[Volume 2 issue 2 Feb 2013]
Page No.328-346 ISSN :2320-7167

INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTER RESEARCH

Accurate and economical solution of Richards' equation by the method of lines and comparison of the computational performance of ODE solvers.

M Sayful Islam[1] M Khayrul Hasan[2]

[1]Assistant Professor, Department of Mathematics, Shahjalal University of Science & Technology, Sylhet, Bangladesh, sislam_25@yahoo.com
[2]Associate Professor, Department of Mathematics, Shahjalal University of Science & Technology, Sylhet, Bangladesh, khayrulmat@gmail.com

CORRESPONDING AUTHOR:
Mohammad Sayful Islam
Assistant Professor, Department of Mathematics, Shahjalal University of Science & Technology
Sylhet-3114, Bangladesh
Phone: 008801775484990
Email: sislam_25@yahoo.com

ABSTRACT

Robust, accurate and efficient numerical simulation of groundwater flow in the unsaturated zone remains computationally expensive, particularly for problems that involved sharp fronts in both space and time. Numerical solution of these problems along with standard approaches that use of uniform spatial and temporal discretizations leads to inefficient and expensive simulations. Accurate solution of pressure head form of Richards' equation is very difficult using standard time integration techniques because in the time integration process the mass balance errors grows unless very small time steps are used. Richards' equation may be solved for many problems more economically and robustly with variable time step size instead of constant time step size. We solve Richards' equation using the method of lines with a standard finite difference technique. We show how a differential algebraic equation implementation of the method of lines can give solutions to Richards' equation that are accurate, have good mass balance properties, and are more economical for a wide range of solution accuracy. We implement the method of lines using four higher order time integration MATLAB ODE solvers **ode15s**, **ode23s**, **ode23t** and **ode23tb** to (i) assure robustness for difficult nonlinear problems and computational efficiency; (ii) develop higher order adaptive methods for the time; (iii) investigate the advantage of using higher-order methods in time; and (iv) compare the computational performance of the ODE solvers. The numerical results demonstrate that the proposed method provides a robust and efficient alternative to standard approaches for simulating variably saturated flow in one spatial dimension.

Keywords: Variably saturated flow; Finite difference; Numerical solution; Richards' equation; Method of lines

TITLE: Accurate and economical solution of Richards' equation by the method of lines and comparison of the computational performance of ODE solvers.

1. INTRODUCTION

Numerical simulation of variably saturated flow is one of the most important problems of practical interest for which main issues remain unresolved. Proper formulation of governing equation and constitutive relations are two major unresolved issues [1]. While main formulation issues remain, the standard approach to model variably saturated flow is through the use of numerical solution to Richards' equation. The movement of water in unsaturated soils described by the Richards' equation (RE) and defined by coupling a statement of flow continuity equation with Darcy's law. It is a highly nonlinear parabolic partial differential equation (PDE) which is often complicated to approximate since it does not have a closed-form analytical solution.

The highly nonlinear character of RE due to the dependence of the hydraulic conductivity and diffusivity on the moisture content, in combination with the non-trivial forcing conditions that are often encountered in engineering practice, makes RE almost impossible to solve using analytical approaches except for a few special cases [2, 3, 4]. The practical utility of analytical and semi-analytical solutions is restricted by their respective assumptions, which most notably, are homogeneity of the soil medium and a simple mathematical form for the constitutive and forcing equations. Most of the analytical solutions are obtained by using the exponential hydraulic model. The governing flow equation becomes linearized by this exponential model, which allows us to determine the possible analytical solution.

Although analytical solutions may have limited practical applications, they do serve as a means for verifying many numerical models for unsaturated flow. These are especially useful for infiltration in very dry layered soils where numerical models often suffer from lack of convergence and mass balance problems. Besides this, the analytical solutions may improve our understanding of the infiltration process under a transient state in layered soils.

The analytical solutions mentioned above are restrictive in nature and also limited to one-dimensional problems. Therefore, to solve more complicated problems, other numerical treatments are needed. In the last few decades, numerical methods have been developed for solving RE. Appropriate numerical schemes are desired, for moisture flow in unsaturated porous media. Accordingly, a huge number of numerical algorithms has been proposed to approximate Richard' equation, typically based low-order finite-difference or finite element schemes [2, 5, 6].

The numerical solution of RE need to take decisions about the form of the equation to be solved, the constitutive relations use to close the equation, the spatial approximation, the temporal approximation, the nonlinear equation solution, and the linear equation solution methods. Standard approaches have evolved for each of these decisions, together with potentially attractive alternatives to the standard choices in some cases [1].

For the numerical solution of RE, it is suitable to decouple the issues of temporal and spatial accuracy. Low-order finite difference or finite element spatial approximations and low-order time integration are most common techniques for approximating the RE [7, 8]. Additionally, most variably saturated flow simulators currently in use are based upon fixed spatial grids and either fixed time-step or an empirically based adaptive time stepping method [5, 9]. The numerical stability of the finite element models is improved by mass lumping since previous findings indicate that consistent mass formulation could cause numerical oscillations [5, 10, 11]. Previous studies emphasize the significance of proper treatment of the time derivative for reliable numerical simulations [5, 12]. Typically used time stepping schemes are the backward Euler and Crank-Nicolson schemes. Other time stepping schemes used for RE include the three-level Lees scheme, the Douglas-Jones predictor corrector method, implicit Runge-Kutta schemes and backward difference formulae [4, 13].

The solution of the non-linear algebraic systems that arise in implicit numerical discretizations of RE has been the subject of significant research studies. Iterative methods (e.g., Picard, Newton iteration methods, fast secant and relaxation methods) as well as non-iterative methods (e.g.,the implicit factored scheme) have been proposed to resolve the nonlinearities [12, 14, 15]. In fact, the fixed-point iteration scheme, the Picard method is prevalent due to its simplicity and satisfactory performance [16].

For variable step size, variable-order temporal integration, the popular approach is the method of lines (MOL) [17, 18]. It is a prescribed decoupling of the temporal and spatial approximations of a given solution to a system of partial differential algebraic equations. In the MOL approach, the temporal integration aspects of the problems can be handled by sophisticated and mature algorithms and codes can be developed to solve systems of ordinary differential or differential algebraic equations (DAE) [19, 20]. As a result, peoples are motivating with the MOL approach.

To obtain the numerical solution of problems involving flow and transport in porous media, including RE, lately the MOL has become an attractive alternative approach than traditional approach [1, 6, 20, 21, 22]. For solving RE, the MOL technique has been proven to be significantly more efficient than standard fixed time-step or fixed order empirically based adaptive approaches [6]. For the finite difference and mixed finite element spatial discretization approaches [22, 23], the issues involving the solution of the resulting system of algebraic equations [23, 24] are aspects of multidimensional and heterogeneous systems which have been investigated. With such developments, temporal integration of RE is considered relatively mature.

A DAE-based MOL solution of RE can result in a more robust and efficient than traditional approach in some cases [6]. In this approach, estimates of temporal truncation error were used explicitly to control the solution order, which ranged from first to fifth order in time, and the time-step size.

In spite of the above mentioned, for certain classes of difficult test problems, mainly those that give rise to sharp fronts that propagate through the domain have remained the major issues of robustness and efficiency [1]. These sharp fronts can require significant changes in spatial adaptation as a function of temporal evolution of the problem in both space and time. Numerical methods for RE were mainly inadequate to simple time stepping schemes coupled with finite difference or finite element spatial approximations. Fixed and heuristic time stepping schemes are the two commonly used simple time stepping strategies that are crude and uneconomical of computational resources [4] within numerical hydrological simulations. The introduction of adaptive algorithms, which adjust to the behavior of the solution and are usually more consistent and efficient than uncontrolled methods, is the fundamental advance in the numerical analysis for RE. Adaptive spatial approximations for RE include mesh refinement, moving mesh, and subspace enrichment schemes. A combination of two or three of the basic methods is the most effective approaches that are often used. Theoretically, fastest possible convergence rate may be achieved from the combined ψ and θ-based methods. However, the difficulty of data structures for some combined adaptive methods can be substantial [25]. Adaptive time integration methods included variable-order variable-step (up to fifth) DAE solver (DASPK) integrators [1, 6, 21] and lower-order adaptive backward Euler and related schemes [13, 26]. In all cases, formal truncation error control leads to significant gains in accuracy and efficiency over fixed step and heuristic time stepping algorithms, and also improves the mass balance of schemes based on the ψ formulation of RE. To solve RE, it suffices to say that the unconditional stability is an important property of an effective time stepping scheme due to the stiffness of spatially discrete parabolic partial differential equations [12]. Therefore, most invention and research codes utilize the implicit Euler algorithm, which is first order accurate but very stable.

This paper is organized as follows. In Section 2 we present the mathematical model of saturated-unsaturated groundwater flow equation, i.e. RE complemented by initial and boundary conditions. A brief discussions on the constitutive relationship. spatial discretization by the MOL technique are introduced. Implementation issues of the different MATLAB ODE solvers are also discussed including the evaluation criteria of performance measures of model. Section 3 presents two numerical examples that show the applicability of the proposed methodology to the problem at hand. In section 4 we evaluate the accuracy of the scheme. In addition to obtain accuracy and efficiency, we also discuss briefly other features of the method including root mean square error, modelling efficiency and mass balance errors. Finally, we present our concluding remarks.

2. NUMERICAL PROCEDURES

2.1 Problem formulation

RE is possible to write in a number of different forms, depending on whether pressure (ψ-based form), moisture (θ-based form), or both (mixed form) are used as state variables. Assuming the porous media and water are incompressible, the temporal variation of the water saturation is significantly larger than the temporal variation of the water pressure. We assume that the air phase is infinitely mobile, so the air pressure remains constant, and we neglect source or sink terms. With these assumptions the mixed form of RE can be written as:

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\right], \tag{2.1}$$

where ψ is the pressure head [L], θ(ψ) is the volumetric soil moisture content [L^3 L^{-3}], K(ψ) is the nonnegative hydraulic conductivity [LT^{-1}], t is the time [T], and z is the vertical coordinate assumed positive upward [L].

Taking the advantage of the differentiability of the soil retention function, one may write as follows:

$$c(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\right], \tag{2.2}$$

where $c(\psi) = \frac{d\theta}{d\psi}$ is the moisture capacity [L^{-1}].

The version is referred to as the head-form (ψ-form) of RE. Another formulation of the RE is based on the water content θ,

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left[D(\theta)\frac{\partial \psi}{\partial z}\right] + \frac{\partial K}{\partial z} \tag{2.3}$$

where $D = \frac{K}{c(\psi)} = K\frac{d\psi}{d\theta}$ is the soil water unsaturated diffusivity [L^2 T^{-1}].

Generally, ψ-based form of RE is the most commonly used because it follows for both saturated and unsaturated conditions. However, in highly nonlinear problems, such as infiltration into very dry heterogeneous soils, these approximations exhibit very poor preservation of mass balance problems, unacceptable time-step limitations [11] and relatively slow convergence [27] which seriously undermines its physical basis [26].

One of the advantages of the θ-based formulation is that perfectly mass conservative discrete approximations can be applied. However, this form degenerates under fully saturated conditions as heterogeneous material produces discontinuous θ profiles and a pressure-saturation relationship no longer exists. Hence, this form may be useful only for a homogeneous media [28].

Numerical techniques that employ both θ and ψ (the mixed formulation of RE) in the solution procedure have been developed to minimize mass-balance errors and enhance computational efficiency. It is applicable to both saturated and unsaturated porous media. This form of RE is generally considered superior to the other two forms because of robustness with respect to mass balance [5, 11, 25]. Thus, conservation of mass alone does not ensure acceptable numerical solutions, as shown by some studies [5, 25].

Expressing equation (2.1) in the pressure-head form where the primary variable is the pressure head. The new form of RE is:

$$\frac{\partial \psi}{\partial t} = \frac{1}{c(\psi) + S_s\left(\frac{\theta}{\theta_s}\right)}\frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\right] \tag{2.4}$$

or,

$$\frac{\partial \psi}{\partial t} = \frac{1}{c(\psi) + S_s S_a(\psi)}\frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z} + 1\right)\right], \tag{2.5}$$

where $S_a(\psi) = \frac{\theta(\psi)}{\theta_s}$ and S_s is the specific storage coefficient, which accounts for fluid matrix compressibility. In accounting for the effects of specific storage, the governing differential equation is an extension of the classical RE.

To complete this RE model, we need to consider auxiliary conditions of the form:

$$\psi(z, t = 0) = \psi_0(z), \tag{2.6}$$

$$\psi(z=0, t>0) = \psi_1, \quad (2.7)$$
$$\psi(z=Z, t>0) = \psi_2 \quad (2.8)$$

where Z is the length of the domain and ψ_0 (defined as initial condition) may be a function of space, but ψ_1 (bottom boundary condition) and ψ_2 (top boundary condition) are constants. These simple conditions are adequate to develop a meaningful set of test problems.

2.2 Constitutive relationship

For solving RE numerically we must specify the constitutive relations between the dependent variable (pressure head) and the nonlinear terms (moisture content, moisture capacity, and conductivity). These characteristic relations can be input to a numerical model as data in tabular form, or, more commonly as empirical expression fitted to observed data.

The van Genuchten form [29] is the most common form of these constitutive relations used in this work and the model is given by:

$$\theta(\psi) = \theta_r + \frac{\theta_s - \theta_r}{[1+|\alpha\psi|^n]^m} \quad \text{if} \quad \psi \le 0 \quad (2.9a)$$

$$\theta(\psi) = \theta_s \quad \text{if} \quad \psi > 0 \quad (2.9b)$$

$$K(\psi) = K_s \left[\frac{\theta-\theta_r}{\theta_s-\theta_r}\right]^{\frac{1}{2}} \left\{1 - \left[1 - \left(\frac{\theta-\theta_r}{\theta_s-\theta_r}\right)^{\frac{1}{m}}\right]^m\right\}^2 \quad \text{if} \quad \psi \le 0 \quad (2.10a)$$

$$K(\psi) = K_s \quad \text{if} \quad \psi > 0 \quad (2.10b)$$

$$c(\psi) = \alpha m n \frac{\theta_s - \theta_r}{[1+|\alpha\psi|^n]^{m+1}} |\alpha\psi|^{n-1} \quad \text{if} \quad \psi \le 0 \quad (2.11a)$$

$$c(\psi) = 0 \quad \text{if} \quad \psi > 0 \quad (2.11b)$$

where θ_r is the residual volumetric water content, θ_s is the porosity, α is the mean pore size, and $n = 1 - {}^1/_m$ is the uniformity of the pore-size distribution.

2.3 Method of lines

The MOL is the common method to solve the time dependent PDE. This approach is more efficient with respect to numerical accuracy and enhances computational efficiency than the regular finite difference method. It is a well recognized numerical technique or rather a semi analytical method to solve practical complex field problems. Basically, it involves discretising a given differential equation in one or two dimensions whilst using an analytical solution in the remaining direction. The MOL has the advantages in both of analytical and numerical methods. It does not yield spurious modes nor have the problem of relative convergence.

The semi analytical spirit of the formulation by the MOL guides to a straightforward and compact algorithm, which yields accurate results with less computational effort than other methods. By separating discretisation of space and time, it is easier to establish the stability and convergence for a wide range of problems. By using well documented and reliable ordinary differential equations solvers, such as MATLAB ODE solvers (**ode45**, **ode15s**, **ode113**, **ode23t**, **ode23tb**, **ode23s**, etc.), the programming effort can be substantially reduced. It is not necessary to solve a large system of equations since only a small amount of discretisation lines are necessary in the computation, hence computing time is small. In this work, we consider the simple case of applying MOL to solve parabolic PDE, namely RE by using the MATLAB ODE solvers **ode15s**, **ode23s**, **ode23t**, **ode23tb.**

The basic idea of the MOL is to replace the spatial (boundary value) derivatives in the PDE with algebraic approximations. After doing this, the spatial derivatives are no longer stated explicitly in terms of the spatial independent variables. Thus, we have a system of ordinary differential equations that approximate the original PDE using standard approaches (e.g., finite difference, finite element, or finite volume methods). The challenge, then, is to formulate the approximating system of ordinary differential equations. Once this is done, we can apply any integration algorithm for initial value ordinary differential equations to compute an approximate numerical solution to the PDE. In this approach one can specify the temporal accuracy; therefore the error checking, robustness, order selection, and time-step adaptivity features available in sophisticated ODE codes can be applied to the time integration of the PDE. However, ordinary differential equations approaches have received only limited application in the subsurface science literature, and significant implementation issues require resolution before such approaches realize their full potential in routine applications for difficult nonlinear problems, such as RE. MOL technique for RE require a formulation such as (2.2), which is a single equation in one unknown and is independent of the particular method of discretization in space. Mixed forms of RE, such as (2.1), simultaneously advance ψ and θ in time with a standard, low-order time integration method (e.g., fully implicit or Crank-Nicolson methods) so as to preserve mass balance. Equation (2.1) is formally one equation in two unknowns

and as such cannot be given to an ODE solver. Therefore, mixed methods typically use first-order schemes to advance θ and obtain mass balance [5, 30]. The MOL will permit higher-order integration when based upon a high-order ODE solver.

A MOL approach based on reduction of the original equation to a set of explicit ordinary differential equations would typically require rewrite (2.5) to the following form:

$$\frac{\partial \psi}{\partial t} = \frac{1}{c(\psi)+S_s S_a(\psi)} \frac{\partial}{\partial z}\left[K(\psi)\left(\frac{\partial \psi}{\partial z}+1\right)\right], \tag{2.12}$$

which gives an unambiguous set of ordinary differential equations after approximated the spatial derivatives term. This technique is the clear one; nevertheless, the beginning work found it to be an ineffective and comparatively costly approach for computing solutions to RE [6]. When saturated conditions are developed and fluid compressibility is small, the term $S_a(\psi)$ becomes very small, makes the approach is very challenging.

Numerical accuracy can be lost due to effect of small $S_a(\psi)$ in time integration and/or ill conditioning of the Jacobian matrix of the nonlinear system that must be solved at each time step. The time step is dependent in part on how the nonlinear solver performs during a corrector step; the latter in turn depends on the norm of the inverse Jacobian and the Lipschitz constants of the Jacobian [21]. Division by the small function $S_a(\psi)$, can influence both of these parameters, and ill-conditioned Jacobians can lead to loss of accuracy in the solution itself. Furthermore, analytic Jacobians are easier to calculate if we do not have to divide by $S_a(\psi)$ before computing the Jacobian. A DAE approach, discretizes (2.1) in space, but does not divide by $S_a(\psi)$. Popular methods based on (2.1) use this approach but are constrained to low-order methods in time [3, 4].

2.4 Spatial approximation

We consider a uniform spatial discretization comprised of *N*-1 intervals of length Δz, with Δz = Z/(*N*-1), and $z_i = (i-1)\Delta z$ for $1 \le i \le N$.

The spatial operator;

$$\mathrm{O_s}(\psi) = \frac{\partial}{\partial \mathrm{z}}\left[\mathrm{K}(\psi)\left(\frac{\partial \psi}{\partial \mathrm{z}}+1\right)\right] \tag{2.13}$$

is approximated at z = z_i for 1 < i < N by

$$\begin{aligned} O_{si}(\psi) &= \frac{\left(K\frac{\partial \psi}{\partial z}\right)_{i+1/2} - \left(K\frac{\partial \psi}{\partial z}\right)_{i-1/2}}{\Delta z} + \frac{K_{i+1/2}-K_{i-1/2}}{\Delta z} \\ &= \frac{K_{i+1/2}\frac{\psi_{i+1}-\psi_i}{\Delta z}}{\Delta z} - \frac{K_{i-1/2}\frac{\psi_i-\psi_{i-1}}{\Delta z}}{\Delta z} + \frac{K_{i+1/2}-K_{i-1/2}}{\Delta z} \\ &= \frac{1}{\Delta z^2}\left[K_{i-1/2}\psi_{i-1} - \left(K_{i-1/2}+K_{i+1/2}\right)\psi_i + K_{i-1/2}\psi_{i+1}\right] + \frac{1}{\Delta z}\left[K_{i+1/2}-K_{i-1/2}\right] \\ &= r[(K_i+K_{i-1})\psi_{i-1} - (K_{i-1}+2K_i+K_{i+1})\psi_i + (K_i+K_{i+1})\psi_{i+1}] + \frac{1}{2\Delta z}[K_{i+1}-K_{i-1}] \end{aligned} \tag{2.14}$$

where r= $r = \frac{1}{2\Delta z^2}$ and *N* is the total number of spatial nodes in the solution, ψ_i is the approximation to $\psi(z_i)$, $K_i=K(\psi_i)$ and

$$K_{i+1/2} = \frac{1}{2}[K(\psi_{i+1}) + K(\psi_i)] \tag{2.15}$$

$$K_{i-1/2} = \frac{1}{2}[K(\psi_i) + K(\psi_{i-1})] \tag{2.16}$$

The DAE system that we solve a system of *N-2* differential equations for the *N-2* unknown functions of $\psi_i(t)$, subject to the boundary conditions ψ_1 and ψ_N. The ith equation is:

$$A(\psi_i)\frac{d\psi_i}{dt} = O_{si}(\psi) \tag{2.17}$$

where O_{si} is a spatial operator given by (2.14) and

$$A(\psi_i) = C(\psi_i) + S_s S_a(\psi_i) \tag{2.18}$$

2.5 ODE solver

The above set of equations (2.17) can be solved by an implicit ODE or DAE integrator, with a stiff solver being the most reasonable choice. To solve this set of equations one can choose the packages LSODE, VODE, DASSL, DASPK (is the latest version of DASSL). All are based upon forms of the backward differentiation formulas (BDFs). LSODE and VODE are ODE solvers, while DASSL is a DAE solver. All of these solvers are appropriate to stiff systems of differential equations of the type encountered in the MOL solution of RE.

The spatial domain is discretized by the MOL approach and thus replacing the PDE with a vector system of ordinary differential equations, for which efficient and effective integrating packages have been developed [31, 32]. The MATLAB package has strong vector and matrix handling capabilities, a good set of ODE solvers, and an extensive functionality which can be used to implement the MOL [32].

We used MATLAB ODE solvers **ode15s**, **ode23s**, **ode23t**, and **ode23tb** as temporal integration. The ODE solver **ode15s** is a variable order solver based on the numerical differentiation formulas (NDFs). Solver **ode15s** uses the BDFs (also known as Gear's method) which is usually less efficient. In other words, it is a quasi-constant step size implementation of the NDFs in terms of backward differences. When **ode45** fails, or the solution is very inefficient, and/or one suspects that the problem is stiff, or when solving a differential-algebraic problem, then **ode15s** is an appropriate ODE solver [32, 33, 34, 35] but the accuracy of **ode15s** is low to medium. The solver **ode23s** is based on a modified Rosenbrock formula of order 2. Because it is a one step solver, it may be more efficient than **ode15s** at crude tolerances. It can solve some kinds of stiff problems for which **ode15s** is not effective but the order of accuracy of this solver is low. Local extrapolation is not done and by default, Jacobians are generated numerically in this solver. Again, **ode23t** is an implementation of the trapezoidal rule using a free interpolant and maintain low order of accuracy. This solver can be used if the problem is only moderately stiff and if need a solution without numerical damping. Besides, the **ode23tb** is an implementation of TR-BDF2, an implicit Runge-Kutta formula with a first stage that is a trapezoidal rule (TR) step and a second stage that is a BDF of order two and also a free interpolant is used. By construction, the same iteration matrix is used in evaluating both stages. Like **ode23s**, this solver may be more efficient than **ode15s** at crude tolerance but its order of accuracy is low. In this study, each test case was run with Relative Tolerance, $RelTol = 1.0 \times 10^{-5}$ and Absolute Tolerance, $AbsTol = 1.0 \times 10^{-7}$.

2.6 Performance measure

To assess the robustness and efficiency of the MOL approach, we use three set of spatial nodes for each of the test cases. We study different features of ODE solver, such as, the number of successful steps, failed attempts, function evaluations, partial derivatives, LU decompositions, and solutions of linear systems. Also study the characteristic of the method relative to changes in the refinement level and the number of cells in the coarse grid. Thus CPU time was considered as a one suitable choice of work measure. The number of nodes in the grid is also an important parameter for comparison in this work, since our one main objective is to obtain accurate results. The accuracy for each of the simulations is evaluated by means of the root mean square error (RMSE) and mass balance (MB) calculated with respect to the surrogate exact solution. The deviation between the fine and several sets of coarse solutions is quantified using a RMSE formula. The RMSE are measured at the three different time for both the test problems. In addition, a conventional MB approach was also used to track the numerical error. However, maintain a good closure of mass in the domain even though the internal structure of the solution was different from the fine solution. So, the calculated MB also reported for each of the runs. Adequate conservation of global mass is crucial but not enough for acceptability of a numerical simulator. In other words, accurate solutions ensure small MB error. So, we evaluated model performance by calculating differences with an exact solution of the RE. For this purpose, the numerical solution is computed using a very fine grid. Also, we calculated the modeling efficiency (ME), root square error (RSE), and relative error (RE) to ensure more accuracy and efficiency of the approach. In this study, all of the numerical codes have been written by MATLAB 7.6.0 (2008a) software and executed on a Dell INSPIRON, 2.56 GHz system.

We consider the following formula for measures of RMSE :

$$\|E\|_2 = \left[\frac{1}{N}\sum_{i=1}^{N}(\psi_e - \psi_i)^2\right]^{\frac{1}{2}} \tag{2.19}$$

We consider ψ_e is the base solution, which is made by dense-grid resolution of the vertical soil column, ψ_i is the computed solution and ***N*** is the total number of nodes.

MB measurement for determination the ability of a scheme for mass conservation can be defined as follows:

MB= (Total additional mass in the domain)/ (Total net flux into the domain),

Here the additional mass is evaluated with respect to the initial mass in the system. For the finite difference approximations, this is calculated by the following formula:

$$MB(t^n) = \frac{\sum_{i=2}^{N-1}(\theta_i^n - \theta_i^o)\Delta z}{\sum_{j=0}^{n}\left\{-K^j_{N-1/2}\left(\frac{\psi^j_{N-1}-\psi^j_N}{\Delta z}+1\right)+K^j_{3/2}\left(\frac{\psi^j_1-\psi^j_2}{\Delta z}+1\right)\right\}} \tag{2.20}$$

where $K^j_{N-1/2} = \frac{K^j_N + K^j_{N-1}}{2}$ and $K^j_{3/2} = \frac{K^j_1 + K^j_2}{2}$, n is the number of time steps, ψ^j_N is the pressure head in the jth time and Nth node and θ_i^0 and θ_i^n are the initial and final values of moisture content in node i respectively.

The following measures of analysis [36] were also used in the comparisons:
Modeling efficiency:

$$ME = 1 - \sum_{i=1}^{N} \left(\frac{\psi_i - \psi_*}{\psi_e - \psi_*}\right)^2 \quad (2.21)$$

Root square error:

$$RSE = \left[\frac{1}{N}\sum_{i=1}^{N} \left(\frac{\psi_i - \psi_*}{\psi_e}\right)\right]^{0.5} \quad (2.22)$$

and relative error:

$$RE = 100 \times \left[\frac{1}{N}\sum_{i=1}^{N} \left(\frac{\psi_i - \psi_*}{\psi_e}\right)\right] \quad (2.23)$$

where ψ_* is the mean calculated values. The desired value for ME is one, while those for RSE and RE are zero. The ME, is a global model performance measure which gives the ratio of the deviations with regard to the calculated solution. ME compares fluctuations in the computed solution with the exact solution. The RSE measure is a term by term comparison of the actual difference between the predicted value and the calculated value. This measure provides an estimate of the average relative error between the computed and the exact solutions and does not differentiate between under and over-estimation. The RE value actually provides this information. RE is positive (respectively negative), when the computed values are on average greater (respectively smaller) than the exact values.

3. NUMERICAL TESTS

To assure the purposes of this work, we sum up two test problems on which these investigations were based, examine methods for quantifying the accuracy of the resultant solutions, and draw methods to compute the computational work required to achieve the results. A set of numerical experiments was performed, to assess the robustness of the approach, to investigate methods for improving the efficiency of solutions to RE and to evaluate the advantage of using higher-order methods in time.

3.1 Description of test problems

In order to test and evaluate the solution procedure outlined above, the method is applied to two sets of one-dimensional test problems reported by [6]. We denote the numerical test problems are Problem A and Problem B. The simulation conditions are described in Table 1, including constitutive relation properties, spatial and temporal domains, and auxiliary conditions. Problem A was previously examined by [4, 6, 30] and material properties were used by [5] as well. The material properties for Problem B correspond to a dune sand as reported by [37], while the auxiliary conditions vary to yield a range of solution behavior.

The unsaturated hydraulic conductivity for both problems was described by the van Genuchten model [29]. Both sets of simulation conditions yield a difficult sharp-front problem. Problem A is considered because it is a standard test problem [6]. It is a common test problem and very helpful for illustrating some important aspects.

Problem B is a vertical infiltration problem and has been analyzed by [4, 6, 21, 38]. It has constant head boundary conditions at both top and bottom boundaries and a hydrostatic equilibrium initial condition. The combination of the initial and boundary conditions along with the constitutive relationships makes it a very difficult problem to solve accurately, since the solution includes an extremely sharp front in space that moves through the domain as a function of time.

However, Problem A is substantially less complicated than Problem B because the domain is much shorter, the media is not as uniform (i.e., *n* is smaller for Problem A), and saturated conditions are not developed for Problem A. These factors, combined, suggest that while both problems are relatively difficult sharp-front problems. Problem B is considerably more difficult than Problem A and will provide a stringent and meaningful test for the methods proposed in this work.

4. RESULTS AND DISCUSSION

4.1 Test Problem A

The typical van Genuchten soil moisture curves for the test Problem A are shown in Figure 1. A series of simulations were made and each run is tested for three sets of vertical discretizations, *N*=251, 501, and 1001 where the exact solution is made by 3001 nodes. The RMSE is evaluated at *t*=0.10, 0.20, and 0.25 days with respect to the reference solution.

The computational performance of the ODE solvers for the several vertical discretizations of the Problem A is reported in Table 2. Table 2 shows that when doubling the number of layers, the successful steps of ODE solvers increase by approximately 1.5 times for all solvers. No failed attempts are occurred by **ode23s** for 251 and 501 nodes and a small number of failed attempts are observed for 1001 nodes but required to calculate a lot of function evaluations, partial derivatives and system of linear equations. The CPU increases rapidly with the increasing number of layers. Table 2 shows that the solver **ode23s** is computationally very expensive. It is noted that **ode23t** is less expensive than other solvers to complete the simulations for all the cases. On the basis of overall assessment, it can be concluded, **ode15s** and **ode23t** are more convenient than other MATLAB ODE solvers to solve RE with the MOL technique.

The computed solution profile for four different MATLAB ODE solvers is presented in Figure 2. The solution profiles agree well with the published report [5, 6]. The solution figure indicates, there is no any difference among the ODE solvers.

The graphical representation of time stepping behavior of all ODE solvers is shown in Figures 3, 4, and 5 for all vertical discretizations. It is shown that the ODE solver **ode15s** takes larger time steps than other solvers during the simulation. After 0.13 days, Figure 5 shows that, the time step size increase smoothly for the solver **ode15s** with very small time stepping reduction and it takes bigger step size than other solvers. On the other hand, for 251 and 501 nodes, the solver **ode15s** needs reduce step size to achieve convergence. The solvers **ode23t** and **ode23tb** show approximately same behavior. But **ode23s** has faced more trouble to meet the convergence criteria than other solvers. Individual behavior of all MATLAB ODE solvers is shown in Figure 6 for all three vertical discretizations. Clearly, when the number of nodes increases, time step sizes reduce for all the nodes during simulation for all the solvers. The maximum, minimum and average time step sizes for all nodes of various solvers are summarized in Table 3. The average step size is large for **ode15s** for all the vertical discretizations are noted.

The evolution of the error is obtained by computing the exact and approximate solutions at a series of a specified output times=0.10, 0.20, and 0.25 days. The evaluated RMSE for 500 and 1000 layers is shown in Figure 7 for each distinct ODE solvers. Clearly from the Figure 7, the highest errors are shown at 0.142 m, 0.105 m and 0.0429 m at the times 0.10, 0.15, and 0.25 days respectively. Sufficiently small errors are observed in the remaining domain of the problem. Similar behavior is shown by 251 nodes for all the solvers. The RMSE graph at the three different times for the different number of layers of **ode15s** are shown in Figure 8 (left graph) including a zoom part (right graph) for clear observation of error for all layers. All other solvers show same error behavior. Evaluated RMSE for Problem A of the three sets of layers at three different time levels are presented in Table 4 for all ODE solvers. It is clear from the error table, the error reduce with increasing the number of nodes. From the statistics of Table 4 it can be concluded that all runs have adequate and comparable accuracy.

The mass balance error values for Problem A at different number of layers for various MATLAB solvers are shown in Table 5. Table 5 clearly shows, for all three discretization, the MB are acceptably small. MB is not a sufficient criterion for comparing the relative accuracy of the models. For this reason we also analysed residual errors. The results for ME, RSE and relative errors (RE) for all the solvers are summarized in Table 5. ME for all the ODE solvers are 1, this implies that, the solution is very close to the analytical solution. The RSE and RE values are well suited with evaluation criteria.

4.2 Test Problem B

Figure 9 shows the van Genuchten soil moisture retention curves for test Problem B. The first graph shows moisture content profile and second graph shows the log hydraulic conductivity profile as a function of pressure head ψ.

To solve this difficult sharp front infiltration problem and compare the accuracy of solution, we have taken three sets of vertical discretization, N=201, 401, and 801 nodes and the exact solution is made by 3201 nodes for the determination of solution accuracy.

We have presented a summary of comparison of computational statistics of ODE solvers **ode15s**, **ode23s**, **ode23t**, and **ode23tb** for this problem, such as the number of nodes, the number of successful steps taken, failed steps, the number of function evaluations, the number of Jacobians, the number of LU decompositions, the number of solutions of linear systems, and the total CPU for the various runs by the MOL approach in Table 6. Many fail attempts are occurred for all the cases. The CPU time column shows that the computational cost highly increases with increasing the number of nodes.

Figure 10 is a comparison of solution profiles for pressure head for all runs of Problem B that agrees well with the published papers [5, 6]. It is clear that all the solvers give same solution profile.

The time stepping behavior of all ODE solvers for all vertical discretizations is shown in Figures 11, 12, and 13. It is evident from these figures, all the ODE solvers perform very closely during the simulation. All solvers are forced to take very small step sizes from the beginning to end of the simulation. Constraining the time integrators to take extremely small time steps for prolonged periods during a simulation can represent an enormous computational burden for subsurface solvers. **ode15s** takes large stepping than other solvers within the simulation for all the cases. Figure 14 is the time stepping behavior of individual solver with all three vertical discretizations and it shows that the time stepping behaviors are all most same for all the solvers. For all solvers, time step size decreases along with the number of nodes increase. The maximum, minimum and average time step sizes for all the nodes of various solvers are summarized in Table 7. It is clear that large time steps are taken by **ode15s**.

The accuracy for each of the simulations is evaluated by the RMSE with respect to the exact solution. The error is evaluated for all the discretizations at three times, specifically, 0.10, 0.15, and 0.20 *days*. Figure 15 shows the error behavior of all the solvers for 401 (left graph) and 801 (right graph) nodes respectively. For these three vertical discretizations, the peak error are shown at 7 *m*, 5.73 *m* and 4.04 *m* at the times 0.10, 0.15 and 0.25 *days* respectively and low error values are found in the remaining points of problem domain. Figure 16 shows the errors of **ode15s** for all layers at the three different times and also note that similar error behavior are shown by all other schemes including a zoom (right graph) of error behavior of **ode15s** for clear observation. Detail calculated errors at the three time levels for all cases of discretizations of all the various solvers are presented in Table 8. The errors compared with the fine grid solution are shown comparable accuracy for all runs.

The MB of all discretizations by the various solvers are summarized in Table 9. All cases give sufficiently small values, which means that the model is robust and efficient. For additional accuracy and robustness of the solution evaluation, ME, RSE and relative errors (RE) are calculated for all the solvers along with all the discretizations (Table 9). ME for all the ODE solvers approach to 1, implies that the result is very close to the analytical solution. Also RSE and RE values are satisfied the evaluation criteria.

5. CONCLUSIONS

We have formulated a spatial discretization approach for a one-dimensional finite difference solution of RE and implemented successfully in a MOL framework along with MATLAB ODE solvers. It is a straightforward approach for applying sophisticated adaptive temporal integration methods to PDE models of flow in porous media. We have demonstrated that the nonlinear models of one-dimensional flow can be solved efficiently in the context of MOL using the standard time integrators. The key objectives are the numerical accuracy and the convergence of the

associated iterative schemes which are commonly used to solve the discretized problem. It is shown that higher accuracy can be achieved by using high order time discretization within the framework of the MOL. Two difficult test problems have been solved by using the MOL for a wide variety of error tolerances and spatial discretizations without difficulty. Furthermore, the convergence behavior of the MATLAB ODE solvers has been investigated. For our two test problems, tables of computational statistics show that the number of function evaluations, the number of Jacobian evaluations, and the number of steps taken were at the acceptable level for various solvers. High accuracy can be achieved with a substantial savings in computational effort, and with excellent RMSE and mass-conservation properties for all solvers. A better convergence property has been generally observed when the ODE solver **ode15s** is used. Hence, the MOL approach is an attractive alternative to solve difficult sharp-front problems that arising in the RE. Therefore the proposed approaches similar to that used in this work may provide efficient solutions for other difficult nonlinear (2D or 3D) subsurface flow and transport problems.

REFERENCES

[1] Williams, G. A. and Miller, C. T., An evaluation of temporally adaptive transformation approaches for solving Richards' equation, *Adv. Water Resources, 1999;22(8):831-840*.

[2] Arampatzis, G et.al., Estimation of unsaturated flow in layered soils with the finite control volume method, *Irrigation and Drainage, 20o1;5:349-358.*

[3] Koorevaar, P. et.al., *Elements of soil physics: Developments in Soil Science* (Elsevier Science Publishers B.V., The Netherlands, 1983).

[4] Miller, C. T. et.al., A spatially and temporally adaptive solution of Richards' equation, *Adv. Water Resources, 2005;29:525-545*.

[5] Celia, M. A. et.al., A General mass-conservative numerical solution for the unsaturated flow equation, *Water Resources Res., 1990;26(7):1483-1496.*

[6] Tocci, M. D. et. al., Accurate and economical solution of the pressure-head form of Richards' equation by the method of lines, *Adv. Water Resources, 1997;20(1):1-14*.

[7] Hanks, R. J. and Bowers, S. A., Numerical solution of the moisture flow equation for the infiltration into layered soils, *Soil. Sci. Proc., 1962:530-534*.

[8] Romano, N. et.al., Numerical analysis of one dimensional unsaturated flow in layered soils, *Adv. Water Resources, 1998;21:315-324.*

[9] Abriola, L. M. and Lang, J. R., Self-adaptive finite element solution of the one dimensional unsaturated flow equation, *Int. J. Numer. Methods Fluids, 1990;10: 227-246.*

[10] Ju, S. H. and Kung, K. J. S., Mass types, Element orders and Solution schemes for Richards' equation, *Computers and Geosciences, 1997;23(2):175-187.*

[11] Milly, P. C. D., A mass-conservative procedure for time-stepping in models of unsaturated flow, *Adv. Water Resources, 1985;8:32-36*.

[12] Kavetski, D., et.al., Noniterative time stepping schemes with adaptive truncation error control for the solution of Richards' equation, *Water Resources Res., 2002;38(10):1211-1220.*

[13] Kavetski, D., et.al., Adaptive backward Euler time stepping with truncation error control for numerical modelling of unsaturated fluid flow, *Int. J. Numer. Meth. Eng., 2001a ;53: 1301-1322*.

[14] Fassino, C. and Manzini, G., Fast-secant algorithms for the non-linear Richards Equation. *Communications in Numerical Methods in engineering, 1998;14: 921-930.*

[15] Bergamaschi, L. and Putti, M., Mixed finite elements and Newton-type linearizations for the solution of Richards' equation, *Int. J. Numer. Meth. Eng., 1999;45:1025-1046.*

[16] Lehmann, F. and Ackerer, P. H., Comparison of iterative methods for improved solutions for fluid flow equation in partially saturated porous media, *Transport in Porous Media, 1998;31:275-292.*

[17] Saucez, P. et.al., An adaptive method of lines solution of the Korteweg-de Vries equation, *Comput. Math. Appl., 1998;35(12):13–25*.

[18] Schiesser, W. E., Method of lines solution of the Korteweg-de Vries equation, *Comput. Math. Appl., 1994;28(10–12):147–54.*

[19] Brenan, K. E., et.al., *The numerical solution of initial value problems in differential–algebraic equations* (Philadelphia, PA: Soc. Ind. Appl. Math., 1996).

[20] Kees, C. E. and Miller, C. T., C++ implementations of numerical methods for solving differential–algebraic equations: design and optimization considerations, *ACM Trans Math Software, 1999;25(4): 377–403.*

[21] Miller, C. T., et.al., Robust solution of Richards' equation for non uniform porous media, *Water Resources Res., 1998;34:2599-2610*.

[22] Tocci, M. D., et.al., Inexact Newton methods and the method of lines for solving Richards' equation in two space dimensions, *Comput. Geosci., 1999;2(4); 291–309.*

[23] Farthing, M. W., et.al., Mixed finite element methods and higher-order temporal approximations, *Adv. Water Resources, 2002;25(1); 85–101.*

[24] Farthing, M. W., et.al., Mixed finite element methods and higher order temporal approximations for variably saturated groundwater flow, *Adv Water Resources, 2003;26(4);373–94.*

[25] Mansell, R. S., et.al., Adaptive Grid Refinement in Numerical Models for Water Flow and Chemical Transport in Soil: A Review, *Vadose Zone Journal, 2002;1:222-238.*

[26] Kavetski, D., et.al., Adaptive time stepping and error control in a mass conservative numerical solution of the mixed form of Richards equation, *Adv. Water Resources, 2001b;24:595-605.*

[27] Baca, R. G., et.al., Mixed transform finite element method for solving the non-linear equation for flow in variably saturated porous media, *Int. J. Numer. Meth. Fluids, 1997;24:441-455.*

[28] Guarracino, L. and Quintana, F., A third-order accurate time scheme for variably saturated groundwater flow modeling, *Communications in Numerical Methods in engineering, 2004;20:379-389.*

[29] van Genuchten, M. T., A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils, *Soil Sci. Soc. Am. J., 1980;44:892–898.*

[30] Rathfelder, K. and Abriola, L. M., Mass conservative numerical solutions of the head-based Richards equation, *Water Resources Res., 1994;30(9):2579-2586.*

[31] Schiesser, W. E., *The numerical method of lines: Integration of Partial Differential equations: ODEs, DAEs and PDEs* (Academic Press: San Diego, 1991).

[32] Shampine, L. F. and Reichelt, M. W., *The MATLAB ODE Suite Report 94-6* (Math. Dept. SMU, Dallas, 1994).

[33] Shampine, L. F. and Reichelt, M. W., The MATLAB ODE Suit, *SIAM J. Sci. Comput., 1997;18:1-22.*

[34] Shampine, L. F., et.al., Solving Index-1 DAEs in MATLAB and Simulink, *SIAM Review, 1999;41:538-552.*

[35] Shampine, L. F., *Numerical Solution of ordinary Differential Equations* (Chapman and Hall, Newyork, 1994).

[36] Loague, K. and Green, R. E., Statistical and graphical methods for evaluating solute transport models: Overview and application, *J. Contam. Hydrol., 1991;7:51–73.*

[37] Kool, J. B. and Parker, J. C., Development and evaluation of closed form expressions for hysteretic soil hydraulic properties, *Water Resources Res., 1987;23(1):105-114.*

[38] Kees, C. E. and Miller, C. T., Higher order time integration methods for two-phase flow, *Adv. Water Resources, 2002;25(2):159–77.*

TABLES

Table 1: Soil hydraulic properties used in the test problems

Variables	Problem A	Problem B
θ_r (-)	0.102	0.093
θ_s (-)	0.368	0.301
α (m^{-1})	3.35	5.47
n (-)	2.00	4.26
K_s (*m/days*)	7.97	5.40
S_s (m^{-1})	0.00	1.0×10^{-6}
z (*m*)	[0 0.3]	[0 10]
t (*days*)	[0 0.25]	[0 0.20]
$\psi_o(m)$	-10.0	-z
$\psi_1(m)$	-10.0	0.00
ψ_2 (*m*)	-0.75	0.10

Table 2: Computational performance of various ODE solvers for the Problem A

	No. of nodes	15s	23s	23t	23tb
No. of	251	2273	4208	4862	3676

Successful Steps	501 1001	3152 4124	6068 8630	7064 9418	5392 7198
No. of Failed Attempts	251 501 1001	196 271 358	0 0 46	27 61 130	33 57 109
No. of Function Evaluations	251 501 1001	19679 74055 251267	1.06042e+6 3.04614e+6 8.64735e+6	15652 42031 142346	20021 45059 129002
No. of Partial Derivatives	251 501 1001	59 134 242	4208 6068 8630	27 59 126	23 49 102
No. of LU Decompositions	251 501 1001	488 641 827	4208 6068 8676	404 448 511	355 389 459
No. of Solutions of Linear Systems	251 501 1001	4928 7054 9266	12624 18204 26028	8908 12530 16345	17957 25959 34207
CPU (s)	251 501 1001	19.95 380.44 4136.30	734.65 14753.76 172088.79	19.44 240.73 2417.97	27.54 276.60 1899.31

Table 3: Maximum, minimum and average time steps of various ODE solvers of the Problem A

No. of nodes	Time (*days*)	*15s*	*23s*	*23t*	*23tb*
251	Δt_{max} Δt_{min} Δt_{ave}	3.66290e-4 2.06693e-10 1.09987e-4	1.82940e-4 1.40818e-9 5.93965e-5	1.45430e-4 1.80676e-10 5.14086e-5	2.02130e-4 1.23093e-9 6.79902e-5
501	Δt_{max} Δt_{min} Δt_{ave}	3.66290e-4 2.06693e-10 7.92896e-5	1.82940e-4 1.40818e-9 4.11924e-5	1.45230e-4 1.80676e-10 3.53857e-5	2.02130e-4 1.23093e-9 4.63564e-5
1001	Δt_{max} Δt_{min} Δt_{ave}	3.66290e-4 2.06693e-10 6.06061e-5	1.82940e-4 1.40818e-9 2.89654e-5	1.45430e-4 1.80676e-10 2.65421e-5	2.02130e-4 1.23093e-9 3.4727e-5

Table 4: Computed RMSE for various ODE solvers for the problem A

No. of nodes	Time (*days*)	*15s*	*23s*	*23t*	*23tb*
251	0.10 0.15 0.25	0.2646 0.1942 0.1844	0.2554 0.1939 0.1849	0.2339 0.1862 0.1852	0.2559 0.2117 0.1852
501	0.10 0.15 0.25	0.1371 5.99e-2 7.58e-2	0.1283 4.46e-2 7.60e-2	0.1137 4.97e-2 7.66e-2	0.1212 4.37e-2 7.65e-2
1001	0.10 0.15 0.25	9.89e-2 1.28e-2 2.77e-2	8.41e-2 1.83e-2 2.60e-2	6.73e-2 2.00e-2 2.86e-2	6.74e-2 1.30e-2 2.85e-2

Table 5: Computed MB, ME, RSE and RE (%) for various ODE solvers of the Problem A

No. of nodes		*15s*	*23s*	*23t*	*23tb*
251	MB ME RSE RE (%)	3.68e-3 1.00 3.28e-2 -0.207	3.68e-3 1.00 3.29e-2 -0.208	3.68e-3 1.00 3.30e-2 -0.208	3.68e-3 1.00 3.30e-2 -0.207
501	MB ME RSE RE (%)	1.85e-3 1.00 1.42e-2 -6.34e-2	1.85e-3 1.00 1.42e-2 -6.36e-2	1.85e-3 1.00 1.44e-2 -6.43e-2	1.85e-3 1.00 1.44e-2 -6.42e-2
1001	MB ME RSE RE (%)	9.25e-4 1.00 5.37e-3 -1.70e-2	9.25e-4 1.00 4.80e-3 -1.52e-2	9.25e-4 1.00 5.61e-3 -1.77e-2	9.25e-4 1.00 5.58e-3 -1.76e-2

Table 6: Computational Performance of various ODE solvers for the Problem B

	No. of nodes	*15s*	*23s*	*23t*	*23tb*

No. of Successful Steps	201 401 801	3911 7350 14132	5094 9580 18730	5437 10317 19873	4203 7987 15449
No. of Failed Attempts	201 401 801	1034 1710 2979	480 744 1892	487 698 1084	600 883 1282
No. of Function Evaluations	201 401 801	96472 232389 553833	1.03006e+6 3.85288e+6 1.50257e+7	77609 189489 446070	95093 236719 541205
No. of Partial Derivatives	201 401 801	432 535 649	5094 9580 18730	330 420 506	390 511 598
No. of LU Decompositions	201 401 801	1613 2775 4975	5574 10324 20622	1486 2600 4735	1368 2336 4155
No. of Solutions of Linear Systems	201 401 801	10064 18382 34617	16722 30972 61866	11604 21480 41260	21488 40664 78910
CPU (s)	201 401 801	75.41 815.55 6580.22	742.64 12598.65 163496.71	65.02 692.80 5216.08	77.85 865.59 6253.48

Table 7: Maximum, minimum and average time steps of various ODE solvers of the Problem B

No. of nodes	Time (*days*)	*15s*	*23s*	*23t*	*23tb*
201	Δt_{max} Δt_{min} Δt_{ave}	1.41150e-4 3.29005e-12 5.11247e-5	8.77500e-5 1.52711e-11 3.92542e-5	8.67200e-5 2.87592e-12 3.67782e-5	1.08240e-4 1.33488e-11 4.75737e-5
401	Δt_{max} Δt_{min} Δt_{ave}	8.36600e-5 8.23023e-13 2.72072e-5	4.64200e-5 3.82014e-12 2.08746e-5	4.47000e-5 7.19426e-13 1.93836e-5	5.76200e-5 3.33928e-12 2.50376e-5
801	Δt_{max} Δt_{min} Δt_{ave}	4.88500e-5 2.05821e-13 1.41513e-5	2.36200e-5 9.55337e-13 1.06775e-5	2.29500e-5 1.79914e-13 1.006390e-5	2.98700e-5 8.35085e-13 1.29450e-5

Table 8: Computed RMSE for various ODE solvers for the problem B

No. of nodes	Time (*days*)	*15s*	*23s*	*23t*	*23tb*
201	0.10 0.15 0.20	0.8840 0.4055 0.3024	0.8815 0.4055 0.3040	0.8792 0.4007 0.3028	0.8808 0.3979 0.3051
401	0.10 0.15 0.20	0.5985 0.4013 0.3177	0.5988 0.4129 0.3199	0.5987 0.4000 0.3217	0.6002 0.4084 0.3210
801	0.10 0.15 0.20	0.2851 0.3247 0.2636	0.2594 0.3237 0.2646	0.2977 0.3367 0.2656	0.2906 0.3278 0.2650

Table 9: Computed MB, ME, RSE and RE (%) for various ODE solvers of the Problem B

No. of nodes		*15s*	*23s*	*23t*	*23tb*

201	MB	5.79e-2	5.79e-2	5.79e-2	5.79e-2
	ME	0.970	0.970	0.970	0.970
	RSE	1.3484	1.3480	1.3484	1.3484
	RE (%)	9.5111	9.5105	9.5105	9.5105
401	MB	2.84e-2	2.84e-2	2.84e-2	2.84e-2
	ME	0.985	0.985	0.985	0.985
	RSE	0.3874	0.3858	0.3843	0.3874
	RE (%)	1.9346	1.9264	1.9193	1.9220
801	MB	1.41e-2	1.41e-2	1.41e-2	1.41e-2
	ME	0.974	0.974	0.974	0.974
	RSE	0.2265	0.2262	0.2258	0.2260
	RE (%)	0.8001	0.7991	0.7977	0.7984

FIGURES

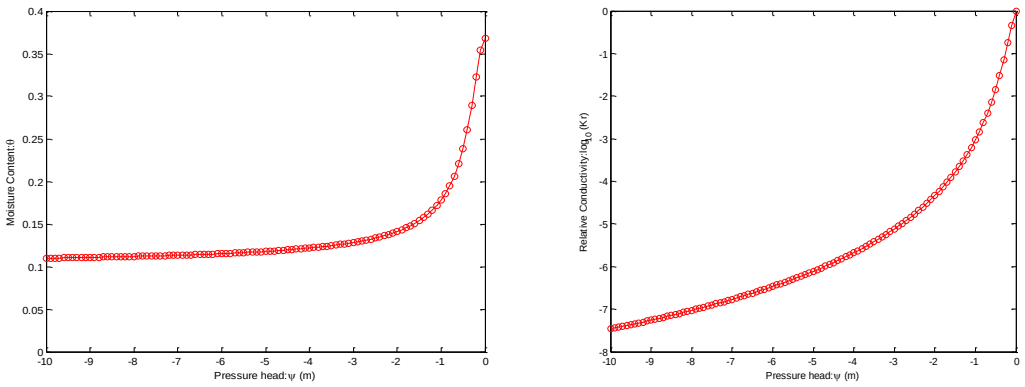

Figure 1. Soil moisture retention curves for test problem A

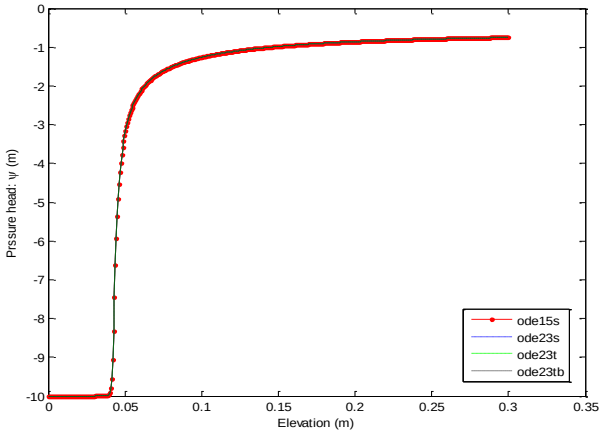

Figure 2. Pressure head profile computed by the various ODE solvers at t=0.25 *days.*

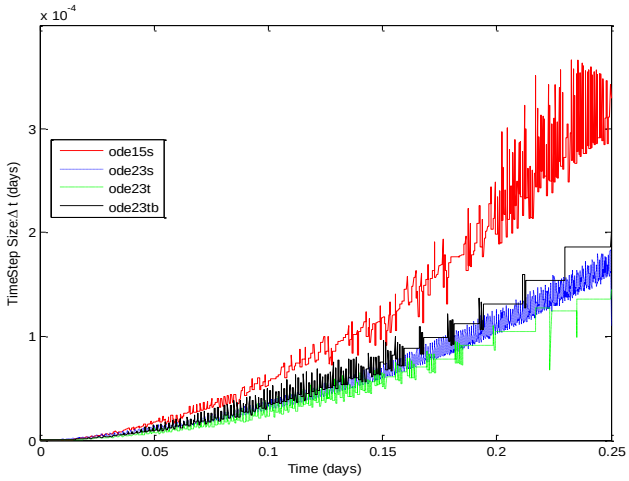

Figure 3. Time stepping behavior of various solvers for 251 nodes.

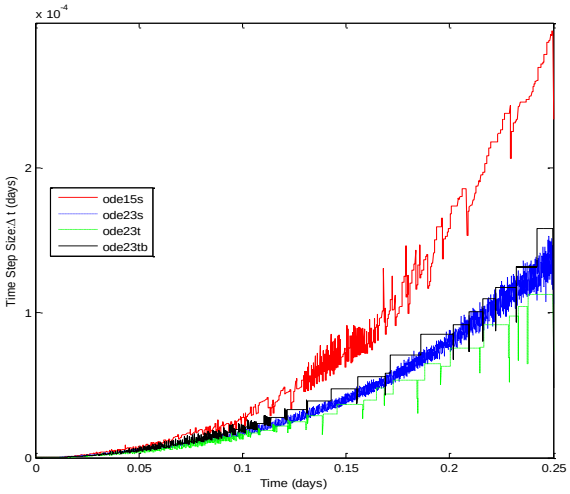

Figure 4. Time stepping behavior of various solvers for 501 nodes.

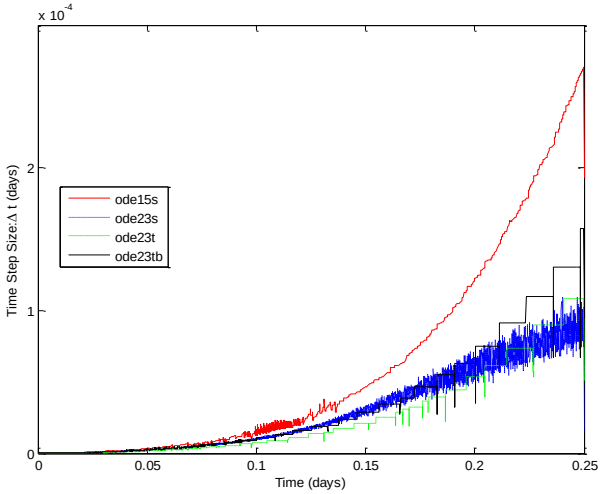

Figure 5. Time stepping behavior of various solvers for 1001 nodes.

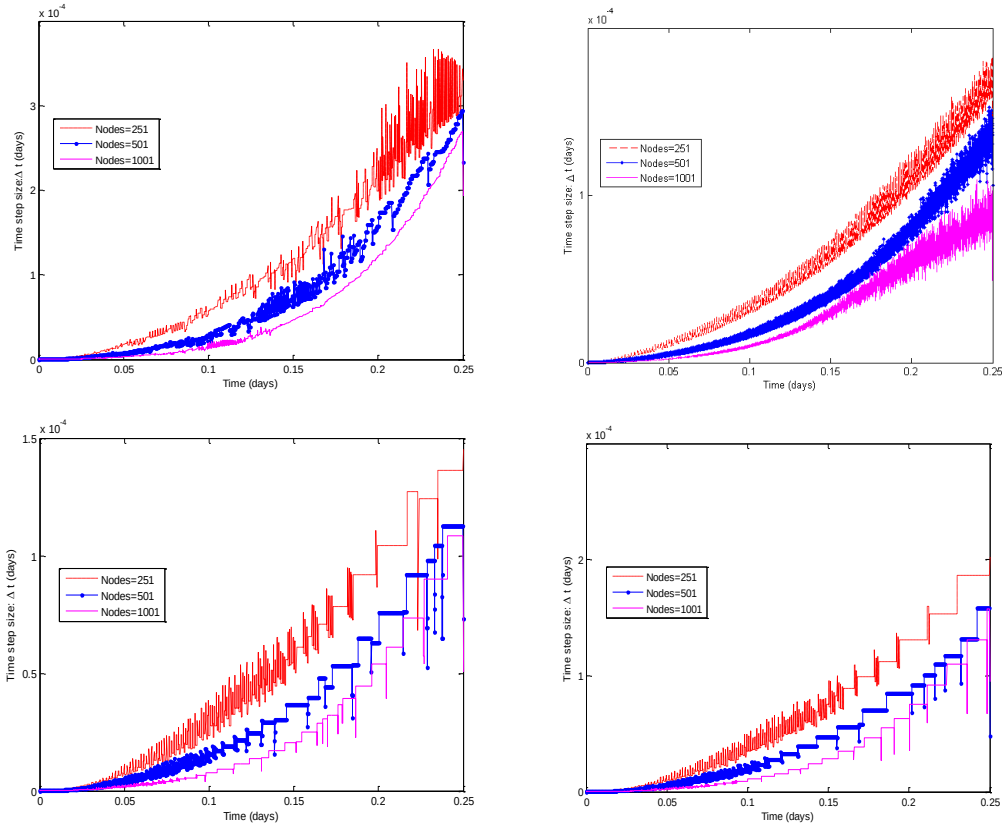

Figure 6. Time stepping behavior of **ode15s** (top left), **ode23s** (top right), **ode23t** (bottom left), and **ode23tb** (bottom right)

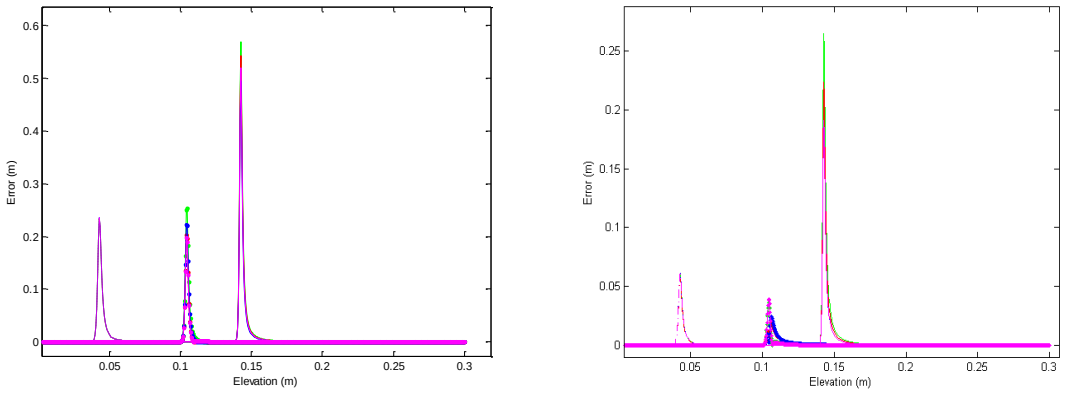

Figure 7. Computed RMSE of 501 (left) and 1001 (right) nodes for Problem A. The solid, dot-dashed, and dash-dotted lines are at times *t*=0.10, 0.20, and 0.25 *days* respectively. Green, red, blue, and magenta: **ode15s**, **ode23s**, **ode23t**, and **ode23tb** respectively.

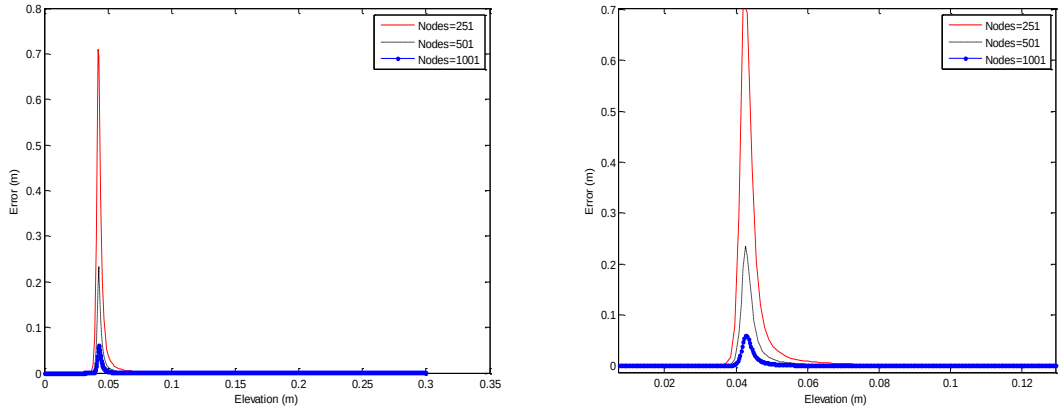

Figure 8. Computed RMSE for all nodes by **ode15s** and its zoom (right graph) portion.

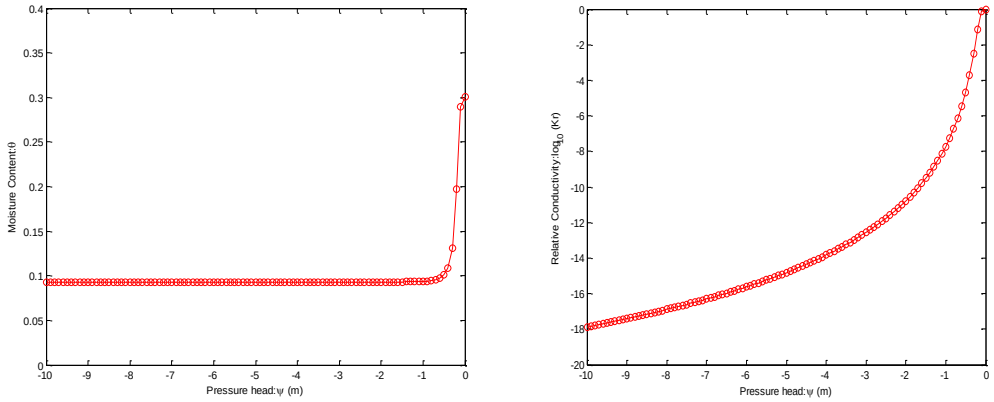

Figure 9. Soil moisture retention curves for test Problem B.

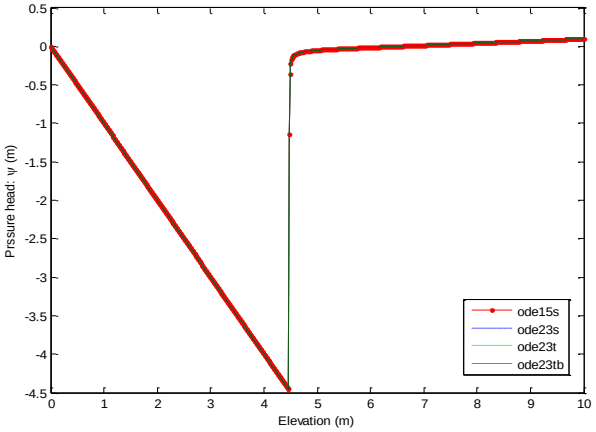

Figure 10. Pressure head profile computed by the various ODE solvers at *t*=0.20 *days*.

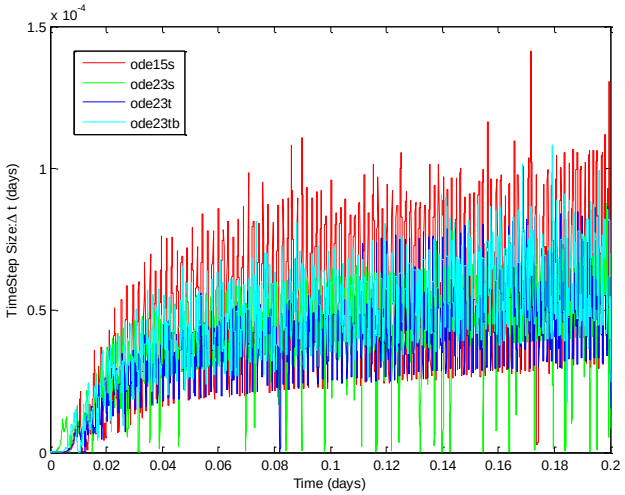

Figure 11. Time stepping behavior of various solvers for 201 nodes.

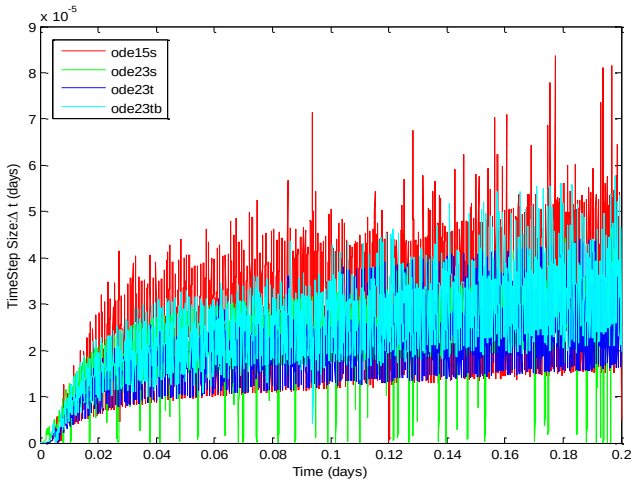

Figure 12. Time stepping behavior of various solvers for 401 nodes.

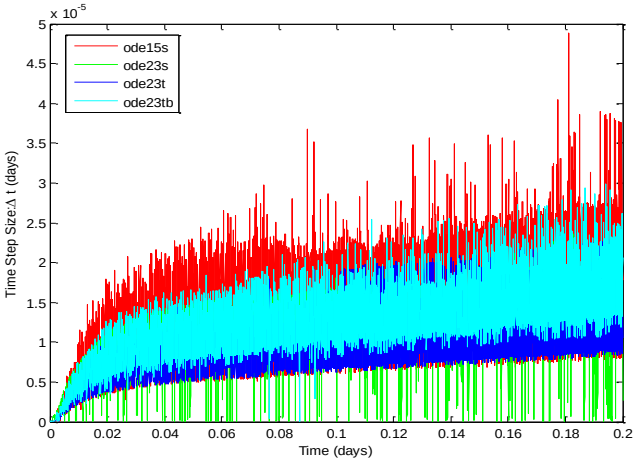

Figure 13. Time stepping behavior of various solvers for 801 nodes.

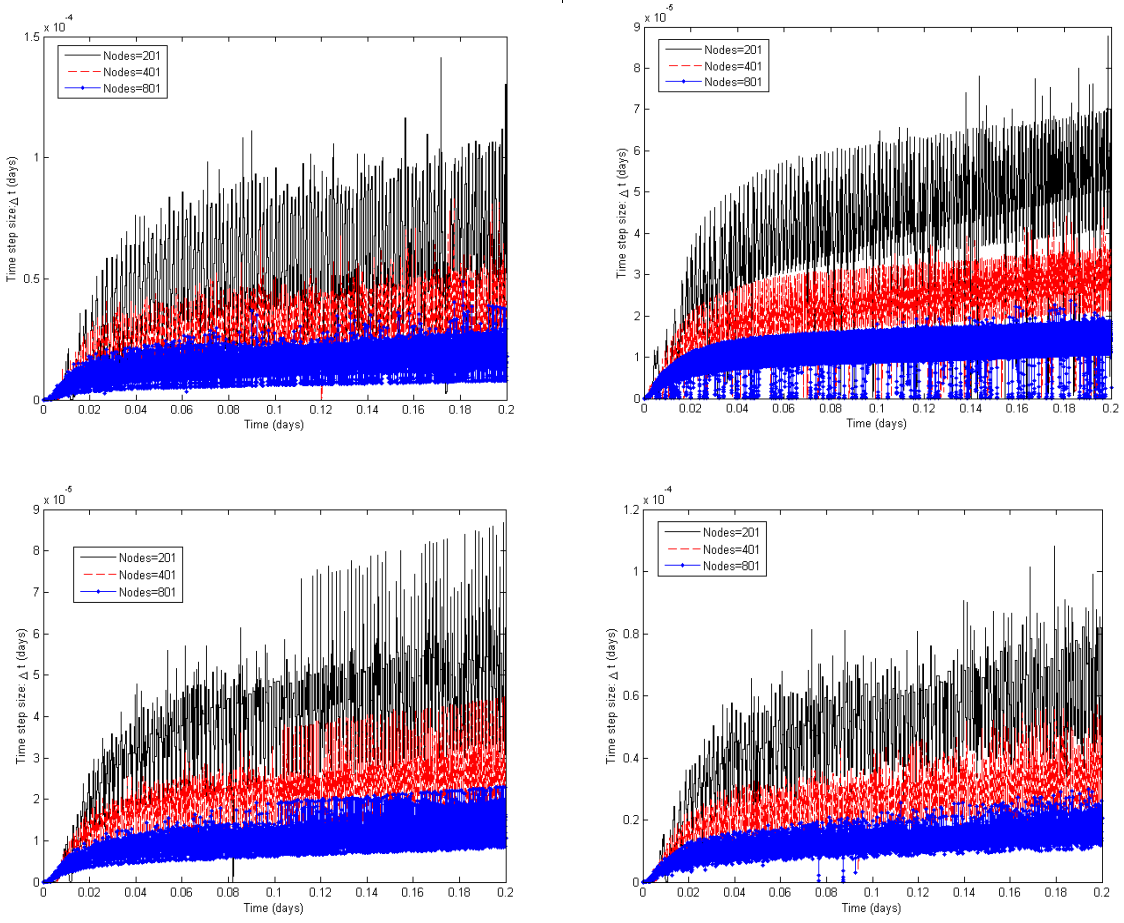

Figure 14. Time stepping behavior of **ode15s (**top left), **ode23s** (top right), **ode23t** (bottom left), and **ode23tb** (bottom right) respectively for all the nodes.

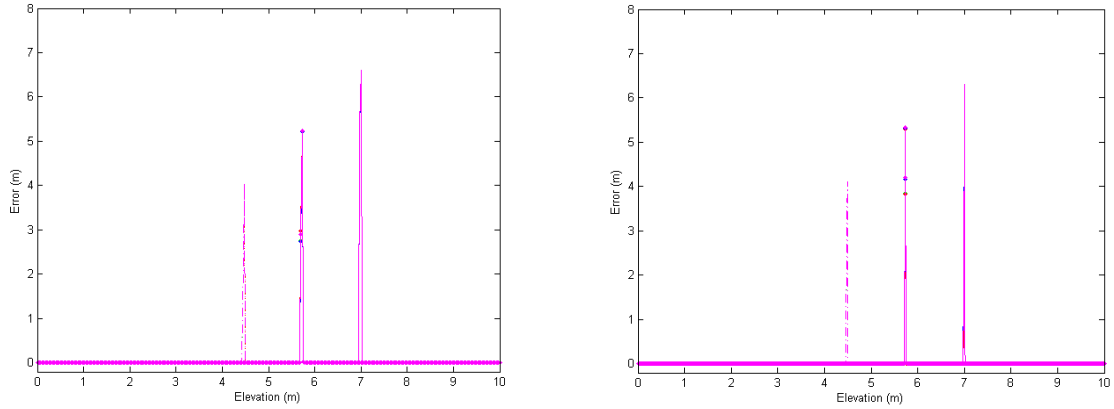

Figure 15. Computed RMSE of 401 (left) and 801 (right) nodes for Problem B. Solid, dot-dashed, and dash-dotted lines are at times *t*=0.10, 0.20, and 0.25 *days* respectively. Green, red, blue, and magenta: **ode15s**, **ode23s**, **ode23t**, and **ode23tb** respectively.

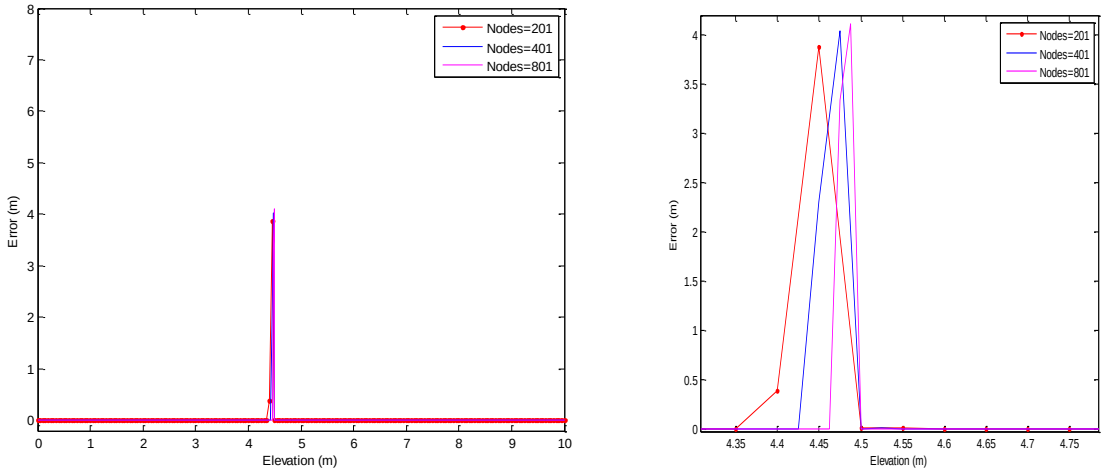

Figure 16. Computed RMSE for all nodes by **ode15s** and its zoom (right) portion.