

Research Article

An Explicit Stabilized Runge–Kutta–Legendre Super Time-Stepping Scheme for the Richards Equation

Ramesh Chandra Timsina ¹, Harihar Khanal,² and Kedar Nath Uprety³

¹Department of Mathematics, Patan Multiple Campus, Tribhuvan University, Kathmandu, Nepal

²Department of Mathematics, Embry-Riddle Aeronautical University, Daytona Beach, FL, USA

³Central Department of Mathematics, Tribhuvan University, Kathmandu, Nepal

Correspondence should be addressed to Ramesh Chandra Timsina; timsinaramesh72@yahoo.com

Received 28 January 2021; Revised 20 March 2021; Accepted 3 April 2021; Published 17 April 2021

Academic Editor: Adrian Neagu

Copyright © 2021 Ramesh Chandra Timsina et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We solve one-dimensional Kirchhoff transformed Richards equation numerically using finite difference method with various time-stepping schemes, forward in time central in space (FTCS), backward in time central in space (BTCS), Crank–Nicolson (CN), and a stabilized Runge–Kutta–Legendre super time-stepping (RKL), and compare their performances.

1. Introduction

The prediction of fluid movement in unsaturated porous media is important in many branches of science and engineering such as soil sciences, agricultural engineering, environmental engineering, and ground water hydrology. There are so many problems related to the movement of water in the unsaturated soil. One of the most important problems is the transfer of the different contaminants from ground surface to groundwater through the unsaturated zone. Flow of water through unsaturated porous media is described by Richards equation which is derived by combining the Darcy's law and mass conservation equation.

The Richards equation can be written in different forms with ψ (the pressure head), or θ (the volumetric moisture content), as the dependent variable. We consider the Richards equation in the following (mixed) form [1]:

$$\frac{\partial \theta}{\partial t} - \nabla \cdot (K(\psi) \nabla \psi) + \frac{\partial K}{\partial z} = 0, \quad (1)$$

where $K(\psi)$ is the unsaturated hydraulic conductivity and z is the vertical dimension (downward positive).

The constitutive relationship between $\theta = \theta(z, t)$ and $\psi = \psi(z, t)$ allows to express the Richards equation (1) with a single dependent variable ψ or θ .

The hydraulic conductivity $K(\psi)$ describes the ease with which water can move through pore space and depends on the intrinsic permeability of the material and the properties of fluid such as degree of saturation, density, and viscosity [2]. There are many empirical formulations for the hydraulic conductivity $K(\psi)$ and the moisture content $\theta(\psi)$ functions. We use the following popular model from groundwater hydrology due to Haverkamp et al. found at Celia's paper [3] which describes these constitutive relations as the continuous functions of ψ .

$$\begin{aligned} K(\psi) &= K_s \frac{A}{\alpha + |\psi|^\gamma}, \\ \theta(\psi) &= \theta_r + \frac{\alpha(\theta_s - \theta_r)}{\alpha + |\psi|^\beta}, \end{aligned} \quad (2)$$

where θ_s and θ_r represent the saturated and residual moisture content, respectively, K_s corresponds to the saturated hydraulic conductivity, and A , α , β , and γ are dimensionless soil parameters.

Both the hydraulic conductivity and the moisture content functions K and θ in (2) are highly nonlinear since they can dramatically change over a small range of ψ . As such, the Richards equation (1) becomes a highly nonlinear partial differential equation and its analytic solutions are limited to some simplified cases with no practical importance. Because of the lack of analytic solution to the Richards equation, study on the computational methods for the unsaturated flow problem has grown significantly. Several numerical schemes based on finite difference, finite element, and finite volume spatial discretization techniques to the partial differential equation have been developed to approximate the solution of the Richards equation [4–9]. Different numerical approaches yield different accuracy. Adaptive time-stepping strategies are studied in [10].

Due to the stability restriction of the explicit schemes for the parabolic partial differential equations, it seems there has not been much attention given to develop explicit schemes for the Richards equation and extensive amount of research work is devoted to developing implicit schemes. Due to the highly nonlinear nature of the problem, to implement implicit schemes, no matter which numerical approaches we use, the problem has to be linearized somehow at some stage. As such, some iterative approaches need to be applied to tackle this highly nonlinear problem [3, 11]. The implicit schemes are unconditionally stable, but with the iterative process involved, they all turn out to be computationally expensive and in certain circumstances, unreliable.

Recent trends in computational approach are to develop efficient parallel algorithm for high-performing computers. To be able to parallelize the numerical procedures, the algorithm should be iteration free. For this, an explicit scheme is the best option. In [12], a linearized Richards equation model is studied. In [8], stability analysis of an explicit scheme for the Richards equation is studied.

In this paper, we use Kirchhoff integral transform to reduce the highly nonlinear equation to a functional linear parabolic equation and solve it numerically.

The main purpose of this paper is to implement a relatively new numerical approach (super time-stepping to stabilize an explicit scheme [13] to transformed Richards equation). The work here presented describes and verifies the employment and accuracy of a stabilized Runge–Kutta–Legendre super time-stepping strategy (RKL) to an explicit finite difference scheme to simulate flow in unsaturated porous media. The advantage of using our approach is that it is unconditionally stable, easy to implement, can be easily extended to problems in higher dimensions, and can be easily parallelized.

This paper is organized as follows: in Section 2, we present Kirchhoff transformation to (spatially) linearize the Richards equation. In Section 3, we present the numerical methods based on finite difference method with different time-stepping schemes. In Section 4, we solve multiple test

cases and compare the results. Finally, in Section 5, we present our conclusions.

2. Simplified One-Dimensional Model

We consider Richards equation (1) in one space dimension with no source term.

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} \left(K(\psi) \frac{\partial \psi}{\partial z} \right) - \frac{\partial K}{\partial z}. \quad (3)$$

Richards equation (3) is typically used to simulate infiltration experiments (in both lab and field scale). These experiments begin with a dry soil and then water is poured on top of the ground surface, showing a clear connection with the Darcy's law. We assume that the infiltration with known surface flux does not exceed the infiltration intensity and does not generate runoff. That is, we use the following initial and boundary conditions:

$$\begin{cases} \psi(z, 0) = \psi_0(z), & 0 < z < L, \\ K(\psi) - K(\psi) \frac{\partial \psi}{\partial z} = q(t), & z = 0, t > 0, \\ \psi(L, t) = \beta(t), & t > 0. \end{cases} \quad (4)$$

2.1. Kirchhoff Integral Transform. To apply Kirchhoff integral transformation to equation (3), we let $h = \psi - z$ and define

$$\phi(h) = \int_0^h \bar{K}(\lambda) d\lambda. \quad (5)$$

Since $K(h) > 0$ from (2), the function $\phi(h)$ is strictly increasing with $\bar{K}(h) = K(\psi)$. Taking derivative of both sides of the transformation, we obtain

$$\frac{\partial \phi}{\partial z} = \frac{\partial \phi}{\partial h} \frac{\partial h}{\partial z} = \bar{K}(h) \frac{\partial(\psi - z)}{\partial z} \quad (6)$$

$$= K(\psi) \left(\frac{\partial \psi}{\partial z} - 1 \right) = K(\psi) \frac{\partial \psi}{\partial z} - K(\psi).$$

Again, taking derivative of equation (6),

$$\frac{\partial^2 \phi}{\partial z^2} = \frac{\partial(K(\psi)(\partial \psi / \partial z))}{\partial z} - \frac{\partial}{\partial z} (K(\psi)). \quad (7)$$

Using equation (7), the Richards equation (3) takes the form

$$\frac{\partial \bar{\theta}}{\partial t} = \frac{\partial^2 \phi}{\partial z^2}, \quad (8)$$

with $\bar{\theta}(\phi) = \theta(h)$. The corresponding initial and boundary conditions for the transformed equation (8) take the following form:

$$\left\{ \begin{array}{l} \phi(z, 0) = \phi_0(z), \quad 0 < z < L, \\ \frac{\partial \phi}{\partial z} = \bar{q}(t), \\ z = 0, \quad t > 0, \\ \phi(L, t) = \bar{\beta}(t), \quad t > 0. \end{array} \right. \quad (9)$$

The Kirchhoff transformation transformed the doubly nonlinear equation (3) to a nonlinear parabolic problem (8). Also, we note that the Kirchhoff transformation preserves the uniqueness result for the transformed problem.

3. Numerical Method

To solve the transformed equation (8) numerically with the prescribed initial and boundary conditions (9), it is more convenient to have a single state variable. For this, we assume θ and ϕ are single valued continuous functions of one another and rearrange to obtain

$$\begin{aligned} \frac{\partial \theta}{\partial t} &= \frac{\partial \theta}{\partial \phi} \frac{\partial \phi}{\partial t} = \left(\frac{1}{(\partial \phi / \partial \theta)} \right) \frac{\partial \phi}{\partial t}, \\ \frac{\partial \phi}{\partial \theta} &= \frac{\partial \phi}{\partial h} \frac{\partial h}{\partial \theta} \end{aligned} \quad (10)$$

Differentiating (2) and (9) with respect to h , we get

$$\begin{aligned} \frac{\partial \theta}{\partial h} &= \alpha(\theta_s - \theta_r)(\alpha + |h|^\beta)^{-2} \cdot \beta |h|^{\beta-1}, \\ \frac{\partial \phi}{\partial h} &= \bar{K}(h) = K(\psi). \end{aligned} \quad (11)$$

Using (10) and (11), the transformed Richards equation (8) takes the form

$$c(\phi) \frac{\partial \phi}{\partial t} = \frac{\partial^2 \phi}{\partial z^2}, \quad (12)$$

where the functional coefficient c depends on ϕ through h as

$$c(\phi(h)) = \frac{\alpha \beta (\theta_s - \theta_r) |h|^{\beta-1}}{\bar{K}(h) (\alpha + |h|^\beta)^2}. \quad (13)$$

3.1. Finite Difference Discretization. Let $\Delta z = L/M$ and $\Delta t = T/N$. We construct a grid (z_j, t_n) , with $z_j = j\Delta z$, $j = 0, 1, 2, \dots, M$ and $t_n = n\Delta t$, $n = 0, 1, 2, \dots, N$. Let ϕ_j^n denote $\phi(z_j, t_n)$. The partial differential equation (12) can be approximated using forward difference in time and central difference in space as

$$\begin{aligned} \frac{\partial \phi}{\partial t} \Big|_{(z_j, t_n)} &\approx \frac{\phi_j^{n+1} - \phi_j^n}{\Delta t}, \\ \frac{\partial^2 \phi}{\partial z^2} \Big|_{(z_j, t_n)} &\approx \frac{\phi_{j-1}^n - 2\phi_j^n + \phi_{j+1}^n}{\Delta z^2}. \end{aligned} \quad (14)$$

Let $0 \leq \lambda \leq 1$. Using a weighted average of the derivative $(\partial^2 \phi / \partial z^2)$ at two time levels, t_n and t_{n+1} , equation (12) can be discretized as

$$\begin{aligned} \phi_j^{n+1} &= \phi_j^n + \sigma_j^n \left[\lambda (\phi_{j-1}^{n+1} - 2\phi_j^{n+1} + \phi_{j+1}^{n+1}) \right. \\ &\quad \left. + (1 - \lambda) (\phi_{j-1}^n - 2\phi_j^n + \phi_{j+1}^n) \right], \end{aligned} \quad (15)$$

where $\sigma_j^n = \Delta t / (c_j^n \Delta z^2)$.

Equation (15) is used to update the values of ϕ_j^{n+1} for the internal nodes. Using the second order central difference with ghost node approach at the upper boundary, we get

$$\left\{ \begin{array}{l} (1 + 2\lambda \sigma_j^n) \phi_0^{n+1} - 2\lambda \sigma_j^n \phi_1^{n+1} = [1 - 2(1 - \lambda) \sigma_0^n] \phi_0^n + 2(1 - \lambda) \sigma_0^n \phi_1^n - 2\Delta z [\lambda \sigma_0^{n+1} q(t_{n+1}) + (1 - \lambda) \sigma_0^n q(t_n)], \\ \phi_M^{n+1} = \bar{\beta}(t_{n+1}). \end{array} \right. \quad (16)$$

The numerical schemes (15) and (16) represent a forward in time central in space (FTCS), backward in time central in space (BTCS), and Crank-Nicolson (CN) schemes for $\lambda = 0$, $\lambda = 1$, and $\lambda = (1/2)$, respectively [14]. The error associated with this approximation is $\mathcal{O}(\Delta z^2 + \Delta t)$ for all $\lambda \neq (1/2)$. In

the case of Crank-Nicolson, it is $\mathcal{O}(\Delta z^2 + \Delta t^2)$, second order accurate in both space and time.

We can express the above numerical procedure in a tridiagonal matrix system as

$$\begin{bmatrix} 1 + 2\lambda\sigma_0^n & -2\lambda\sigma_0^n & 0 & \dots & 0 & 0 & 0 \\ -\lambda\sigma_1^n & 1 + 2\lambda\sigma_1^n & -\lambda\sigma_1^n & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & -\lambda\sigma_{M-1}^n & 1 + 2\lambda\sigma_{M-1}^n & -\lambda\sigma_{M-1}^n \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \phi_0^{n+1} \\ \phi_1^{n+1} \\ \dots \\ \dots \\ \phi_{M-1}^{n+1} \\ \phi_M^{n+1} \end{bmatrix} = \begin{bmatrix} (1 - 2(1 - \lambda)\sigma_0^n)\phi_0^n + 2(1 - \lambda)\sigma_0^n\phi_1^n - 2\Delta z(\lambda\sigma_0^{n+1}q(t_{n+1}) + (1 - \lambda)\sigma_0^nq(t_n)) \\ (1 - \lambda)\sigma_1^n\phi_0^n + (1 - 2(1 - \lambda)\sigma_1^n)\phi_1^n + (1 - \lambda)\sigma_1^n\phi_2^n \\ \dots \\ \dots \\ \dots \\ (1 - \lambda)\sigma_{M-1}^n\phi_{M-2}^n + (1 - 2(1 - \lambda)\sigma_{M-1}^n)\phi_{M-1}^n + (1 - \lambda)\sigma_{M-1}^n\phi_M^n \\ \bar{\beta}(t_{n+1}) \end{bmatrix} \quad (17)$$

The numerical scheme (17) can be used to update the transformed variable ϕ_j^n to its value in the next time level ϕ_j^{n+1} . But we cannot advance the algorithm to the next time level ϕ_j^{n+2} without evaluating the function $c(\phi_j^{n+1})$ which requires computing the intermediate variable h_j^{n+1} . For this, we employ equation (11) which can be approximated as

$$h_j^{n+1} = h_j^n + \frac{\phi_j^{n+1} - \phi_j^n}{\bar{K}(h_j^n)} \quad (18)$$

3.2. Von Neumann Stability Analysis. Substituting Fourier component $\phi_j^n = \tilde{\phi}^n e^{ji2\pi h\xi}$ in equation (15) and rearranging the terms, we get

$$\tilde{\phi}_j^{n+1} (1 - \lambda\sigma_j^n (e^{-2i\pi h\xi} - 2 + e^{2i\pi h\xi})) = \tilde{\phi}_j^n (1 + (1 - \lambda)\sigma_j^n (e^{-2i\pi h\xi} - 2 + e^{2i\pi h\xi})) \quad (19)$$

Utilizing the relations $2h\pi\xi = u$ and $\cos u = (e^{iu} + e^{-iu})/2$, equation (19) becomes

$$\tilde{\phi}_j^{n+1} (1 + 2\lambda\sigma_j^n (1 - \cos u)) = \tilde{\phi}_j^n (1 - 2(1 - \lambda)\sigma_j^n (1 - \cos u)) \quad (20)$$

Defining the amplification factor G by

$$G(u) = \frac{(1 - 2(1 - \lambda)\sigma_j^n (1 - \cos u))}{(1 + 2\lambda\sigma_j^n (1 - \cos u))} = \frac{1 - 4(1 - \lambda)\sigma_j^n \sin^2(u/2)}{1 + 4\lambda\sigma_j^n \sin^2(u/2)} \quad (21)$$

we write equation (20) as

$$\tilde{\phi}_j^{n+1} = G(u)\tilde{\phi}_j^n \quad (22)$$

For a stable solution, the absolute value of G must be less or equal to 1 for all values of u . Since $G'(u) = 0$ at $u = 0, \pm\pi$, and $G(0) = 1$, the stability requirement can be expressed as

$$\left| \frac{1 - 4(1 - \lambda)\sigma_j^n}{1 + 4\lambda\sigma_j^n} \right| \leq 1 \quad (23)$$

Since $(1 - 4\sigma_j^n(1 - \lambda)/1 + 4\lambda\sigma_j^n)$ is always ≤ 1 , stability condition (24) reduces to

$$4\sigma_j^n (1 - 2\lambda) \leq 2 \quad (24)$$

It is easy to see that if $\lambda \geq (1/2)$, inequality (24) is always satisfied and if $\lambda < (1/2)$, inequality (24) is satisfied only if

$$\sigma_j^n \leq \frac{1}{2(1 - 2\lambda)} \quad (25)$$

Hence, we see that if $\lambda \geq (1/2)$, our scheme (15) is unconditionally stable and for $\lambda < (1/2)$ scheme is stable only if $\max \sigma_j^n \leq (1/2)$ which puts a severe restriction on the time-step size known as Courant-Friedrichs-Lewy (CFL) condition, as

$$\Delta t \leq \frac{1}{2} \left[\min c(\phi_j^n) \Delta z^2 \right] \quad (26)$$

Note that when $c(\phi) = 0$ in (12) becomes purely a diffusion equation. Thus, to avoid degeneracy, we need to adjust a small parameter in the calculation of $c(\phi_j^n)$.

$$\begin{bmatrix} \phi_0^{n+1} \\ \phi_1^{n+1} \\ \dots \\ \dots \\ \phi_{M-1}^{n+1} \\ \phi_M^{n+1} \end{bmatrix} = \begin{bmatrix} 1 - 2\sigma_0^n & 2\sigma_0^n & 0 & 0 & 0 \\ \sigma_1^n & 1 - 2\sigma_1^n & \sigma_1^n & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \sigma_{M-1}^n \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix} \begin{bmatrix} \phi_0^n \\ \phi_1^n \\ \dots \\ \dots \\ \phi_{M-1}^n \\ \phi_M^n \end{bmatrix} + \begin{bmatrix} 2\sigma_0^n \Delta z q(t_n) \\ 0 \\ \dots \\ \dots \\ 0 \\ \bar{\beta}(t_{n+1}) \end{bmatrix}, \quad (27)$$

which can be expressed in the following form:

$$\Phi^{n+1} = R(\Delta t A)\Phi^n + \Delta t F^n, \quad (28)$$

where the amplification factor $R = I - \Delta t A$ with the tri-diagonal matrix A appropriately defined from the above linear system.

The explicit scheme (26) is easy to implement as the right hand side of this equation is explicitly known at the time level t_n and there is no need to invert the matrix or need to apply any iterative approach to solve it. However, there is a serious drawback of this scheme. It is restricted by the stability criterion, also known as CFL condition (18).

As is well known, abovementioned algorithm (26) is subject to the restrictive stability condition which is equivalent to the fact that the spectral radius of matrix R is less than or equal to unity.

Super time-stepping scheme relaxes restriction of the CFL condition by requiring stability at the end of one super time-step $\Delta \tau$ consisting of a cycle of s substeps, rather than at the end of each time-step Δt , thus leading to a Runge-Kutta-like method with s stages. The inner values have, in general, no approximation properties and should only be considered as intermediate calculations.

In Runge-Kutta-Chebyshev (RKC) scheme [15, 16], the amplification factor R is defined as

$$R_s = \prod_{k=1}^s (I - \tau_k A), \quad (29)$$

and the intermediate substeps $\tau_1, \tau_2, \dots, \tau_s$ in one cycle of the super time-step are chosen as the roots of a modified Chebyshev polynomial which maximizes the duration of the super time-step $\Delta \tau = \sum_{k=1}^s \tau_k$ and ensures stability condition ($\rho(R_s) < 1$) at the end of one super time-step. Similar to the Chebyshev polynomials, Legendre polynomials are also bounded in magnitude by unity and are useful to develop a stable scheme. Here, we use a stabilized Runge-Kutta-Legendre super time-stepping scheme where the amplification factor $R_s(\Delta \tau A)$ is defined in terms of the Legendre polynomials [13, 16].

$$R_s(z) = a_s + b_s P_s(w_o + w_1 z). \quad (30)$$

3.3. Super Time-Stepping Scheme. When $\lambda = 0$, equation (17) takes the form

The parameters $a_s = 0$, $b_s = 1$, $w_o = 1$, and $w_1 = (2/s^2 + s)$ are chosen to satisfy the consistency conditions at first order $R_s(0) = 1$ and $R'_s(0) = 1$. Thus, the s -stage RK scheme takes the form

$$\Phi^{n+1} = P_s(I - w_1 \Delta \tau A)\Phi^n + \Delta \tau F^n. \quad (31)$$

Using the three-point recursion property of the Legendre polynomials,

$$(k)P_k(x) = (2k - 1)xP_{k-1}(x) - (k - 1)P_{k-2}(x). \quad (32)$$

RKL scheme (30) can be written as

$$\begin{cases} U_0 = \Phi^n, \\ U_1 = U_0 - \bar{\mu}_1 \Delta \tau A U_0 + \Delta \tau F^n, \\ U_k = \mu_k U_{k-1} + \nu_k U_{k-2} - \bar{\mu}_k \Delta \tau A U_{k-1}, \quad 2 \leq k \leq s, \\ \Phi^{n+1} = U_s, \end{cases} \quad (33)$$

where

$$\begin{aligned} \mu_k &= \frac{2k - 1}{k}, \\ \bar{\mu}_k &= \frac{2k - 1}{k} w_1, \\ \nu_k &= \frac{1 - k}{k}. \end{aligned} \quad (34)$$

The maximum stable super time-step for the above s -stage RKL scheme is given by

$$\Delta \tau_{\max} = \frac{\Delta t}{w_1} = \Delta t \left(\frac{s^2 + s}{2} \right). \quad (35)$$

Note that unlike RKC, the RKL method described by the recursive relation is consistent and stable at each of the intermediate stages which is more flexible for output.

4. Simulation Results

4.1. Numerical Setup. The numerical procedure developed in the previous section is written in Python and ran on a laptop with 2.8 GHz Quad-Core Intel Core i7 processor. We inspect

the behavior of the four numerical schemes presented in the previous section in a specific infiltration experiment. In this simulation, we consider a vertical soil column of depth $L = 70$ cm in a time period of $t_{\max} = 1$ hr.

Following Haverkamp et al. [7], we use the soil parameters and the characteristics relationship between the soil moisture content $\theta(\psi)$ and the hydraulic conductivity $K(\psi)$ as follows:

$$\theta(\psi) = \theta_r + \frac{\alpha(\theta_s - \theta_r)}{\alpha + |\psi|^\beta} = 0.075 + \frac{1.616 \times 10^6 (.287 - .075)}{1.616 \times 10^6 + |\psi|^{3.96}},$$

$$K(\psi) = K_s \frac{A}{A + |\psi|^\gamma} = \frac{34 \times 1.175 \times 10^6}{1.175 \times 10^6 + |\psi|^{4.74}}. \quad (36)$$

The simulation starts with a uniform saturation $\theta = 0.1$ cm³/cm³ and a constant water head $\psi = -61.5$ cm is maintained at the bottom boundary $z = L$. At the upper boundary $z = 0$ (the soil surface), a constant flux $q(t) = 13.69$ cm/hr for $t < 0.3$ hr and zero normal flux condition for $t > 0.3$ hr.

To compute the approximate solution using the explicit scheme FTCS, we have used a uniform spatial step size $\Delta z = 1$ cm and the time-step size $\Delta t = 0.005$ sec which guarantees CFL condition (26). This being highly resolved in temporal direction, we use it as the representative solution to compare with all other methods. The implicit schemes BTCS and Crank–Nicolson (CN), being unconditionally stable, have no restriction on the time-step size. The step size is limited only by the accuracy of approximation. By construction, the stabilized Runge–Kutta–Legendre super time-stepping (RKL) scheme is also unconditionally stable. The step size in RKL is determined by the choice of number of substeps in one super time-stepping cycle and, of course, this can be chosen by compromising the accuracy of approximation. As such, to analyze the performance of RKL compared to the implicit schemes BTCS and CN, we run BTCS and CN with $\Delta t = \sum_{k=1}^s \Delta \tau$ for different values of s . To deal with the nonlinear dependence of the functional coefficient $c(\phi)$ in the implicit schemes, we use fix point iteration with maximum allowable iteration MAXIT = 10 and relative error tolerance TOL = 10^{-6} .

4.2. Results and Discussion. Richards equation is a highly nonlinear degenerate partial differential equation. The nonlinear behavior appears from the use of constitutive relationship between θ and ψ and K and ψ .

The relative curves to the experimental data with appropriate parameter values in the moisture retention function and the hydraulic conductivity model as found in the study by Haverkamp et al. [7] are shown in Figure 1.

The numerical solutions of the Richards equation are computed using four different schemes, namely, FTCS, BTCS, CN, and RKL. First, we observed that FTCS scheme is conditionally stable, whereas BTCS, CN, and RKL schemes are unconditionally stable. Thus, to make a long simulation with the explicit scheme, we need to fix the CFL-satisfying time-step size. Since we do not have exact solution to estimate the errors, we use the numerical solution obtained from the fully explicit numerical scheme with a fine mesh ($\Delta z = 1$ cm and $\Delta t = 0.002$ sec) as the reference solution.

For the performance comparison, we make simulations with a fairly fine mesh of $\Delta z = 2$ cm and find Δt_{expl} using CFL criteria and define $\Delta \tau = \Delta t_{\text{expl}}(s^2 + s/2)$ as the time-step size of a substep in a super time-step. Note that, in general, comparison of explicit and implicit scheme is not realistic. Explicit scheme requires too many time steps, and implicit scheme needs too many iterations. Since we are using linear system solver in the implicit schemes, it is impossible to compare the efficiency using the same time-step size. To make the comparison more realistic, we have used the time-step size for implicit schemes equal to the duration of one super-step in RKL, $\Delta t = s\Delta \tau$, the duration of one super time-step size. The simulation results (speedup and the accuracy) are displayed in Tables 1–3 and Figures 2–4. The CPU time (in sec) of all the numerical schemes implemented in this work is shown in Table 1. The CPU timings of the RKL scheme for different values of s (the number of substeps in one super time-step) show the acceleration in the speed of the explicit scheme FTCS ($s = 1$). In Figure 2, the comparison of the CPU timings of the RKL for various values of s with the implicit schemes BTCS and CN is plotted. Tables 2 and 3 show comparison of errors in the final values of moisture content θ at $t = 1$ hr obtained from RKL and the other three schemes in L_∞ -norm and L_2 -norm. Table 2 also shows the stability of the explicit scheme RKL as the overall error in L_∞ -norm with respect to the reference solution which is not increasing much with the increase in the values of s . Similar behavior can also be observed from Table 3 with the errors measured in L_2 -norm.

Comparison of the solution $\theta_s^{\text{RKL}}(z, t)$ at $t = 0.3$ hr with different values of s to fully explicit $\theta^{\text{FTCS}}(z, .3)$, implicit $\theta^{\text{BTCS}}(z, .3)$, and Crank–Nicolson $\theta^{\text{CN}}(z, .3)$ methods is shown in Figure 3. These values of s are selected randomly to show that as s becomes large, there will be less accurate approximation and as $s \rightarrow 1$, the method becomes the most accurate. From Figure 3, we can see that $s = 35$ gives the best performance. It should be noted that RKL can be run with higher s by compromising with the low accuracy. Finally, we present some numerical simulation results in a very fine spatial mesh $\Delta z = 0.5$ cm obtained from the RKL method ($s = 100$) and CN ($\Delta t = s\Delta \tau$). In Figures 4 and 5, we describe the profile of moisture content $\theta(z, t)$ and pressure head $\psi(z, t)$ along the soil depth at various times $t = 0.1, 0.2, \dots, 1$ hr.

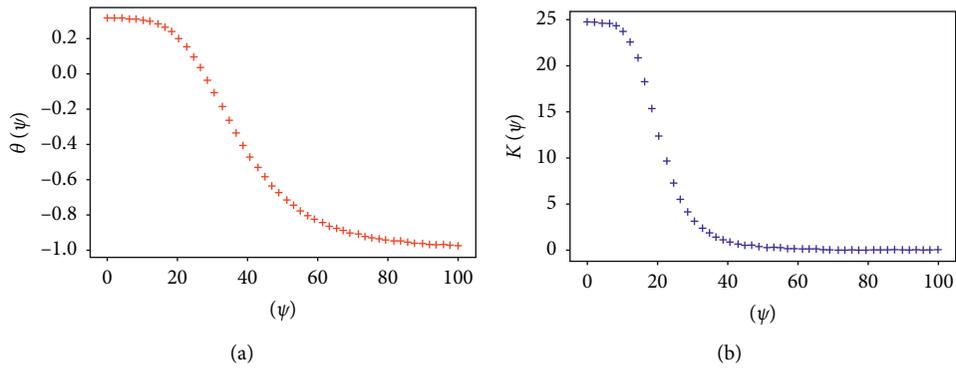


FIGURE 1: The shape of function $\theta(\psi)$ (a) and $K(\psi)$ (b).

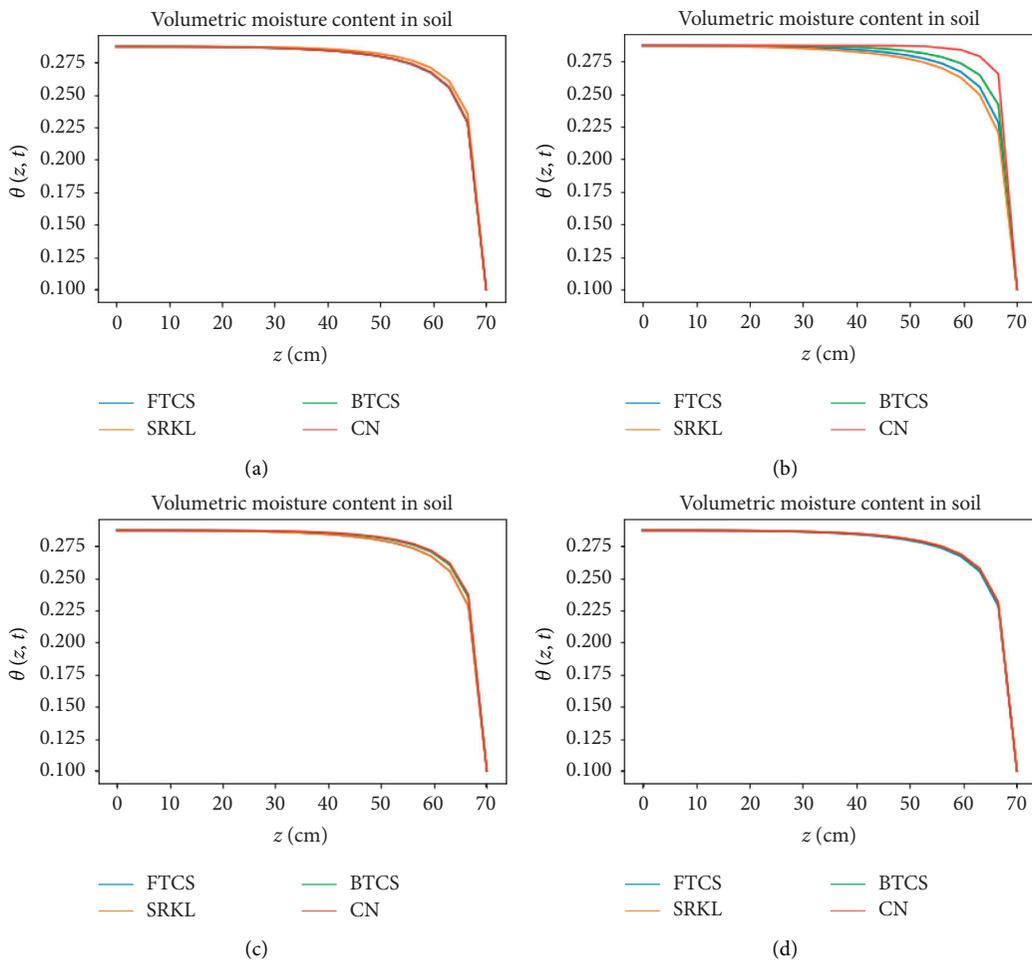


FIGURE 2: Convergence of the methods (EXPL, RKL, IMPL, and CN): $s = 20$ in (a), $s = 100$ in (b), $s = 50$ in (c), and $s = 35$ in (d).

TABLE 1: Timing of four methods.

s	EXPL	RKL	IMPL	CN
5	33.14	15.40	101.23	101.25
10	32.66	8.26	28.18	28.34
20	32.69	4.79	8.41	8.29
25	33.04	4.31	5.92	6.05
30	32.80	3.69	4.44	4.55
35	32.76	3.27	3.60	3.62
40	32.80	3.17	3.08	3.12
45	32.75	2.99	2.67	2.85
50	32.60	2.84	2.41	2.53
55	32.68	2.64	2.22	2.34
60	32.83	2.44	2.08	2.00
65	32.87	2.48	1.89	2.05
80	32.81	2.21	1.68	1.53
100	32.77	2.45	1.57	1.71

TABLE 2: Test of RKL scheme with EXPL, IMPL, and CN in L^∞ .

s	CPU time (sec)	$\ \text{RKL} - \text{EE}\ _\infty$	$\ \text{RKL} - \text{IE}\ _\infty$	$\ \text{RKL} - \text{CN}\ _\infty$
5	46.2	2.25192E-02	2.24732E-02	2.24649E-02
10	24.78	1.37265E-02	2.24732E-02	1.34624E-02
20	14.37	7.33902E-03	6.61120E-03	6.48773E-03
25	12.93	5.80247E-03	4.63486E-03	4.43717E-03
30	11.07	4.20458E-03	1.93189E-03	1.63057E-03
35	9.81	3.27050E-03	5.05055E-04	1.19139E-04
40	9.51	2.53303E-03	1.19531E-03	1.78467E-03
45	8.97	1.61003E-03	4.21368E-03	4.69720E-03
50	8.52	3.80157E-05	6.90071E-03	9.08979E-03
55	7.92	5.36361E-04	7.11451E-03	7.43036E-03
60	7.32	2.17285E-03	1.13744E-02	1.53756E-02
65	7.44	3.23959E-03	1.15905E-02	1.69259E-02
80	6.33	9.77096E-03	2.24249E-02	5.21857E-02
100	7.35	7.96584E-03	2.15554E-02	4.52289E-02

TABLE 3: Test of RKL scheme with EXPL, IMPL, and CN in L^2 .

s	CPU time (sec)	$\ \text{RKL} - \text{EE}\ _2$	$\ \text{RKL} - \text{IE}\ _2$	$\ \text{RKL} - \text{CN}\ _2$
5	46.2	3.23587E-02	3.22906E-02	3.22784E-02
10	24.78	2.02935E-02	1.99448E-02	1.98998E-02
20	14.37	1.10585E-02	9.97846E-03	9.79713E-03
25	12.93	8.77606E-03	7.04349E-03	6.75439E-03
30	11.07	6.38021E-03	2.99619E-03	2.55873E-03
35	9.81	4.97194E-03	8.79174E-04	2.55407E-04
40	9.51	3.84175E-03	1.69154E-03	2.53512E-03
45	8.97	2.43071E-03	6.18330E-03	6.84142E-03
50	8.52	4.63038E-05	1.01977E-02	1.32881E-02
55	7.92	8.80764E-04	1.05394E-02	1.08956E-02
60	7.32	3.32750E-03	1.67543E-02	2.22226E-02
65	7.44	5.05941E-03	1.72343E-02	2.45424E-02
80	6.33	1.35737E-02	3.17820E-02	7.01856E-02
100	7.35	1.27567E-02	3.22055E-02	6.30587E-02

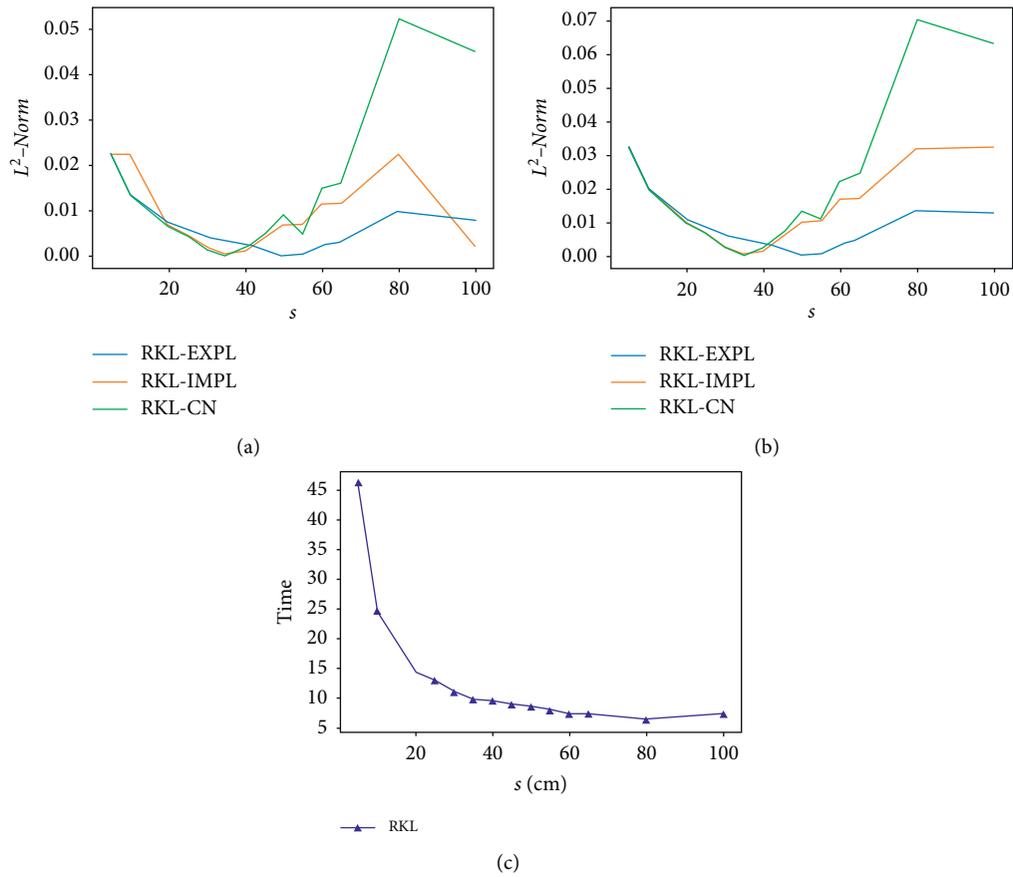


FIGURE 3: Test comparison of the four methods' error in infinity norm (a), in 2 norms (b), and speedup with RKL (c).

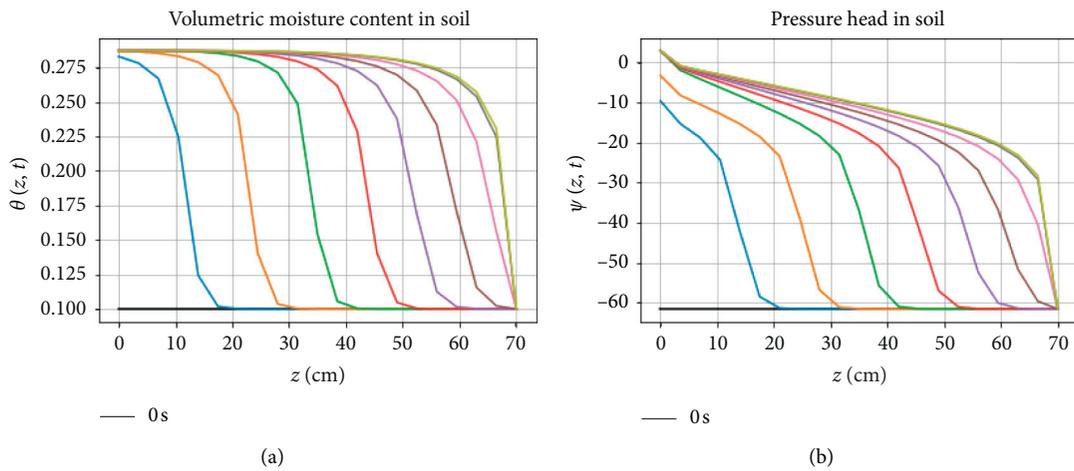


FIGURE 4: Variational trend of moisture content in depth (a) and pressure head (b).

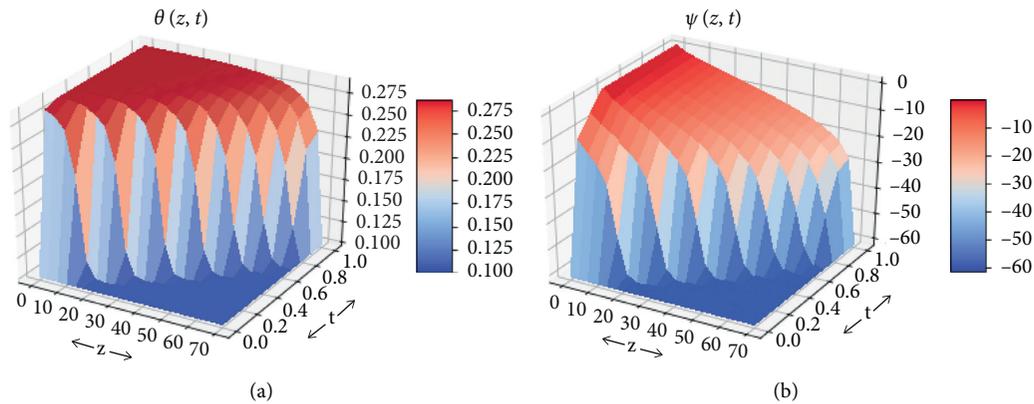


FIGURE 5: Variational trend of moisture content in depth and time (a) and pressure head (b).

5. Conclusion

In this work, we considered one-dimensional Richards equation (a highly nonlinear degenerate parabolic partial differential equation) and solved it numerically. We implemented various finite difference schemes FTCS, BTCS, CN, and a stabilized Runge–Kutta–Legendre super time-stepping scheme RKL. The numerical simulations show that the RKL scheme boasts large efficiency gains compared to the standard FTCS scheme and is comparable to CN. Explicit schemes are simple and accurate to implement and convenient for parallelization but suffer severely from stability restriction on the time-step size. In another side, implicit schemes involve iterative approach to solve the nonlinear systems, hence are moderately efficient, can be computationally very costly for highly nonlinear problems, and comparatively difficult to implement. From computational point of view, super time-stepping is more efficient than the standard implicit schemes, in that it runs at least as fast with better accuracy and it is much easier to program; also, it can be easily extended to problems in higher dimensions.

Data Availability

The data used for supporting the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

References

- [1] L. A. Richards, "Capillary conduction of liquids through porous mediums," *Physics*, vol. 1, no. 5, pp. 318–333, 1931.
- [2] M. T. Van Genuchten and E. A. Sudicky, *Recent Advances in Vadose Zone Flow and Transport Modeling, Vadose Zone Hydrology Cutting across Disciplines*, pp. 155–193, Oxford University Press, Oxford, UK, 1999.
- [3] M. A. Celia, E. T. Bouloutas, and R. L. Zarba, "A general mass-conservative numerical solution for the unsaturated flow equation," *Water Resources Research*, vol. 26, no. 7, pp. 1483–1496, 1990.
- [4] L. Berganaschi and M. Putti, "Mixed finite elements and Newton-type linearizations for the solution of the Richards equations," *International Journal for Numerical Methods in Engineering*, vol. 45, pp. 1025–1046, 1999.
- [5] R. Eymard, M. Gutnic, and D. Hilhorst, "The finite volume method for richard equation," *Computational Geosciences*, vol. 3, no. 3/4, pp. 259–294, 1999.
- [6] G. Gottardi and M. Venutelli, "Richards: computer program for the numerical simulation of one-dimensional infiltration into unsaturated soil," *Computers & Geosciences*, vol. 19, no. 9, pp. 1239–1266, 1993.
- [7] R. Haverkamp, M. Vauclin, J. Touma, P. J. Wierenga, and G. Vachaud, "A comparison of numerical simulation models for one-dimensional infiltration," *Soil Science Society of America Journal*, vol. 41, no. 2, pp. 285–294, 1997.
- [8] F. Liu, Y. Fukumoto, and X. Zhao, "A linearized finite difference scheme for the Richards equation under variable-flux boundary conditions," *Journal of Scientific Computing*, vol. 83, pp. 2–21, 2020.
- [9] C. E. Zambra, M. Dumbser, E. F. Toro, and N. O. Moraga, "A novel numerical method of high-order accuracy for flow in unsaturated porous media," *International Journal for Numerical Methods in Engineering*, vol. 89, no. 2, pp. 227–240, 2012.
- [10] D. Kavetsky, P. Binning, and S. W. Sloan, "Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation," *Advances in Water Resources*, vol. 24, pp. 595–605, 2001.
- [11] F. A. Radu and F. List, "A Study on iterative methods for solving Richards equation," *Computers and Geo-Sciences*, vol. 20, pp. 341–353, 2016.
- [12] N. Egid, E. Gioia, P. Maponi, and L. Spadoni, "A numerical solution of Richards equation: a simple method adaptable in parallel computing," *International Journal of Computer Mathematics*, vol. 97, pp. 1–18, 2018.
- [13] C. D. Meyer, D. S. Balsara, and T. D. Aslam, "A stabilized runge-kutta-legendre method for explicit super-time-stepping of parabolic and mixed equations," *Journal of Computational Physics*, vol. 257, pp. 594–626, 2014.
- [14] J. W. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, Springer, New York, NY, USA, 2010.
- [15] V. Alexiades, G. Amiez, and P.-A. Gremaud, "Super-time-stepping acceleration of explicit schemes for parabolic problems," *Communications in Numerical Methods in Engineering*, vol. 12, no. 1, pp. 31–42, 1996.

- [16] P. Gremaud and H. Khanal, "Super time-stepping schemes for conduction-radiation heat transfer problem," *Neural, Parallel and Scientific Computing*, vol. 25, pp. 457–468, 2017.