Finally we present numerical result to show the efficiency of the method.

3.1. **Seepage Problem.** We consider a two dimensional element of soil of dimensions $dx$ and $dy$ in the $x$ and $y$ direction respectively. It is assumed that the soil is homogeneous and isotropic with respect to permeability. Furthermore, the pore fluid is assumed to be incompressible. Thus, the governing differential equation for the fluid flow is described by

$$\begin{array}{ll} \bigtriangledown^2 u(x,y) & = 0 \\ u(0,y) & = u(x,0) = u(a,y) = 0 \\ u(x,b) & = V_0 \end{array} \tag{17}$$

where $V_0 = 1 - e^{-y}$, $0 \leq y \leq \infty$
The Crank-Nicolson's method is obtained by averaging the implicit and explicit finite difference methods. Thus, the Crank-Nicolson's scheme for (17) is as follows

$$\begin{array}{c} \bigtriangledown^2 u(x,y) = 0 \\ \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2} = 0 \Leftrightarrow \\ \dfrac{1}{2h^2}[u_{i+1,j} - 2u_{i,j} + u_{i-1,j} + u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}] \\ + \quad \dfrac{1}{2h^2}[u_{i,j+1} - 2u_{i,j} + u_{i,j-1} + u_{i+1,j+1} - 2u_{i+1,j} + u_{i+1,j-1}] = 0 \end{array} \tag{18}$$

(19)

$$\Rightarrow \frac{1}{2h^2}[-4u_{i,j} - u_{i+1,j} + 2u_{i+1,j+1} - u_{i,j+1} + u_{i-1,j} + u_{i-1,j+1} + u_{i,j-1} + u_{i+1,j-1}] = 0$$

$$\Rightarrow -4u_{i,j} - u_{i+1,j} + 2u_{i+1,j+1} - u_{i,j+1} + u_{i-1,j} + u_{i-1,j+1} + u_{i,j-1} + u_{i+1,j-1} = 0 \tag{20}$$

In order to define the C.N.M (i.e finite difference), we need a grid on the domain $\Omega$.

Let $N+1$ be given and define $h = \dfrac{1}{N+1}$. Then we define the grid as the collection of points

$$\mathcal{T}_h = \{(nh, mh) : \; n, m = 0, 1, 2, \ldots, N+1\}$$

We denote the interior points by

$$\mathcal{T}_I = \{(nh, mh) : \; n, m = 1, 2, \ldots, N\}$$

and the boundary points as $\mathcal{T}_b = \mathcal{T}_h \setminus \mathcal{T}_I$. In particular we consider the following grid:

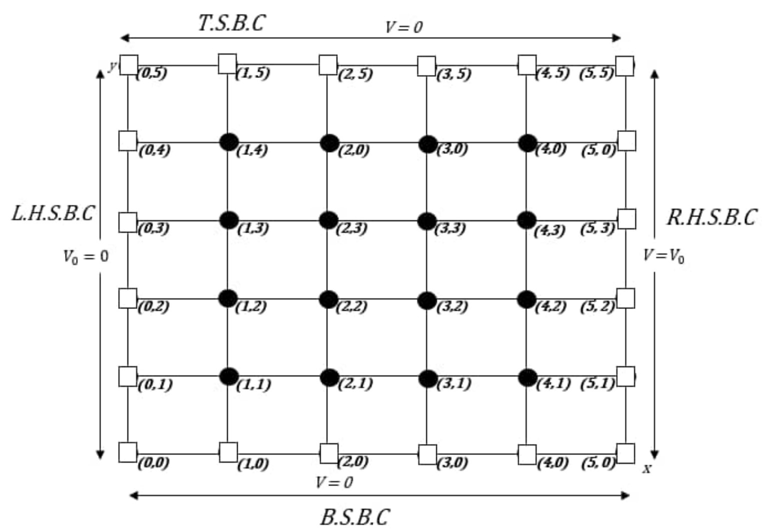



Figure 4: Grid $N = 4$, B.S.B.C grid points $\mathcal{T}_b$ are in □ and $\mathcal{T}_I$ are in ∘.

Applying (20) on the grid points in Figure (4), it is straight forward to obtain a system of equation of the form $Ax = B$ which can be solved by any known method. However, for the purpose of this article, we solve the seepage problem modelled by the Laplace equation (17) in the spirit of Nyachwaya et al., (2014). We shall equally compare our result with the results in Nwachwaya et al., (2014) for $j = 1$ and $i = 1, 2, \cdots, 15$.

Now, from (20), we fix $j = 1$, and vary $i = 1, 2, \cdots, 15$ as follows:

$$
\begin{array}{ll}
i = 1: & -4u_{1,1} - u_{2,1} = -V_0 \\
i = 2: & -4u_{2,1} - u_{3,1} + u_{1,1} = 0 \\
i = 3: & -4u_{3,1} - u_{4,1} + u_{2,1} = 0 \\
i = 4: & -4u_{4,1} - u_{5,1} + u_{3,1} = 0 \\
i = 5: & -4u_{5,1} - u_{6,1} + u_{4,1} = 0 \\
i = 6: & -4u_{6,1} - u_{7,1} + u_{5,1} = 0 \\
i = 7: & -4u_{7,1} - u_{8,1} + u_{6,1} = 0 \\
i = 8: & -4u_{8,1} - u_{9,1} + u_{7,1} = 0 \\
i = 9: & -4u_{9,1} - u_{10,1} + u_{8,1} = 0 \\
i = 10: & -4u_{10,1} - u_{11,1} + u_{9,1} = 0 \\
i = 11: & -4u_{11,1} - u_{12,1} + u_{10,1} = 0 \\
i = 12: & -4u_{12,1} - u_{13,1} + u_{11,1} = 0 \\
i = 13: & -4u_{13,1} - u_{14,1} + u_{12,1} = 0 \\
i = 14: & -4u_{14,1} - u_{15,1} + u_{13,1} = 0 \\
i = 15: & -4u_{15,1} - u_{16,1} + u_{14,1} = 0
\end{array}
\tag{21}
$$

3.2. **Implementation of the Method.** Here, we consider (17) with a Dirichilet boundary condition motivated in Nyachwaya et al., (2014) and assemble the algebraic equation

$$
\begin{bmatrix}
-4 & -1 & 0 & 0 & 0 & 0 \\
1 & -4 & -1 & 0 & 0 & 0 \\
0 & 1 & -4 & -1 & 0 & 0 \\
0 & 0 & 1 & -4 & -1 & 0 \\
\vdots & \vdots & & & & \vdots \\
0 & 0 & 0 & 1 & -4 & -1 \\
0 & 0 & 0 & \dots & 1 & -4
\end{bmatrix}
\begin{bmatrix}
u_{1,1} \\ u_{2,1} \\ u_{3,1} \\ u_{4,1} \\ \vdots \\ u_{15,1}
\end{bmatrix}
=
\begin{bmatrix}
-1.7293279434 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0
\end{bmatrix}
$$

We then use MATLAB software to obtain:

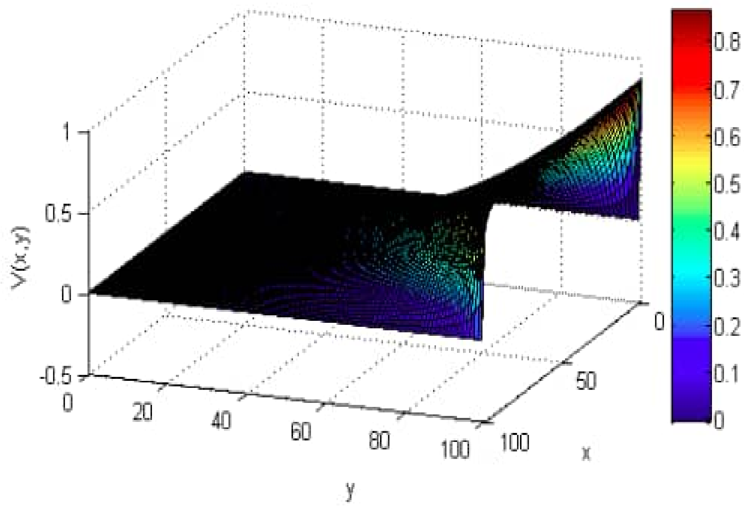
$u_{1,1} = 0.408239301803523$
$u_{2,1} = 0.096372226312684$
$u_{3,1} = 0.022750396552788$
$u_{4,1} = 0.005370640101531$
$u_{5,1} = 0.001267836146663$
$u_{6,1} = 0.000299295514877$
$u_{7,1} = 0.000070654087155$
$u_{8,1} = 0.000016679166258$
$u_{9,1} = 0.000003937422123$
$u_{10,1} = 0.000000929477767$
$u_{11,1} = 0.000000219511054$
$u_{12,1} = 0.000000051433552$
$u_{13,1} = 0.000000013776844$
$u_{14,1} = 0.000000003673825$
$u_{15,1} = 0.000000000918456$

Table 1: The Method (20)

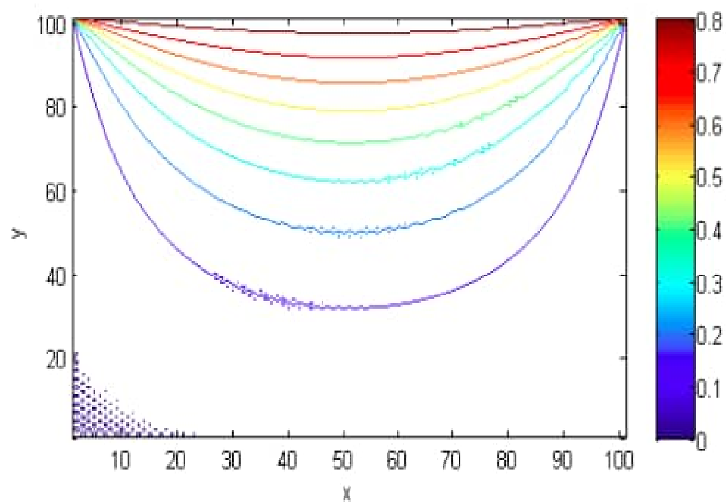
| | j= 1 | j= 2 | j= 3 |
|---|---|---|---|
| i= 1 | 0:408239301803523 | 0:448629689929089 | 0:463488483341419 |
| i= 2 | 0:096372226312684 | 0:105907103547917 | 0:109414788856854 |
| i= 3 | 0:022750396552788 | 0:0250012757374190 | 0:025829327914005 |
| i= 4 | 0:005370640107531 | 0:005902000598240 | 0:006097477200834 |
| i= 5 | 0:001267836146663 | 0:001393273344447 | 0:001439419110670 |
| i= 6 | 0:000299295514877 | 0:0003289071220455 | 0:000339800758155 |
| i= 7 | 0:000070654087155 | 0:000077644462629 | 0:000080216078252 |
| i= 8 | 0:000016679166258 | 0:000018329369940 | 0:000018936445948 |
| i= 9 | 0:000003937422123 | 0:000004326982871 | 0:000004470294261 |
| i= 10 | 0:000000929477767 | 0:000001021438457 | 0:000001055268904 |
| i= 11 | 0:000000219511054 | 0:000000241229043 | 0:000000249218649 |
| i= 12 | 0:000000051433552 | 0:0000000056522286 | 0:000000058394327 |
| i= 13 | 0:000000013776844 | 0:000000015139898 | 0:000000015641338 |
| i= 14 | 0:000000003673825 | 0:0000000040373086 | 0:000000004171023 |
| i= 15 | 0:000000000918456 | 0:000000001009327 | 0:000000001042756 |

From table (1), we observe that the seepage flow decreases as $i$ increases from $i = 1, 2, \ldots, 15$. Also, for each value of $i$, $i = 1, 2, \ldots, 15$, the seepage flow increased as $j$ increases from $j = 1, 2, 3$, and the seepage flow decreases as the value of $i$ increases steadily. However, this behaviour is due to the large volume of water at the surface and the source. The flow of the fluid also experiences friction due to viscosity and medium particles.

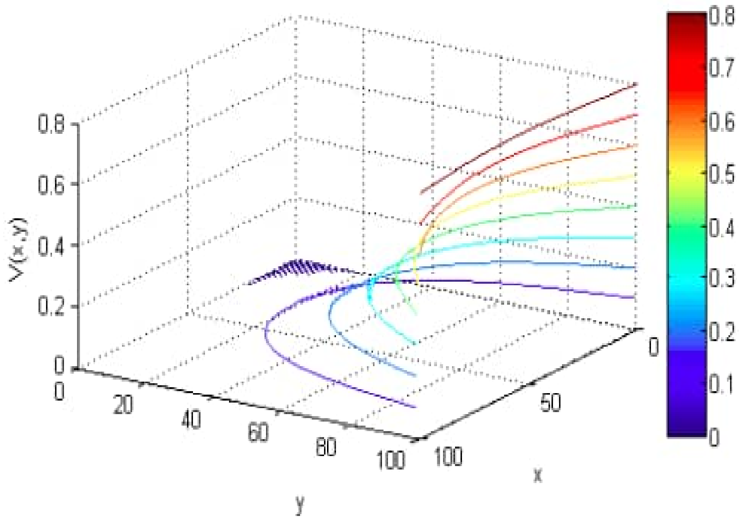
Next, we write a MATLAB program to plot the behaviour of our method in (19) to study the seepage problem. Figure (5a) shows the surface plot of the solution, figure (5b) depicts the contour plot of the solution, figure (5c) is a contour plot with coloured lines, and figure (5d) is the mesh plot of the solution.



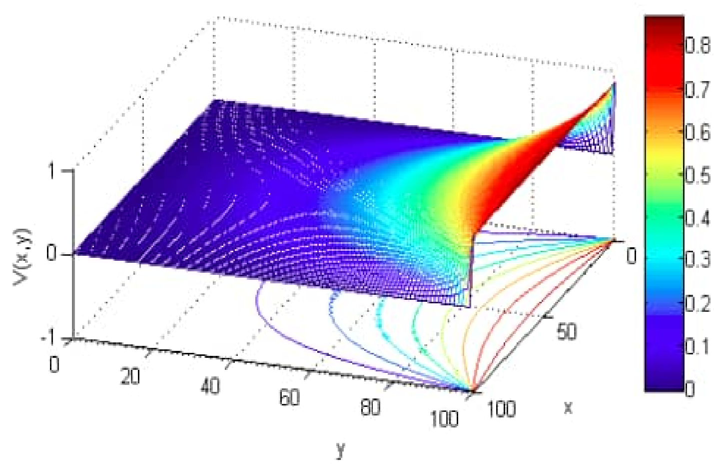

(a): Surface plot of the solution




(b): Contour plot of the solution




(c): Contour plot with coloured line




(d): Mesh plot of the solution

**Figure 5: Graphical representation**

From the the above diagrams, we observe that the plot is not smooth. This is due to the fact that the governing differential equations is only satisfied at a selected number of nodes within the domain of evaluation. The solutions $u(x, y)$ of equation (17) decrease with increase length of the fluid seepage from the source.

The fundamental solution of the equation (17) which we developed into (21) through (20) efficiently solved the seepage problem, see table (1) and figure (5d) above. Our results agrees and compare favourably well with solutions to the seepage problem, and it is superior to the results in Nyachwaya et al., (2014).

3.3. **Stability Result.** We apply matrix method to analyse the stability of our method. Again, let us consider our scheme (20). If we fix $j$ and vary $i$, $i = 1, 2, \dots, (N-2), (N-1)$ we have

(22)

$$\begin{array}{lcl} -4u_{1,j} - u_{2,j} + 2u_{2,j+1} - u_{1,j+1} + u_{0,j} + u_{0,j+1} + u_{1,j-1} + u_{2,j-1} & = & 0 \\ -4u_{2,j} - u_{3,j} + 2u_{3,j+1} - u_{2,j+1} + u_{1,j} + u_{1,j+1} + u_{2,j-1} + u_{3,j-1} & = & 0 \\ \vdots \qquad\qquad \vdots \qquad\qquad \vdots & = & \vdots \\ -4u_{N-1,j} - u_{N-2,j} + 2u_{N-2,j+1} - u_{N-1,j+1} + u_{N,j} + u_{N,j+1} + u_{N-1,j-1} + u_{N-2,j-1} & = & 0 \end{array}$$

In a more compact form, we re-write (22) into the form

$$\phi u_{N-1} = 0 \qquad (23)$$

It is known that the Crank-Nicolson method is unconditionally stable, and according to Riley et al., (2011), the stability of (23) can be investigated by computing the eigenvalue of the coefficient matrix $\phi$. Thus, if the maximum eigenvalue of $\phi$ is less than or equal to 1, then the method is said to be stable. See also Nyachwaya et al., (2014), and the references therein. In what follows, we write a MATLAB program to compute $|\phi\lambda - I| = 0$ from (23) and obtained $\lambda_i$, $i = 1, 2, \dots, 15$ such that $\lambda_{\max} \leq 1$ which generates the stability of our method.

Table 2: The C.N.M (20)

| Node | j= 1 | j= 2 | j= 3 |
|---|---|---|---|
| i= 1 | 0:408239301803523 | 0:448629689929089 | 0:463488483341419 |
| i= 2 | 0:096372226312684 | 0:105907103547917 | 0:109414788856854 |
| i= 3 | 0:022750396552788 | 0:0250012757374190 | 0:025829327914005 |
| i= 4 | 0:005370640107531 | 0:005902000598240 | 0:006097477200834 |
| i= 5 | 0:001267836146663 | 0:001393273344447 | 0:001439419110670 |
| i= 6 | 0:000299295514877 | 0:0003289071220455 | 0:000339800758155 |
| i= 7 | 0:000070654087155 | 0:000077644462629 | 0:000080216078252 |
| i= 8 | 0:000016679166258 | 0:000018329369940 | 0:000018936445948 |
| i= 9 | 0:000003937422123 | 0:000004326982871 | 0:000004470294261 |
| i= 10 | 0:000000929477767 | 0:000001021438457 | 0:000001055268904 |
| i= 11 | 0:000000219511054 | 0:000000241229043 | 0:000000249218649 |
| i= 12 | 0:000000051433552 | 0:000000056522286 | 0:000000058394327 |
| i= 13 | 0:000000013776844 | 0:000000015139898 | 0:000000015641338 |
| i= 14 | 0:000000003673825 | 0:0000000040373086 | 0:000000004171023 |
| i= 15 | 0:000000000918456 | 0:000000001009327 | 0:000000001042756 |

Table 3: Method in Nyachwaya et. al., (2014)
For Case 1: ADES Results

| | j= 1 | j= 2 | j= 3 |
|---|---|---|---|
| i= 1 | 0:5601236 | 0:6670508 | 0:7305356 |
| i= 2 | 0:3699449 | 0:4678016 | 0:5372144 |
| i= 3 | 0:242888 | 0:3245777 | 0:3896345 |
| i= 4 | 0:1586531 | 0:2231869 | 0:2792263 |
| i= 5 | 0:1031687 | 0:1523049 | 0:1980823 |
| i= 6 | 0:0682435 | 0:1039865 | 0:1396645 |
| i= 7 | 0:04313089 | 0:06951022 | 0:0970872 |
| i= 8 | 0:02774829 | 0:04649375 | 0:0671340 |
| i= 9 | 0:01779631 | 0:03087895 | 0:0460417 |
| i= 10 | 0:0111363 | 0:02018054 | 0:0311984 |
| i= 11 | 0:007211911 | 0:01311101 | 0:0209239 |
| i= 12 | 0:004524799 | 0:00813458 | 0:0136494 |
| i= 13 | 0:002764718 | 0:05125716 | 0:0087360 |
| i= 14 | 0:001575447 | 0:002992201 | 0:0051651 |
| i= 15 | 0:007133674 | 0:001364700 | 0:0039536 |

3.4. **Comparison of Result.** From tables (2) and (3), we observed that our method performs efficiently more than the method in Nyachwaya et al., (2014). Because the seepage problem decreases more for every increase in table (2).

3.5. **Discussion of Results.** Table 2 depicts the solution of the seepage problem obtained by Crank-Nicolson Method (CNM). Table 3 is the seepage solution obtained by alternating direction explicit finite difference scheme (ADES) case 1 in Nyachwaya et al., (2014). We observed that the seepage flow decreases as $i$ increases from $i = 1, 2, \ldots, 15$ and the seepage flow increases as $j$ increases from $j = 1, 2, 3$ in both methods. However, we noticed from the table of results that the CNM performs efficiently better than the ADES of Nyachwaya et al., (2014) as the flow decreases more as the mesh point becomes smaller. It becomes obvious that the seepage flow decrease in the ADES is slower than that of CNM.

## 4. Conclusion

In this paper, we developed a consistent Crank-Nicolson Method (CNM) for the numerical solution of seepage problem. The scheme developed was found to be stable for $\phi \leq 1$. Since the model used in this work was evaluated at a discrete point within a domain, the surface of the plots was found to be naturally not smooth.

From the results, we deduced that for a given value of $x$, the solution $u(x, y)$ increases to almost one as $y \to \infty$. Also, for a given value of $y$, the solution $u(x, y)$ decreases indefinitely as $x \to \infty$. And we observed that the seepage gradually goes to null as the mesh sizes become smaller, which shows that the fluid flow of seepage problem may eventually stop.

## References

[1] B. Abdelfatah and M. Nabil, (2006), "Solution to a Semi-Linear Pseudoparabolic Problem with Integral Conditions", *Electronic Journal of Differential Equbersations*, Vol. 115, Pp. 118.

[2] O. A. Al-Damluji, M. Y. Fattah and R. A. Al-Adthami, (2004), "Solution of two-dimensional steady-state flow field problems by the boundary element method", *Journal of Engineering and Technology*, 23(12), 750766.

[3] G. Barenblatt, I. Zheltov and I. Kochina, (1960), "Basic Concepts in the Theory of Seepage of Homogeneous Liquids in Fissured Rocks", *J. Appl. Mat. Mech.*, Vol. 24, Pp. 12861303.

[4] H.R. Cedergren, (1992), Seepage, Drainage and Flow Nets, Wiley, New York.

[5] K.Chaiyo, P. Rattemadecho, and S. Chamtasiriwan, The method of fundamental solutions for solving method of fundamental solution for solving free boundary saturated seepage problem.

[6] B.M Das, (1983), Advanced Soil Mechanics, McGraw-Hill, New York.

[7] M. Davis (2010), Finite difference methods, M. Sc course in Mathematics and Finance, Imperial college, London.

[8] C.S. Desai and J.T. Christain, (1977), Numerical Methods in Geotechnical Engineering. McGraw-Hill, New York.

[9] J. Douglas and D. Peaceman, (1955), "Numerical Solution of Two-Dimensional Heat Flow Problems", *American Institute of Chemical Engineering Journal*, Vol. 1, Pp. 505512.

[10] S. Johnson, Y. Saad and M. Schultz, (1987), "Alternating Direction Methods Multiprocessors", *SIAM, J.sci. Statist. Comput.*, Vol. 8, Pp. 686700.

[11] J. Mosler, (2005). A Novel Algorithmic Framework for the Numerical Implementation of Locally Embedded Strong Discontinuities. *Computer Methods in Applied Mechanics and Engineering*, 194, 4731-4757

[12] S. Murakami, (2012), Continuum Damage Mechanics: A Continuum Mechanics Approach to the Analysis of Damage and Fracture, Springer.

[13] B.J. Noye and K.J. Hayman, (1994), "New LOD and ADI Methods for the Two-Dimensional Diffusion Equation", *Journal of Computer Mathematics*, Vol. 51, Pp. 215228.

[14] N. Nyachwaya and K.S. Johana, (2014), Finite Difference Solution of Seepage Equation, *Journal of Mathematical Model Fluid Flow*, 2(4).

[15] K. Rao Sankar (2004), "Introduction to Partial Differential Equations", *Prentice Hall of India*: New Delhi.

[16] M. Rezk, and A. Senoon, (2011), "Analitical solution of seepage through earth dam whit an internal core", Alex. Eng. J., Alex. Univ., 50, 111115.

[17] K.F. Riley, M.P. Hobson, and S.J Bence, (2006), *Mathematical Method for Physics and Engineering; a comprehensive guide*. Cambridge University Press.

[18] T.R. Taha and M.J. Ablowitz (1984), "Analytical and Numerical Aspects of Certain Nonlinear Evolution Equations III, Nonlinear Korteweg-De Vries Equation", *International Journal of Nonlinear Sciences*, Vol. 17, Pp. 55.

[19] J. Toth, (1962), A Theory of Groundwater motion in small in Drainage Basins in Central Alberta, Canada, *Journal of Geophysical Research*, 67(11): 4375-4387.

[20] J. Toth (1963), A Theoretical Analysis of Ground water flow in small Drainage Basins, *Journal of Geophysical Research*, 68(16): 4795-4812.

[21] F.B. Tracy, D.S. Howington and J. Hensley, (2005), Application of the pseudo-transient technique to a real-world unsaturated flow ground water problem Proc: ICCS, Springer Berlin, Atlantic GA. Pp(66-73).

[22] H.F. Wang, M.P. Anderson, (1995), Introduction to Ground Water Modelling: Finite Difference and Finite Element Methods, Academic Press.