

RESEARCH

Open Access



# A finite point algorithm for soil water-salt movement equation

Fenhong Li<sup>1</sup>, Gang Hu<sup>2</sup>, Thabet Abdeljawad<sup>3,4,5\*</sup> and Muhammad Abbas<sup>6\*</sup> 

\*Correspondence:

[tabdeljawad@psu.edu.sa](mailto:tabdeljawad@psu.edu.sa);  
[muhammad.abbas@uos.edu.pk](mailto:muhammad.abbas@uos.edu.pk)

<sup>3</sup>Department of Mathematics and General Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia

<sup>6</sup>Department of Mathematics, University of Sargodha, Sargodha 40100, Pakistan

Full list of author information is available at the end of the article

## Abstract

In this paper, we propose the meshless finite point method for solving a type of fluid flow problem. The moving least square function is combined with the collocation method to treat nonlinear one- and two-dimensional soil water-salt movement equations. An adaptive windward scheme is used to stabilize the numerical solution in regions with a large gradient change. Numerical examples with the comparison among the proposed method, finite element method and characteristic finite element method show that the meshless finite point method is more accurate and is used to eliminate the numerical oscillation phenomenon.

**MSC:** 26A33; 35R11; 65M06; 65M12; 65F40; 65F50; 65H10; 65N06

**Keywords:** Finite point method; Soil water and salt movement equation; Upwind scheme; Collocation method

## 1 Introduction

Soil salinization caused by unreasonable irrigation has become a major bottleneck restricting the sustainable development of agriculture [1]. With the increase of world population, world food production and water shortages are under greater pressure. Therefore, the rational development of inferior water resources and the improvement of the utilization of saline-alkali land and soil water have become an issue of focus which all countries are concerned about. Studying the process of soil water-salt balance migration and carrying out quantitative numerical simulations of an appropriate mathematical model provide the necessary foundation for monitoring, evaluating and controlling soil salinization in arid areas. It is particularly important to prevent the accumulation of salinity, the loss of water and soil resources, and the desertification of land. And it is also necessary to save water, make irrigation, and increase production and income. The study of soil water-salt movement [2] is derived from the study of soil water movement, which originates from Darcy's law. Since soil water movement and salt transport occur simultaneously and interact with each other, soil salt migration has arisen. As early as the 1960s, the research on the movement of water and salt has gradually developed from simple matter balance calculation to mathematics and computer simulations, which makes the concept of water and salt movement clearer and more quantitative. The numerical simulation can set the initial and boundary conditions flexibly, meanwhile it can describe the process of water and salt mi-

© The Author(s) 2021. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

gration, which is more consistent with the actual situation. Scholars such as Shi Yuanchun, Li Yunzhu and Jia Dalin have done a great deal of research on the simulation in soil salinization treatment. Zhang Weizhen [3] proposed a preliminary study on the simulation of soil water and salt transportation, therefore, the study of soil water and salt transport has entered a complete new stage. Most of the studies are aimed at simulating the infiltration characteristics of water infiltration and the transport of water [1], however, the salt migration of saline-alkali soil is less simulated. The purpose of this paper is to study the law of water and salt migration under the conditions of water infiltration, the accuracy and validity of this algorithm are verified by practical problems, which provides a scientific basis for reasonable irrigation.

The convection diffusion model is classical and popular among researchers of soil water-salt transport models. However, only a handful of partial differential equations can be obtained with exact solutions due to the heterogeneity of soil and the complexity of the conditions. Therefore, the numerical solution of partial differential equations continues to evolve. At present, the numerical methods for solving the equations of soil water and salt movement are the finite difference method (FDM) [4, 5], the finite element method (FEM) [6–8] and the characteristic finite element method (CFEM) [9, 10]. The central idea of FEM and CFEM is that a problem domain can be split into small, non-overlapping elements so that simple interpolation functions can be used to approximate any field function within each element. In addition, the Newton-Kantorovich method can also better solve the problems associated with partial differential equations [11–14]. However, shape functions for heavily distorted elements will not produce acceptable numerical solutions. One way of dealing with such distortions is to re-mesh the local domain and apply adaptive techniques, which has a high computational cost and adaptive analysis is difficult.

A meshless method based on nodes can completely or partially eliminate the grid. Therefore it overcomes the dependence on the grid and is superior to FEM and CFEM, because it not only reduces the difficulty of the simulation calculation, but also improves the accuracy of the calculation. There are many different meshless methods [15, 16]. The main difference lies in the formation of approximate function and discrete control equation. The shape function is constructed using the moving least squares method (MLS), with its control equation discretized using the collocation method [17]. Without requiring a background grid and numerical integrals, the finite point method has many advantages, including a simple discrete format, short calculation time (CPU) and easy programming. It has been successfully applied to compressible flow [18], sound propagation and other practical problems [19–22].

In this paper, a numerical solution for the convection dispersion model is calculated using the finite point method. A stabilizing term is added to the control equation before discretizing governing equations. Along with an increase in the Reynolds number,  $Re$ , the adaptive upwind support domain is used. This method is called the upwind finite point method (UFPM). This numerical method can eliminate the uncertainty in the boundary area and the case of large gradient changes.

The structure of this paper as follows: the moving least squares approximation, finite point method and its numerical algorithm are given in Sect. 2, the accuracy of the solution is examined with some examples and the actual problem in Sect. 3, and finally, conclusions are summarized in Sect. 4.

## 2 Construction of the finite point algorithm

### 2.1 Moving least squares approximation

Consider an approximation function  $u(x)$  in closed domain  $\Omega \in R^d$  ( $d = 1, 2$  or  $3$ ). Divide  $\Omega$  into subdomain  $\Omega_i$  ( $i = 1, 2, \dots, n$ ), and use  $\sum \Omega_i$  to represent the coverage of  $\Omega$ , so as to obtain the local approximation of  $u(x)$ . A point  $x_i$  and a set of points  $x_j$  ( $j = 1, 2, \dots, m$ ) surrounding  $x_i$  form a local point, thus completing  $\Omega_i$ . Assuming that  $u(x)$  is sufficiently smooth in  $\Omega_i$ , it has the following form:

$$u^h(x, \bar{x}) = \sum_{j=1}^m p_j(\bar{x})a_j(x) \equiv \mathbf{p}^T(\bar{x})\mathbf{a}(x), \tag{2.1}$$

where  $m$  is the number of terms,  $\mathbf{p}^T(x) = [p_1(x), p_2(x), \dots, p_m(x)]$  is a vector and  $\mathbf{a}^T(x) = [a_1(x), a_2(x), \dots, a_m(x)]$  are its undetermined coefficients,  $x$  is the calculating point, and  $\bar{x}$  is the spatial coordinate of the points of nearest to  $x$ .

Define a support function  $W_i(x) = W(x - x_i)$  at each  $\Omega_i$  ( $i = 1, 2, \dots, n$ ). The support domain for  $W_i(x)$  and  $x_i$  is  $\Omega_i$ . The formula  $d_s = scale \cdot d_c$  defines the support domain of the weighting function  $d_s$ , where  $scale > 1$ .

In Eq. (2.1), the weighted least squares are used to approximate the local approximation to obtain the coefficients  $a_j(x)$ . Define the following form:

$$J = \sum_{i=1}^n W(x - x_i) [u^h(x, x_i) - u(x_i)]^2 = \sum_{i=1}^n W(x - x_i) \left[ \sum_{j=1}^m p_j(x_i)a_j(x) - u_i \right]^2. \tag{2.2}$$

The coefficients  $\mathbf{a}(x)$  can be found through the extreme value of  $J$ , and the value is as follows:

$$\partial J / \partial \mathbf{a} = \mathbf{A}(x)\mathbf{a}(x) - \mathbf{B}(x)U_s = 0, \tag{2.3}$$

where

$$U_s = (u_1, u_2, \dots, u_n)^T, \tag{2.4}$$

$$\mathbf{A}(x) = \sum_{i=1}^n W_i(x)p(x_i)p^T(x_i), \tag{2.5}$$

$$\mathbf{B}(x) = [W_1(x)\mathbf{p}(x_i), W_2(x)\mathbf{p}(x_i), \dots, W_n(x)\mathbf{p}(x_i)]. \tag{2.6}$$

From Eq. (2.4), there is

$$\mathbf{a}(x) = \mathbf{A}^{-1}(x)\mathbf{B}(x)U_s. \tag{2.7}$$

Thus,  $u^h(x)$  is

$$u^h(x) = \sum_{i=1}^n \varphi_i(x)u_i = \Phi^T(x)U_s. \tag{2.8}$$

As the weight function is compact, the global approximation function of the unknown function can be obtained:

$$u^h(x) = \sum_{i=1}^N \varphi_i(x)u_i = \Phi^T(x)U,$$

$$U = (u_1, u_2, \dots, u_N)^T,$$

where the shape function is given by

$$\Phi(x) = \{\varphi_1, \varphi_2, \dots, \varphi_n\} = \mathbf{p}^T(x)\mathbf{A}^{-1}(x)\mathbf{B}(x). \tag{2.9}$$

Let  $\gamma^T = \mathbf{A}^{-1}\mathbf{p}$ , then Eq. (2.9) can be expressed as  $\Phi = \gamma^T\mathbf{B}$ . The derivatives of the shape functions are, respectively,

$$\Phi_{,i}^T = \gamma_{,i}^T\mathbf{B} + \gamma^T\mathbf{B}_{,i}. \tag{2.10}$$

$$\Phi_{,ij}^T = \gamma_{,ij}^T\mathbf{B} + \gamma_{,i}^T\mathbf{B}_{,j} + \gamma^T\mathbf{B}_{,ij}. \tag{2.11}$$

### 2.2 Collocation method

The governing equation with boundary conditions is assumed to be of the following form:

$$L[u(x)] = f(x), \quad x \in \Omega, \tag{2.12}$$

$$B[u(x)] = g(x), \quad x \in \Gamma, \tag{2.13}$$

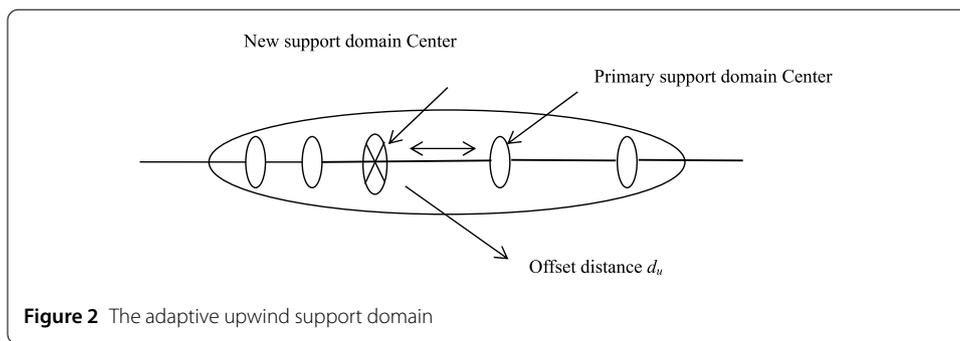
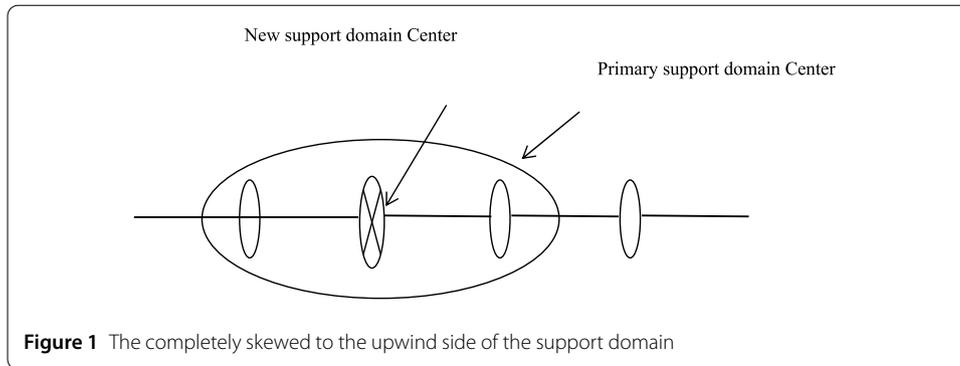
where  $L$  is a differential operator defining the governing equations in the domain  $\Omega$ , and  $B$  is the differential operator defining the boundary condition at boundary  $\Gamma$ .

The collocation method uses  $N_\Omega$  points selected in the domain  $\Omega$  and  $N_\Gamma$  points selected in the boundary  $\Gamma$  to satisfy Eqs. (2.12) and (2.13), respectively. The unknown function  $u(x)$  may be approximated at points  $x_i$  ( $i = 1, 2, \dots, N$ ) with a compact support function  $\varphi_j(x)$  as follows:

$$u(x) \approx u^h(x) = \sum_{j=1}^N u_j\varphi_j(x) = \Phi^T(x)a. \tag{2.14}$$

This is the main idea of the collocation method, put Eq. (2.14) into Eqs. (2.12) and (2.13). When  $N_\Omega + N_\Gamma = N$ , Eq. (2.14) is the determined system needed to approximate the value of each node  $x_i$  ( $i = 1, 2, \dots, N$ ). The method studied in this paper is the problem  $N_\Omega + N_\Gamma = N$ . If  $L, B$  are nonlinear operators, we can obtain the nonlinear equation group. An iterative method is needed to solve the node approximation.

There are two forms of the support field given in Figs. 1–2. One is completely skewed to the upwind side of the support domain; the other is the adaptive upwind support domain.  $d_u$  is the distance between the center of the old and new support areas, i.e., the adaptive upwind support region allows the collocation point to shift in the opposite direction of



flow with the size of  $d_u$  and  $d_u = (\coth |pe| - 1/pe) \cdot r_s$ ,  $r_s$  is the size of the support domain. A schematic diagram of the two support domains is shown below.

### 2.3 Finite point algorithm for soil water-salt transport equation

Water is the carrier of soluble salt. In addition, it also reflects the macro regulation in a saline-alkali area. Corresponding models have been established for the water movement and salt transport equations. The dynamic change in soil water content has been simulated using the finite point method. In this paper, the algorithm developed is based on the stable finite point method, in which the large gradient change making its support domain gradually tends to be windward. That is to say that we are to use the adaptive windward format to solve the stability of numerical solution. The algorithm is given now:

*Step 1.* The nodes are divided in the solution domain according to certain rules.

*Step 2.* A stability term is applied to the control equation.

*Step 3.* The moving least squares method is used to construct the shape function and the approximate expression of the unknown function is obtained.

*Step 4.* The space variables in the equation are semi-discrete and the time variable is fully discrete. The migration distance  $d_u$  of each node can then be solved, where  $d_u = (\coth |pe| - 1/pe) \cdot d_s$  and  $d_s$  are the size of the support domain. The support domain of the new node is obtained by taking the new node as the center. The value of the shape function in each node is solved in the support domain, and then put into the approximation expression for the unknown function.

*Step 5.* Generate and solve a system of algebraic equations.

### 2.3.1 Finite point algorithm for water movement equation

The Richards equation with water content as the variable is

$$\begin{cases} \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x} (D(\theta) \frac{\partial \theta}{\partial x}) + \frac{\partial}{\partial z} (D(\theta) \frac{\partial \theta}{\partial z}) + \frac{\partial K(\theta)}{\partial z} + f, & (x, z, t) \in (0, L) \times (0, L) \times (0, T), \\ \theta(x, z, 0) = \theta_0(x, z), & (x, z) \in [0, L] \times [0, T], \\ \theta(0, z, t) = \psi_1(z, t), & (z, t) \in [0, L] \times [0, T], \\ \theta(L, z, t) = \psi_2(z, t), & (z, t) \in [0, L] \times [0, T], \\ \theta(x, 0, t) = \psi_3(x, t), & (x, t) \in [0, L] \times [0, T], \\ \theta(x, L, t) = \psi_4(x, t), & (x, t) \in [0, L] \times [0, T], \end{cases} \tag{2.15}$$

where  $D(\theta)$  is the moisture diffusion rate,  $K(\theta)$  is hydraulic conductivity,  $\theta_0(x, z)$  is the initial condition,  $\psi_1(z, t), \psi_2(z, t), \psi_3(x, t), \psi_4(x, t)$  are boundary conditions.

*Step 1:* Construct the approximation function

$$\theta(x_p, z_q) = \sum_{i,j} \varphi_{pq,ij}(x_p, z_q) \cdot \theta(x_i, z_j),$$

where  $\varphi_{pq,ij}(x, z)$  is the shape function of  $(x_i, z_j)$ , obtained for the point  $(x_p, z_q)$  in the support domain.

*Step 2:* Discretize the control equation.

The governing equations in (2.15) have the following form:

$$\begin{aligned} \frac{\partial \theta}{\partial t} - \frac{\partial K(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial z} - D(\theta) \frac{\partial^2 \theta}{\partial x^2} - \frac{\partial D(\theta)}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial x}\right)^2 - D(\theta) \frac{\partial^2 \theta}{\partial z^2} \\ - \frac{\partial D(\theta)}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial z}\right)^2 - f = 0. \end{aligned} \tag{2.16}$$

Let

$$\begin{aligned} r = \frac{\partial K(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial z} + \frac{\partial K(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} + D(\theta) \frac{\partial^2 \theta}{\partial x^2} + \frac{\partial D(\theta)}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial x}\right)^2 \\ + D(\theta) \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial D(\theta)}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial z}\right)^2 + f - \frac{\partial K(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial x}, \end{aligned} \tag{2.17}$$

$$\nabla r = \left( \frac{\partial r}{\partial x} \quad \frac{\partial r}{\partial y} \right)^T. \tag{2.18}$$

A stability term is applied to Eq. (2.16):

$$\frac{\partial \theta}{\partial t} - r + \frac{h}{2|u|} \cdot u \cdot \nabla r = 0, \tag{2.19}$$

where  $u = \left(-\frac{\partial K(\theta)}{\partial \theta}, -\frac{\partial K(\theta)}{\partial \theta}\right)$  is the coefficient of convection term and  $h$  is the characteristic length. Assuming  $K(\theta) = \theta$ ,  $D(\theta)$  is a linear function of  $\theta$ . Putting Eqs. (2.17) and (2.18)

into (2.19),

$$\begin{aligned}
 & \frac{\partial \theta}{\partial t} - \frac{\partial \theta}{\partial z} - D(\theta) \frac{\partial^2 \theta}{\partial x^2} - \frac{\partial D(\theta)}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial x}\right)^2 - D(\theta) \frac{\partial^2 \theta}{\partial z^2} - \frac{\partial D(\theta)}{\partial \theta} \cdot \left(\frac{\partial \theta}{\partial z}\right)^2 - f \\
 & + \frac{h}{2|u|} \left[ -\left(\frac{\partial^2 \theta}{\partial z \partial x} + 3 \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} \cdot \frac{\partial^2 \theta}{\partial x^2}\right) + D(\theta) \frac{\partial^3 \theta}{\partial x^3} + \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} \cdot \frac{\partial^2 \theta}{\partial z^2} \right. \\
 & + D(\theta) \frac{\partial^3 \theta}{\partial z^2 \cdot \partial x} + 2 \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial z} \cdot \frac{\partial^2 \theta}{\partial z \cdot \partial x} + \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial z} \cdot \frac{\partial^2 \theta}{\partial x^2} \\
 & + D(\theta) \frac{\partial^3 \theta}{\partial x^2 \cdot \partial z} + 2 \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial x} \cdot \frac{\partial^2 \theta}{\partial x \cdot \partial z} + \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial z} \cdot \frac{\partial^2 \theta}{\partial z^2} + D(\theta) \frac{\partial^3 \theta}{\partial x^3} \\
 & \left. + 2 \frac{\partial D(\theta)}{\partial \theta} \cdot \frac{\partial \theta}{\partial z} \cdot \frac{\partial^2 \theta}{\partial z^2} + \frac{\partial f}{\partial x} + \frac{\partial f}{\partial z} \right]. \tag{2.20}
 \end{aligned}$$

To deal with the problem of two-dimensional partial differential equations, a bottom-up principle is used for rearranging the calculation points  $(x, z)$  in two dimensions. The space variables in the equation are semi-discretized. The time variable is fully discrete. The value of the shape function in each node is solved in the support domain and put into the approximation expression for the unknown function. The following algebraic equations can be obtained:

$$\begin{aligned}
 \theta_n^{m+1} = & \theta_n^m + \Delta t \cdot \sum_{h=1}^N \left[ \frac{\partial \varphi_{pq,ij}(x, z)}{\partial z} + D(\theta^m) \cdot \frac{\partial^2 \varphi_{pq,ij}(x, z)}{\partial x^2} + D(\theta^m) \cdot \frac{\partial^2 \varphi_{pq,ij}(x, z)}{\partial z^2} \right. \\
 & + \frac{h}{2|u|} \cdot \frac{\partial^2 \varphi_{pq,ij}(x, z)}{\partial z \partial x} + \frac{h}{2|u|} \cdot D(\theta^m) \cdot \frac{\partial^3 \varphi_{pq,ij}(x, z)}{\partial x^3} \\
 & + \frac{h}{2|u|} \cdot D(\theta^m) \cdot \frac{\partial^3 \varphi_{pq,ij}(x, z)}{\partial z^2 \partial x} + \frac{h}{2|u|} \cdot D(\theta^m) \cdot \frac{\partial^3 \varphi_{pq,ij}(x, z)}{\partial x^2 \partial z} \\
 & \left. + \frac{h}{2|u|} \cdot D(\theta^m) \cdot \frac{\partial^3 \varphi_{pq,ij}(x, z)}{\partial z^3} \right] \Big|_{x=x_p, z=z_p} \cdot \theta_{ij} \\
 & + \Delta t \cdot \left[ \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{i=1}^N \frac{\partial \varphi_{pq,ij}}{\partial x} \cdot \theta_{ij}^m\right)^2 + \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial z} \cdot \theta_{ij}^m\right)^2 \right. \\
 & + \frac{3h}{2|u|} \cdot \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial x} \cdot \theta_{ij}^m\right) \cdot \left(\sum_{h=1}^N \frac{\partial^2 \varphi_{pq,ij}}{\partial x^2} \cdot \theta_{ij}^m\right) \\
 & + \frac{h}{2|u|} \cdot \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial x} \cdot \theta_{ij}^m\right) \cdot \left(\sum_{h=1}^N \frac{\partial^2 \varphi_{pq,ij}}{\partial z^2} \cdot \theta_{ij}^m\right) \\
 & + \frac{h}{|u|} \cdot \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial z} \cdot \theta_{ij}^m\right) \cdot \left(\sum_{h=1}^N \frac{\partial^2 \varphi_{pq,ij}}{\partial z \partial x} \cdot \theta_{ij}^m\right) \\
 & + \frac{h}{2|u|} \cdot \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial z} \cdot \theta_{ij}^m\right) \cdot \left(\sum_{h=1}^N \frac{\partial^2 \varphi_{pq,ij}}{\partial x^2} \cdot \theta_{ij}^m\right) \\
 & \left. + \frac{h}{|u|} \cdot \frac{\partial D(\theta^m)}{\partial \theta} \cdot \left(\sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial x} \cdot \theta_{ij}^m\right) \cdot \left(\sum_{h=1}^N \frac{\partial^2 \varphi_{pq,ij}}{\partial x \partial z} \cdot \theta_{ij}^m\right) \right]
 \end{aligned}$$

$$\begin{aligned}
 & + \frac{3h}{2|u|} \cdot \frac{\partial D(\theta^m)}{\partial \theta} \left( \sum_{h=1}^N \frac{\partial \varphi_{pq,ij}}{\partial z} \cdot \theta_{ij}^m \right) \cdot \left( \sum_{h=1}^N \frac{\partial^2 \varphi_{pq,ij}}{\partial z^2} \cdot \theta_{ij}^m \right) \\
 & + f + \frac{h}{2|u|} \cdot \left( \frac{\partial f}{\partial x} + \frac{\partial f}{\partial z} \right) \Bigg]_{x=x_p, z=z_q} \quad (2.21)
 \end{aligned}$$

There are two ways to solve the equations. One method is the forward iteration method when the right side of the equation is all about the value of the  $m$  layer of  $\theta$ . Another method is the predictor corrector method when the equation gets this form:  $\theta^{m+1} = A^{m+1}\theta^m + F^{m+1}$ .

### 2.3.2 Finite point algorithm for the soil salt transport equation

The convection dispersion equation for non-volatile, non-reactive, non-adsorptive soil water and salt transport can be written as

$$\frac{\partial}{\partial t}(\theta c) = \frac{\partial}{\partial z} \left( D_{1h} \frac{\partial c}{\partial z} \right) - \frac{\partial}{\partial z} (J_w c), \quad (2.22)$$

where the hydrodynamic dispersion coefficient  $D_{1h}$  is formed by the superposition of the molecular diffusion coefficient and mechanical dispersion coefficient, and  $J_w$  is the soil moisture flux. In the case of saturated flow, the water content  $\theta$  is constant, and Eq. (2.22) gets the following form:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} \left( \frac{D_{1h}}{\theta} \frac{\partial c}{\partial x} \right) - \frac{\partial}{\partial x} \left( \frac{J_w}{\theta} c \right). \quad (2.23)$$

For a steady flow condition, Eq. (2.23) gets the following form:

$$\frac{\partial c}{\partial t} = \frac{\partial}{\partial x} \left( D_v \frac{\partial c}{\partial x} \right) - v \frac{\partial c}{\partial x}, \quad (2.24)$$

where  $D_v \triangleq D_L/\theta$  is the convection dispersion coefficient, and  $v \triangleq J_w/\theta$  is the mean pore water velocity. The finite point algorithm is shown for the migration equation of non-volatile, non-reactive, non-adsorptive soil water and salt:

$$\begin{cases} \frac{\partial c}{\partial t} = \frac{\partial}{\partial x} (D_v \frac{\partial c}{\partial x}) - v \frac{\partial c}{\partial x}, & (x, t) \in (0, L) \times (0, T), \\ c(x, 0) = c_0(x), & x \in [0, L], \\ c(0, t) = \psi_1(t), & t \in [0, T], \\ c(L, t) = \psi_2(t), & t \in [0, T]. \end{cases} \quad (2.25)$$

where  $c$  is the salt concentration,  $c_0(x)$  is the initial value,  $\psi_1(t)$ ,  $\psi_2(t)$  are boundary values.

*Step 1:* Construct the approximation function

$$c^h(x_p) = \sum_{i=1}^n \varphi_{ip}(x_p) c(x_i), \quad (2.26)$$

where  $\varphi_{ip}(x)$  is the shape function of the point  $x_i$ , obtained for  $x_p$  in the support domain,  $\varphi_{ip}(x_p)$  is the value of  $\varphi_{ip}(x)$  at point  $x_p$ , and  $n$  is the number of nodes in the local support domain for the shape function at point  $x_p$ .

Step 2: Discretize the control equation

$$r = -v \frac{\partial c}{\partial x} + \frac{\partial}{\partial x} \left( D_v \frac{\partial c}{\partial x} \right). \tag{2.27}$$

A stability term is applied to governing Eq. (2.25) as follows:

$$L(x, t) = \frac{\partial c}{\partial t} - r + \frac{h}{2|v|} v \cdot \frac{\partial r}{\partial x} = 0. \tag{2.28}$$

The space variables in the equation are semi-discrete and the interval  $I = [0, L]$  has the number of  $N$  nodes with the  $x = x_i$  ( $i = 1, 2, \dots, N$ ) distribution. Discretization at the node  $x_p$  is as follows:

$$\begin{aligned} L(c|_{x=x_p}) = & \frac{\partial c}{\partial t} \Big|_{x=x_p} + v \frac{\partial c}{\partial x} \Big|_{x=x_p} - \frac{\partial D_v}{\partial x} \frac{\partial c}{\partial x} \Big|_{x=x_p} - D_v \frac{\partial^2 c}{\partial x^2} \Big|_{x=x_p} \\ & + \frac{h}{2|v|} \cdot v \cdot \left[ -\frac{\partial v}{\partial x} \frac{\partial c}{\partial x} \Big|_{x=x_p} - v \frac{\partial^2 c}{\partial x^2} \Big|_{x=x_p} + \frac{\partial^2 D_v}{\partial x^2} \frac{\partial c}{\partial x} \Big|_{x=x_p} \right. \\ & \left. + \frac{\partial D_v}{\partial x} \frac{\partial^2 c}{\partial x^2} \Big|_{x=x_p} + \frac{\partial D_v}{\partial x} \frac{\partial^2 c}{\partial x^2} \Big|_{x=x_p} + D_v \frac{\partial^3 c}{\partial x^3} \Big|_{x=x_p} \right] = 0. \end{aligned} \tag{2.29}$$

We can remove the term  $\frac{\partial c}{\partial t} \Big|_{x=x_p}$  at the time layer  $m\Delta t$  in Eq. (2.29) by using the forward difference method,

$$\begin{aligned} L(c|_{t=m\Delta t, x=x_p}) = & \frac{c|_{t=(m+1)\Delta t, x=x_p} - c|_{t=m\Delta t, x=x_p}}{\Delta t} + v \frac{\partial c}{\partial x} \Big|_{t=m\Delta t, x=x_p} - \frac{\partial D_v}{\partial x} \frac{\partial c}{\partial x} \Big|_{t=m\Delta t, x=x_p} \\ & - D_v \frac{\partial^2 c}{\partial x^2} \Big|_{t=m\Delta t, x=x_p} + \frac{h}{2|v|} \cdot v \cdot \left[ -\frac{\partial v}{\partial x} \frac{\partial c}{\partial x} \Big|_{t=m\Delta t, x=x_p} - v \frac{\partial^2 c}{\partial x^2} \Big|_{t=m\Delta t, x=x_p} \right. \\ & \left. + \frac{\partial^2 D_v}{\partial x^2} \frac{\partial c}{\partial x} \Big|_{t=m\Delta t, x=x_p} + \frac{\partial D_v}{\partial x} \frac{\partial^2 c}{\partial x^2} \Big|_{t=m\Delta t, x=x_p} \right. \\ & \left. + \frac{\partial D_v}{\partial x} \frac{\partial^2 c}{\partial x^2} \Big|_{t=m\Delta t, x=x_p} + D_v \frac{\partial^3 c}{\partial x^3} \Big|_{t=m\Delta t, x=x_p} \right] = 0. \end{aligned} \tag{2.30}$$

Putting Eq. (2.26) into Eq. (2.30) gives the equation after discretization,

$$\begin{aligned} c_p^{m+1} = & c_p^m - \Delta t \sum_{i=1}^N \left[ \left( v_p^m - \frac{\partial D_v}{\partial x} \Big|_{t=m\Delta t, x=x_p} - \frac{h}{2} \frac{\partial v}{\partial x} \Big|_{t=m\Delta t, x=x_p} \right. \right. \\ & \left. \left. + \frac{h}{2} \frac{\partial^2 D_v}{\partial x^2} \Big|_{t=m\Delta t, x=x_p} \right) \frac{d\varphi_{pi}}{dx} \Big|_{x=x_p} \right. \\ & \left. + \left( -D_{vp}^m - \frac{h}{2} v_p^m + \frac{h}{2} \frac{\partial D_v}{\partial x} \Big|_{t=m\Delta t, x=x_p} + \frac{h}{2} \frac{\partial D_v}{\partial x} \Big|_{t=m\Delta t, x=x_p} \right) \frac{d^2 \varphi_{pi}}{dx^2} \Big|_{x=x_p} \right. \\ & \left. + \frac{h}{2} D_{vp}^m \frac{d^3 \varphi_{pi}}{dx^3} \Big|_{x=x_p} \right] c_i^m. \end{aligned} \tag{2.31}$$

Let

$$\begin{aligned} \alpha_{pi} = & -\Delta t \cdot \left\{ \left( v_p^m - \frac{\partial D_v}{\partial x} \Big|_{t=m\Delta t, x=x_p} - \frac{h}{2} \frac{\partial v}{\partial x} \Big|_{t=m\Delta t, x=x_p} + \frac{h}{2} \frac{\partial^2 D_v}{\partial x^2} \Big|_{t=m\Delta t, x=x_p} \right) \frac{d\varphi_{pi}}{dx} \Big|_{x=x_p} \right. \\ & + \left( -D_{vp}^m - \frac{h}{2} v_p^m + \frac{h}{2} \frac{\partial D_v}{\partial x} \Big|_{t=m\Delta t, x=x_p} + \frac{h}{2} \frac{\partial D_v}{\partial x} \Big|_{t=m\Delta t, x=x_p} \right) \frac{d^2\varphi_{pi}}{dx^2} \Big|_{x=x_p} \\ & \left. + \frac{h}{2} D_{vp}^m \frac{d^3\varphi_{pi}}{dx^3} \Big|_{x=x_p} \right\}. \end{aligned} \tag{2.32}$$

Equation (2.31) can be written as

$$C^{m+1} = AC^m, \tag{2.33}$$

where

$$C^m = (c_1(t_m), c_2(t_m), \dots, c_N(t_m))^T,$$

$$A = \begin{bmatrix} \alpha_{11} + 1 & \alpha_{12} & \cdots & \alpha_{1N} \\ \alpha_{21} & \alpha_{22} + 1 & \cdots & \alpha_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_{N1} & \alpha_{N2} & \cdots & \alpha_{NN} + 1 \end{bmatrix}.$$

### 3 Numerical simulation

#### 3.1 Linear soil salt transport equation

*Example 1* The one-dimensional steady-state salt transport is

$$\begin{cases} v \frac{\partial c}{\partial z} - \frac{\partial}{\partial z} (D_v \frac{\partial c}{\partial z}) = f, & (x, t) \in (0, 1) \times (0, T), \\ c(0) = c(1) = 0, & t \in [0, T]. \end{cases} \tag{3.1}$$

Let  $v = 1, D_v = 0.01$ . Then the exact solution of the equation is  $c(z) = z - (1 - e^{100z})/(1 - e^{100})$ . The right-hand term can be obtained by putting the exact solution into the control equation. In this paper, we discuss the error of  $L_2$ -norm:  $\|c - c_h\|_{L_2} = (\int_{\Omega} (c - c_h)^2 d\Omega)^{1/2}$ , and choose the Gauss function as the weight function. The Reynolds number is  $Re = vd_c/2D_v$ , where  $d_c$  is the distance between nodes.

Table 1 shows that the finite point computation error with the stability term only, the finite point error of the adaptive upwind scheme and the error of finite element method are given, respectively, in the case of convection dominance.

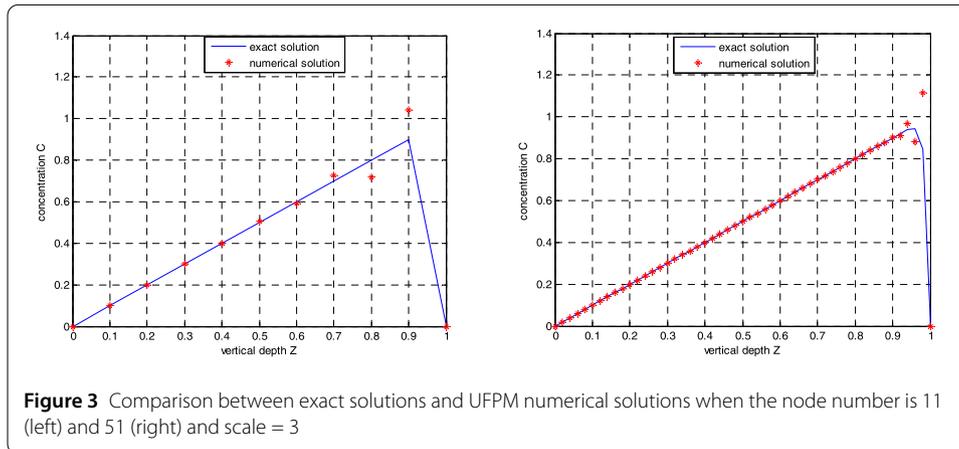
As can be seen from Table 1, the proposed method is very good at eliminating error when convection prevails. Numerical simulation results of the FPM are better than that

**Table 1** The comparison of calculation accuracy for different algorithms when  $v = 1, D_v = 0.01$

The number of nodes	FPM error	UFPM error		FEM error
		scale = 3	scale = 4	
11	$9.8060 \times 10^{-2}$	$5.6762 \times 10^{-2}$	$3.1391 \times 10^{-2}$	$8.4812 \times 10^{-1}$
21	$6.8497 \times 10^{-2}$	$7.9601 \times 10^{-2}$	$7.4891 \times 10^{-2}$	$8.3266 \times 10^{-1}$
51	$7.0516 \times 10^{-2}$	$4.0711 \times 10^{-3}$	$2.8583 \times 10^{-3}$	$8.3195 \times 10^{-1}$

**Table 2** The convergence order of the proposed algorithm when number of nodes is 11, 21, 51, respectively

The number of nodes	$\max  c^h - c $	Convergence order
11	$7.3093 \times 10^{-2}$	
21	$8.3941 \times 10^{-3}$	2.7194
51	$2.5913 \times 10^{-3}$	1.0362



of the FEM. The proposed method is not only simple and easy to implement, but can also achieve good numerical results in solving the convection-dominated partial differential equations.

The calculation of convergence order is

$$e_1 = \max |c_{EXA}^{h_1} - c_{FPM}^{h_1}|, \quad \text{the space step size is } h_1,$$

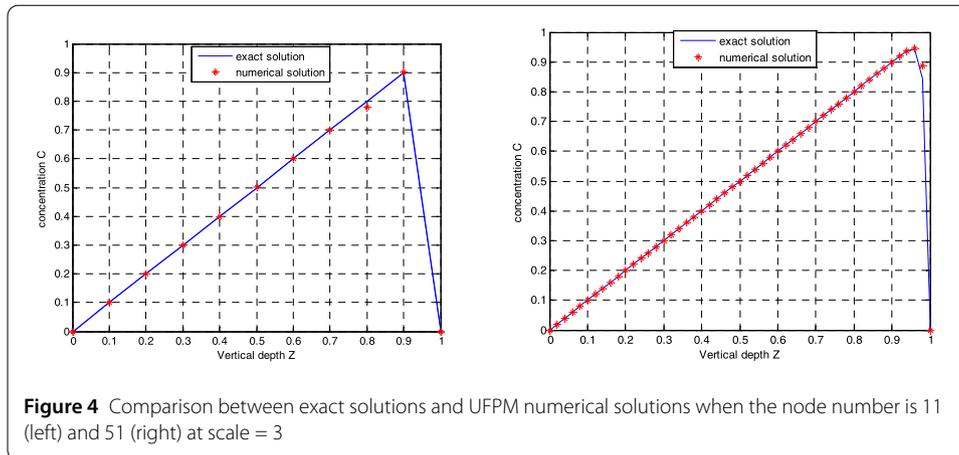
$$e_2 = \max |c_{EXA}^{h_2} - c_{FPM}^{h_2}|, \quad \text{the space step size is } h_2,$$

$$p = \frac{\log e_2/e_1}{\log h_2/h_1}.$$

where  $e_1, e_2$  are the maximum errors between the exact solution and the numerical solution at the nodes, and  $p$  indicates the convergence order.

As can be seen from Table 2, the higher number of nodes, the higher the calculation precision. As the convergence order is positive, the proposed algorithm in this paper is convergent.

According to Fig. 3 and Fig. 4, the gradient change is relatively large in the vicinity of  $x = 1$ . This is because the diffusion term is ignored. A thin boundary layer is formed at the downstream boundary where  $x = 1$ , which is difficult to replicate using a standard numerical algorithm. So it can lead to the shocked solution in the boundary layer. To solve this kind of problem, several methods can be adopted including encrypting the node, expanding the local support domain, and using an adaptive upwind scheme. The method adopted in this paper is the adaptive upwind scheme. As the Reynolds number ( $Re$ ) increases, the support field gradually inclines to the upwind side, so that the solution can capture more upstream information. It can get the numerical solution close to the exact solution, which shows the effectiveness of this method.



**Table 3** Comparison of calculation results of UFPM and FEM

Nodes	Time step	Time	UFPM error	FEM error
11	0.1	0.1	$6.1231 \times 10^{-4}$	$9.8443 \times 10^{-3}$
		0.5	$2.7573 \times 10^{-3}$	$6.0572 \times 10^{-2}$
		1.0	$6.5432 \times 10^{-3}$	$1.3996 \times 10^{-1}$
	0.01	0.1	$6.0982 \times 10^{-4}$	$1.0059 \times 10^{-3}$
		0.5	$2.8516 \times 10^{-3}$	$6.4227 \times 10^{-3}$
		1.0	$7.2553 \times 10^{-3}$	$1.4586 \times 10^{-2}$

**Table 4** Comparison of calculation results of UFPM and FEM

Nodes	Time step	Time	UFPM error	FEM error
21	0.1	0.1	$1.6307 \times 10^{-4}$	$9.8516 \times 10^{-3}$
		0.5	$1.0268 \times 10^{-3}$	$5.2944 \times 10^{-2}$
		1.0	$1.0844 \times 10^{-3}$	$1.0131 \times 10^{-1}$
	0.01	0.1	$1.5432 \times 10^{-4}$	$1.0313 \times 10^{-3}$
		0.5	$1.5568 \times 10^{-3}$	$5.6588 \times 10^{-3}$
		1.0	$3.8016 \times 10^{-3}$	$1.0672 \times 10^{-2}$

**Example 2** The one-dimensional non-steady-state salt transport is

$$\begin{cases} \frac{\partial c}{\partial t} = D_v \frac{\partial^2 c}{\partial x^2} - v \frac{\partial c}{\partial x}, & (x, t) \in (0, 1) \times (0, T), \\ c(x, 0) = e^{-x}, & x \in [0, 1], \\ c(0, t) = e^{(D_v + v)t}, & t \in [0, T], \\ c(1, t) = e^{-1 + (D_v + v)t}, & t \in [0, T]. \end{cases} \tag{3.2}$$

The exact solution is  $c(x, t) = (e^{-x} + x - \frac{x}{e} - 1)e^{(D_v + v)t} - (x - \frac{x}{e} - 1)e^{(D_v + v)t}$ .

Let  $v = 0.1$ ,  $D_v = 0.001$ , and use a Gauss function as the weight function.

According to Table 3, Table 4 and Table 5, the comparison of the proposed method with FEM shows that UFPM is highly accurate. As can be seen from Table 6, the higher the number of nodes, the higher the calculation precision when time step and time are determined. As the convergence order is positive, UFPM in this paper is convergent.

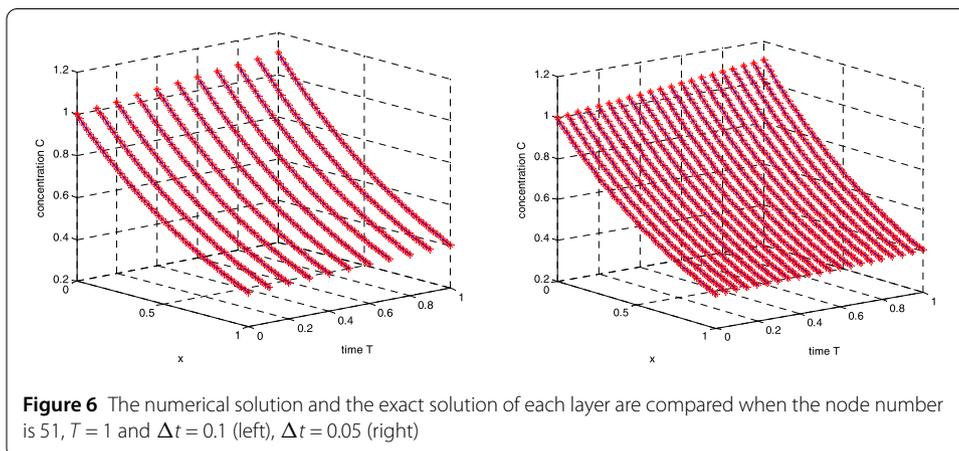
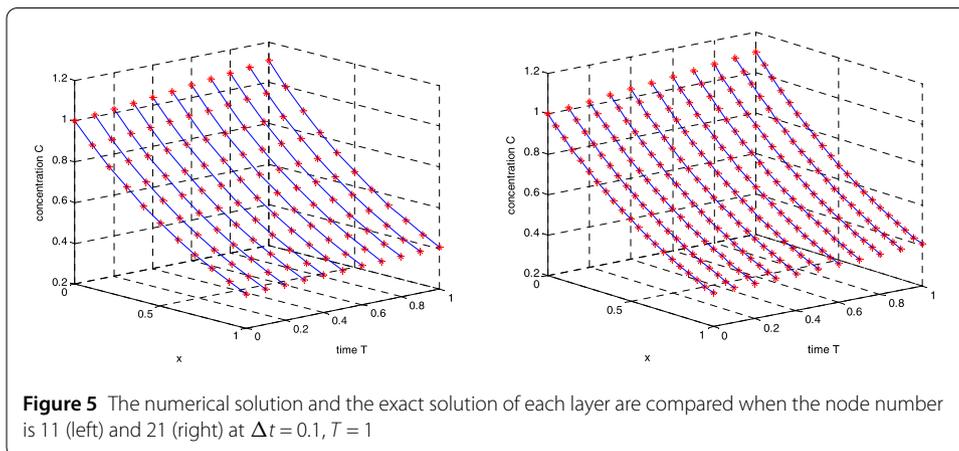
According to Fig. 5 and Fig. 6, by changing the time step, the number of nodes and using the Gauss function, numerical solutions of the proposed algorithm fit well with the exact solution.

**Table 5** Comparison of calculation results of UFPM and FEM

Nodes	Time step	Time	UFPM error	FEM error
51	0.1	0.1	$4.6744 \times 10^{-5}$	$9.3930 \times 10^{-3}$
		0.5	$2.2183 \times 10^{-4}$	$4.4196 \times 10^{-2}$
		1.0	$4.2643 \times 10^{-4}$	$4.4196 \times 10^{-2}$
	0.01	0.1	$9.0657 \times 10^{-6}$	$1.0053 \times 10^{-3}$
		0.5	$2.1202 \times 10^{-5}$	$1.0053 \times 10^{-3}$
		1.0	$2.5579 \times 10^{-5}$	$7.6178 \times 10^{-3}$

**Table 6** The convergence order of the UFPM with  $T = 1$  time steps when  $\Delta t = 0.1$

Nodes	$\max  c^h - c $	Convergence order
11	$9.8846 \times 10^{-4}$	
21	$4.2433 \times 10^{-4}$	1.2199
51	$6.1623 \times 10^{-5}$	2.1059



### 3.2 Nonlinear soil water movement equation

*Example 3* The one-dimensional soil water movement equation is

$$\begin{cases} \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} (D(\theta) \frac{\partial \theta}{\partial z}) + \frac{\partial K(\theta)}{\partial z} + f, & (z, t) \in (0, 1) \times (0, T), \\ \theta(z, 0) = \theta_0(z), & z \in [0, 1], \\ \theta(0, t) = \psi_1(t), & t \in [0, T], \\ \theta(1, t) = \psi_2(t), & t \in [0, T], \end{cases} \quad (3.3)$$

where  $K(\theta) = \theta$ ,  $D(\theta) = 0.001\theta + 0.0001$ , the exact solution  $\theta = (t + 1)(z - z^2)$ , the resource item  $f$ , the initial value  $\theta_0(z)$  and the boundary value  $\psi_1(t)$ ,  $\psi_2(t)$  are determined by the exact solution.

According to Table 7, the error between the numerical and exact solution is very small when the time is calculated to be 6.0. As can be seen from Table 8, the comparison of the proposed algorithm with CFEM shows that the proposed algorithm is highly accurate.

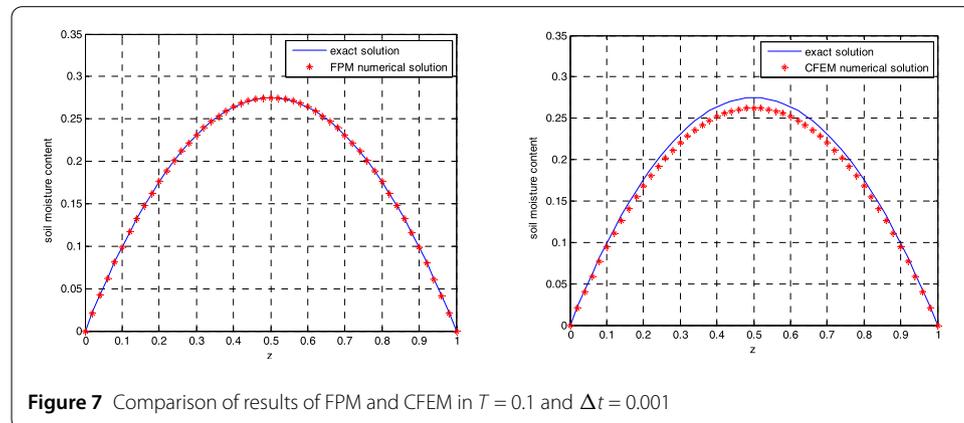
According to Fig. 7 and Fig. 8, the results of CFEM are not good when the number of nodes is 51. CFEM needs to increase the number of nodes to improve accuracy. However, the proposed algorithm can obtain good simulation results whether the number of nodes is 51 or 71.

**Table 7** Comparison of calculation results of FPM and CFEM

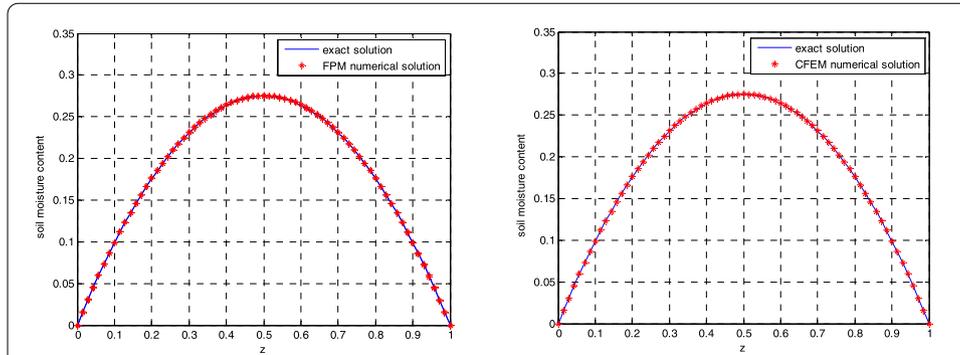
Nodes	Time step	Time	FPM error	CFEM error
21	0.005	$T = 0.01$	$5.7854 \times 10^{-4}$	$9.1582 \times 10^{-3}$
		$T = 0.1$	$5.5226 \times 10^{-3}$	$8.4067 \times 10^{-2}$
		$T = 1.0$	$1.8345 \times 10^{-2}$	$4.6110 \times 10^{-1}$
		$T = 6.0$	$2.5687 \times 10^{-2}$	$7.8642 \times 10^{-1}$

**Table 8** Comparison of calculation results of FPM and CFEM

Time step	Time	Nodes	FPM error	CFEM error
0.001	0.1	11	$8.4977 \times 10^{-4}$	$8.9466 \times 10^{-2}$
		21	$2.1892 \times 10^{-3}$	$8.4068 \times 10^{-2}$
		51	$1.8458 \times 10^{-3}$	$4.5057 \times 10^{-2}$
		61	$1.8463 \times 10^{-3}$	$2.3902 \times 10^{-2}$
		71	$1.8516 \times 10^{-3}$	$1.7377 \times 10^{-2}$



**Figure 7** Comparison of results of FPM and CFEM in  $T = 0.1$  and  $\Delta t = 0.001$



**Figure 8** Comparison of results of FPM and CFEM when the node number is 71,  $\Delta t = 0.001$  and  $T = 0.1$

**Table 9** The error and computing time of FPM

Time step	Nodes	FPM error		CPU
		scale = 2	scale = 3	
0.01	11 × 11	$8.9929 \times 10^{-3}$	$1.5175 \times 10^{-2}$	$2.6473 \times 10$
	21 × 21	$7.8850 \times 10^{-3}$	$1.2012 \times 10^{-2}$	$1.4678 \times 10^2$
	31 × 31	$4.6348 \times 10^{-3}$	$1.6127 \times 10^{-2}$	$1.5375 \times 10^2$
	41 × 41	$1.8824 \times 10^{-2}$	$1.8064 \times 10^{-2}$	$5.8794 \times 10^2$

**Table 10** Comparison of calculation results of FPM and FEM at different times

Nodes	Time step	Time	FPM error	FEM
21 × 21*	0.01	0.1	$1.6183 \times 10^{-2}$	$5.3746 \times 10^{-3}$
		1.0	$2.9284 \times 10^{-2}$	$4.9521 \times 10^{-2}$
		3.0	$2.9268 \times 10^{-2}$	$2.3492 \times 10^{-1}$
	0.001	0.01	$2.7731 \times 10^{-4}$	$2.3541 \times 10^{-3}$
		1.0	$1.0432 \times 10^{-2}$	$1.5324 \times 10^{-2}$
		3.0	$2.3723 \times 10^{-2}$	$4.5327 \times 10^{-1}$

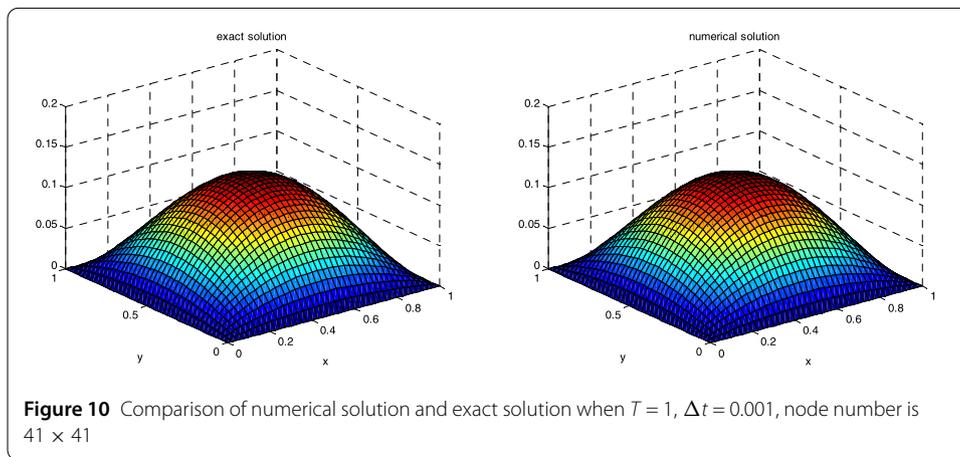
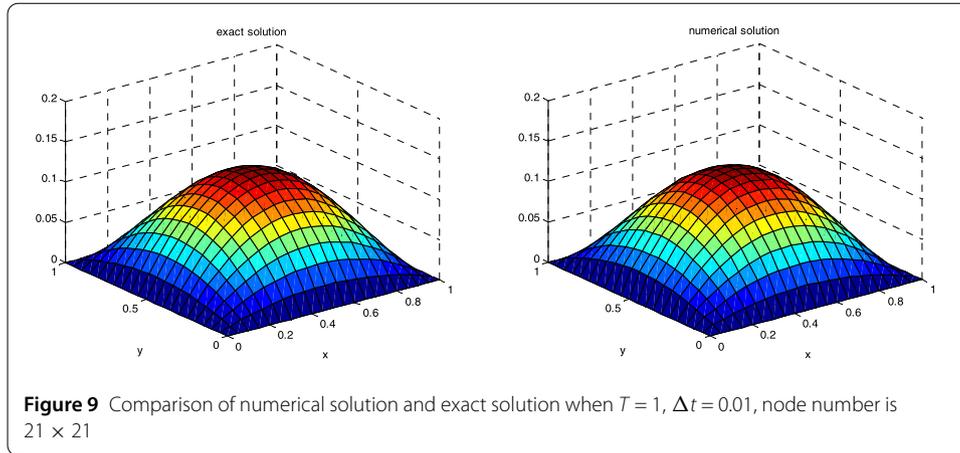
**Example 4** The two-dimensional soil water movement equation is

$$\begin{cases}
 \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial x}(D(\theta) \frac{\partial \theta}{\partial x}) + \frac{\partial}{\partial z}(D(\theta) \frac{\partial \theta}{\partial z}) + \frac{\partial K(\theta)}{\partial z} + f, & (x, z, t) \in (0, 1) \times (0, 1) \times (0, T), \\
 \theta(x, z, 0) = \theta_0(x, z), & (x, z) \in [0, 1] \times [0, 1], \\
 \theta(0, z, t) = \psi_1(z, t), & (z, t) \in [0, 1] \times [0, T], \\
 \theta(1, z, t) = \psi_2(z, t), & (z, t) \in [0, 1] \times [0, T], \\
 \theta(x, 0, t) = \psi_3(x, t), & (x, t) \in [0, 1] \times [0, T], \\
 \theta(x, 1, t) = \psi_4(x, t), & (x, t) \in [0, 1] \times [0, T],
 \end{cases} \tag{3.4}$$

where  $D(\theta) = 0.001\theta + 0.001$ ,  $K(\theta) = \theta$ , the exact solution is  $\theta = (t + 1)(x - x^2)(z - z^2)$ .

Initial condition:  $\theta_0 = (x - x^2)(y - y^2)$ , the resource item  $f$ , the initial value  $\theta_0(x, z)$  and the boundary value  $\psi_1(z, t)$ ,  $\psi_2(z, t)$ ,  $\psi_3(x, t)$ ,  $\psi_4(x, t)$  are determined by exact solution.

Table 9 gives the relative error and calculation time for the proposed method when the time step is 0.001 at time 0.1 under different scale and different number of nodes. Tables 3–10 show the error for finite point method at different time steps. Figures 9–10 depict the comparison of numerical solution and exact solution when  $T = 1$ ,  $\Delta t = 0.01$  at different node numbers.



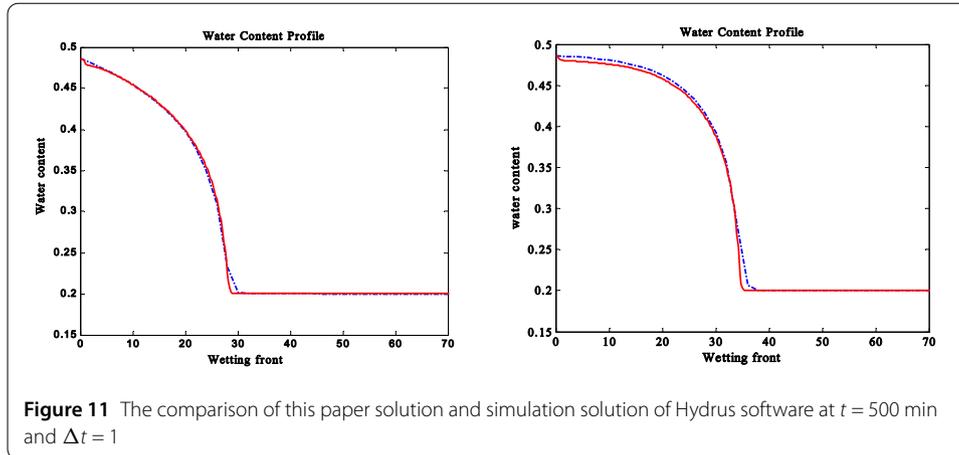
According to Table 9, the size of scale has an impact on the error. That is because the size of support domain directly affects the number of nodes in the support domain. On the whole, the error between the numerical and exact solutions is very small. From the perspective of computing time, the computational efficiency of the proposed algorithm is quite good. As can be seen from Table 10, the proposed algorithm is very stable. But FEM becomes unstable with the increase of time.

*Example 5* Simulation and analysis of soil water movement equation under the background of engineering water conservancy.

The one-dimensional soil water movement equation has the following form:

$$\begin{cases} \frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z} (D(\theta) \frac{\partial \theta}{\partial z}) + \frac{\partial K(\theta)}{\partial z} \cos \alpha, \\ \theta(z, 0) = \theta_0, \\ \theta(0, t) = \theta_s, \\ \theta(L, t) = \theta_0. \end{cases} \tag{3.5}$$

Here  $K(\theta)$  is the unsaturated hydraulic conductivity,  $h$  is the soil-water suction, and  $x$  is the horizontal position.  $\theta$  is the soil water content,  $\theta_s$  is the saturated soil water content, and  $\theta_0$  is the initial soil water content.  $D(\theta)$  is the diffusion rate. Equation (3.5) repre-



sents the vertical infiltration when  $\alpha = 0$ , and the horizontal infiltration when  $\alpha = \pi/2$ . The paper takes the Brooks–Corey [19] model as an example in engineering to discuss the feasibility of this method. So the expressions of  $D(\theta)$  and  $K(\theta)$  are as follows:

$$K(\theta) = K_s \left( \frac{\theta - \theta_r}{\theta_s - \theta_r} \right)^{m/n}, \tag{3.6}$$

$$D(\theta) = K_s \cdot h_d \cdot \left( -\frac{1}{n} \right) \cdot (\theta_s - \theta_r)^{(1-m)/n} \cdot (\theta - \theta_r)^{(m-1-n)/n}. \tag{3.7}$$

Here  $n$  is an index parameter,  $m = 2 + 3n$ ,  $h_d$  is the bubbling pressure (cm). The physical parameters of a given soil are as follows:  $\theta_r = 0.015$ ,  $\theta_s = 0.486$ ,  $\theta_0 = 0.2$ ,  $K_s = 0.0113333$ ,  $n = 0.211$ ,  $h_d = 20.747$ .

Figure 11 shows the simulated results of the one-dimensional vertical (right) and horizontal (left) infiltration for the given soil. The solid red line is the solution of this method, the dotted line represents the solution of the Hydrus software [13, 14].

To be compared with other algorithms, the errors of vertical and horizontal infiltration are 1.141% and 0.5% when the infiltration time is 500 min, respectively. As can be seen from Table 9 and Fig. 11, to a certain degree, this method is not only suitable for numerical examples, but also feasible for the actual background problem.

### 4 Conclusions

Through fully discussing the effects of support domain size, node number and time step size on the simulation results, it can be concluded that the number of nodes has a direct impact on the calculation accuracy. Under normal circumstances, the higher the number of nodes, the higher the accuracy of the numerical results. In addition, the smaller the time step, the smaller the calculation error. It can be seen from the convergence order that the proposed method is feasible. The proposed method applies stability terms to the governing equations before the discretization of the governing equations and makes its support domain gradually tend to windward in the area of large gradient change so that it can effectively avoid numerical shock and obtain good simulation results when dealing with nonlinear water movement equation. Numerical examples with comparison between the FEM and CFEM indicate that the proposed method has high precision. The numerical examples and the problems of hydraulic engineering in the background indicate that this

method can be used to solve the problem of alkaline areas. The accuracy and validity of this method have been verified, which provides a reasonable scientific basis for reasonable irrigation saline-alkali land. In addition, it has a certain positive effect on the development of agriculture. We will focus on studying the finite point procedure for solving three-dimensional problems in future work.

#### Acknowledgements

T. Abdeljawad would like to thank Prince Sultan University for funding this work through research group Nonlinear Analysis Methods in Applied Mathematics (NAMAM) group number RG-DES-2017-01-17.

#### Funding

Not applicable.

#### Availability of data and materials

Not applicable in this article.

#### Competing interests

The authors declare that they have no competing interests.

#### Authors' contributions

All authors have made equally contributions to the writing of this article. The final manuscript was read and approved by all authors.

#### Author details

<sup>1</sup>College of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China. <sup>2</sup>Department of Mathematics, Xi'an Traffic Engineering Institute, Xi'an 710300, China. <sup>3</sup>Department of Mathematics and General Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia. <sup>4</sup>Department of Medical Research, China Medical University, Taichung 40402, Taiwan. <sup>5</sup>Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan. <sup>6</sup>Department of Mathematics, University of Sargodha, Sargodha 40100, Pakistan.

#### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 17 June 2020 Accepted: 8 February 2021 Published online: 22 March 2021

#### References

1. Guo, R., Feng, Q., Si, J., et al.: Progress in the study of models for water and salinity transport in soils. *J. Glaciol. Geocryol.* **30**(3), 527–534 (2008)
2. Peng, J., Shao, A.: Numerical simulation of soil water salt movement. *China Rural Water Hydropower* **2**, 50–53 (2006)
3. Weibang, Z., Yufang, Z.: Preliminary Study on Numerical Simulation of Soil Water and Salt Transport. *Soil Dalian: Soil Physics Symposium Papers*, Dalian (1982)
4. Zhang, R.F., Zhang, D.S., Wu, X.Q.: Numerical simulation of one dimensional unsaturated soil water movement. *J. Basic Sci. Text. Coll. Univ.* **17**(2), 123–127 (2004)
5. Su, Y.H., Zhu, G.F., Qi, F.: Numerical simulation of one-dimensional vertical unsaturated soil water movement under evaporation. *China Desert* **29**(2), 236–240 (2009)
6. Guo, X.H., Sun, X.H., Ma, J.J.: Finite element simulation of soil water movement in single pit irrigation. *J. Agric. Eng.* **23**(10), 1–5 (2007)
7. Thomée, V.: *Galerkin Finite Element Methods for Parabolic Problems*. Springer, Berlin (1984)
8. Girault, V., Raviart, P.A.: *Finite Element Methods for Navier–Stokes Equations: Theory and Algorithms*. Springer, Berlin (2012)
9. Zhang, D.S.: *Study on the Mathematical Model of Solute Transport Through Soils*. Xi'an University of Technology, Xi'an (2004)
10. Zhao, M.L.: *Numerical Simulation for 2-D Water Flow and Solute Transport by Using the Finite Element Method*. Xi'an University of Technology, Xi'an (2006)
11. Brooks, R.H., Corey, A.J.: *Hydraulic Properties of Porous Media*. Colo. State Univ., Fort Collins (1964)
12. Šimůnek, J., van Genuchten, M.T., Šejna, M.: HYDRUS: model use, calibration and validation. *Trans. ASABE* **55**(4), 1261–1274 (2012)
13. Chuiko, S.M., Boichuk, I.A., Pirus, O.E.: On the approximate solution of an autonomous boundary-value problem by the Newton–Kantorovich method. *J. Math. Sci.* **189**(5), 867–881 (2013)
14. Eshkuvatov, Z.K., Ahmedov, A., Long, N.M.A.N., et al.: Approximate solution of the system of nonlinear integral equation by Newton–Kantorovich method. *Appl. Math. Comput.* **217**(8), 3717–3725 (2010)
15. Wang, J.M., Zhou, X.Z.: *The Theory of Meshless Method and Its Programming*. Shandong University Press (2007)
16. Sui, X.Y., Zhou, D.L.: Using the least square collocation method to solve the problem of the steady flow of groundwater. *Ground Water* **32**(1), 15–16 (2010)
17. Onate, E., Idelsohn, S., Zienkiewicz, O.C., et al.: A stabilized finite point method for analysis of fluid mechanics problems. *Comput. Methods Appl. Mech. Eng.* **139**(1), 315–346 (1996)

18. Ortega, E., Oñate, E., Idelsohn, S., et al.: Comparative accuracy and performance assessment of the finite point method in compressible flow problems. *Comput. Fluids* **89**(2), 53–65 (2014)
19. Bajko, J., Čermák, L., Jícha, M.: High order finite point method for the solution to the sound propagation problems. *Comput. Methods Appl. Mech. Eng.* **280**(10), 157–175 (2014)
20. Baleanu, D., Jajarmi, A., Sajjadi, S.S., Asad, J.: The fractional features of a harmonic oscillator with position-dependent mass. *Commun. Theor. Phys.* **72**(5), 055002 (2020)
21. Shao, M.G., Wang, Q.J., Huang, M.B.: *Soil Physics*. Higher Education Press, Beijing (2006)
22. Shi, Y., Li, Y.: The status and trend of saline soil reach. *Arid Zone Res.* **3**(4), 38–44 (1986)

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)

---