Filomat 34:5 (2020), 1517–1533
https://doi.org/10.2298/FIL2005517Q

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

Finite Point Method of Nonlinear Convection Diffusion Equation

Xinqiang Qin[a], Gang Hu[a], Gaosheng Peng[a]

[a]*Department of Applied Mathematics, Xian University of Technology, Xian, China*

Abstract. Aiming at the nonlinear convection diffusion equation with the numerical oscillations, a numerical stability algorithm is constructed. The basic principle of the finite point algorithm is given and the computational scheme of the nonlinear convection diffusion equation is deduced. Then, the numerical simulation of the one-dimensional and two-dimensional nonlinear convection-dominated diffusion equation is carried out. The relationship between the calculation result and the support domain size, step size and time is discussed. The results show that the algorithm has the characteristics of simplicity, stability and efficiency. Compared with the traditional finite element method and finite difference method, the new algorithm can attain a higher calculation accuracy. Simultaneously, it proves that the method given in this paper is effective to solve the nonlinear flow diffusion equation and can eliminate the numerical oscillations.
Keywords: The finite point method; Convection diffusion equation; Moving least square method; Collocation method.

1. Introduction

There are many complex problems in nature and engineering sciences that can be described by some partial differential equations. The fluid in the porous medium undergoes a variety of complex physical and chemical changes during the flow. The mathematical model of the change of the physical quantity comes down to the convection diffusion model. The convection diffusion equation is a class of basic equations of motion, and the convective diffusion equation generated in the practical problems is not linear. Besides, the coefficients of many equations are related to the solution. Therefore, the form of the nonlinear convection diffusion equation is proposed [1,2]. The study of nonlinear convection diffusion equation has its indispensible practical significance in this case. However, due to the complexity of the nonlinear equation and the convection dominance, it is more difficult to obtain the numerical solution of the equation. Many scholars have used the characteristic line method to study the nonlinear convection diffusion equation. The meshless method has developed rapidly in recent years. The core part of the meshless method lies in the construction and discrete way of the shape function, which is the fundamental difference between the meshless method and other numerical methods. More than ten kinds of meshless methods that are currently developed, and they use different shape functions and different discrete schemes, which can be combined into different meshless methods.

The finite point method use the moving least squares method to construct the approximation function. The irregular point distribution is used to disperse the equation with the stabilization term and the boundary

2010 *Mathematics Subject Classification*. Primary 65M70; Secondary 65M99
Keywords. The finite point method; Convection diffusion equation; Moving least square method; Collocation method
Received: 20 May 2018; Revised: 01 April 2019; Revised: 28 December 2019; Accepted: 24 May 2020
Communicated by Marko Petković
Research supported by the National Natural Science Foundation of China (No.51875454, No.51979220)
Email addresses: xqqin2007@163.com (Xinqiang Qin), hugang07@sina.com (Gang Hu)

condition is processed by the corresponding boundary treatment scheme. Thus, the discrete differential equation is obtained. This method was originally proposed by the Spanish and Idelsohn [3, 4, 5] and so on, then many scholars have studied the method. The method has the advantages of simple discrete format, high computational efficiency, easy programming and this method does not need to introduce the background mesh and calculate the numerical integration. Thus, it has been successfully applied in the compressed stream, sound propagation [6] and other practical applications.

At present, there are few studies on non-linear problems. Liu [7] introduced the stable finite point method to solve the convection-diffusion model and numerically simulated the two-dimensional steady-state convection-diffusion equation. In this paper, based on the previous studies, the finite point method is constructed for the application of stable terms by combining the nonlinear model of one-dimensional and two-dimensional convection diffusion equations with moving least squares construction approximation function. In addition, the analysis of the numerical simulation shows that the finite stability term is an effective method for solving the nonlinear convection-diffusion equation.

2. Stability finite point algorithm construction

The finite point method (FPM) is a true meshless method that does not require the background grid to be integrated. The method uses the least squares approximation to construct the shape function, which includes the selection of weight function and the influencing factors of the support domain. The dispersion of the equation use collocation method which has been successfully used in the fields of fluid and aerodynamics. When the finite point method is used to solve the convection diffusion equation, the stability of the method is needed to eliminate the oscillation, because of the existence of the convective term.

2.1. The moving least squares approximation

Moving Least Square (MLS) has a characteristics of strong local numerical analysis because of its special emphasis on tight support, which was first proposed by Lancaster [8], and many scholars have studied it. Cheng [9,10] proposed complex variable moving least squares method and improved moving least squares method. This section gives a brief introduction to the moving least squares approximation method.

In the solution interval [0, L], the unknown functions $u(x)$ can be approximated in the field Ω_x of the calculation point x:

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

$P^T(x) = [p_1(x), p_2(x), ..., p_m(x)]$ is called the interpolation basis function, and m is the number of the basis function, which is usually a monomial determined by the Pascal triangle to ensure its minimum completeness. $a^T(x) = [a_1(x), a_2(x), ..., a_m(x)]$, $a_i(x)$ is the coefficient to be determined, x is the calculation point,$\bar{x}$ is the spatial coordinates of the points Ω_x within the neighborhood of the calculation point.

Node $x = x_i(i = 1, 2, ..., n)$ is used to distribute for interval I=[0,L]. A tight weight function $W_i(x) = W(x - x_i)$ is defined in each node x_i, that is, the $W_i(x)$ weight function only in the domain Ω_i around the finite field of node x_i is greater than zero, outside the domain ω_i is equal to zero, the area ω_i is called as the support domain of the weight function $W_i(x)$ as well as the node x_i.The scale d_s of support domain in node x_i is generally determined by form $d_s = scale \cdot d_c$, where the dimensionless dimension for the support domain can be used to control the size of the actual support domain.d_c is the node spacing distributing around node x_i,d_c can be regarded as the distance between two adjacent nodes, when the nodes are evenly distributed. They can be regarded as the average node spacing in the support domain of node x_i , when the nodes are non-uniformly distributed.

Let the error weighted square sum of the approximate function $u^h(x,\bar{x})$ on the node $\bar{x} = x_i$ be

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

In this equation, n is the node numbers contained in the weighting function $W(x - x_i) \neq 0$ of x support domain. u_i is the node value of the unknown function in x_i node. In order to get the coefficient $a(x)$, we ask for the extreme value of J.

$$\partial J/\partial a = A(x)a(x) - B(x)U_s = 0 \tag{3}$$

In this equation, U_s is a vector that forms the node function values for all nodes in the domain

$$U_s = (u_1, u_2, ..., u_n)^T \tag{4}$$

$$A(x) = \sum^{n} W_i(x)p(x_i)p^T(x_i) \tag{5}$$

$$B(x) = [W_1(x)p(x_i), W_2(x)p(x_i), ..., W_n(x)p(x_i)] \tag{6}$$

Pending coefficient vector is worked out from formula (2.3)

$$a(x) = A^{-1}(x)B(x)U_s \tag{7}$$

Substituting (2.7) into equation (2.1)

$$u^h(x) = \sum^{N} \varphi_i(x)u_i = \Phi^T(x)U_s \tag{8}$$

Since the weight function is compact, we can get the global approximation function of unknown function

$$u^h(x) = \sum^{N} \varphi_i(x)u_i = \Phi^T(x)U \tag{9}$$

$$U = (u_1, u_2, ..., u_N)^T \tag{10}$$

In this equation,$\Phi(x)$ is the moving least squares function of n nodes in the support domain corresponding to the point x

$$\Phi(x) = \varphi_1, \varphi_2, ..., \varphi_n = P^T(x)A^{-1}(x)B(x) \tag{11}$$

If $\gamma^T = p^T A^{-1}$, the shape function will be $\Phi = \gamma^T B$, The first and second derivatives of the shape function are

$$\Phi^T_{,i} = \gamma^T_{,i}B + \gamma^T B_{,i}. \tag{12}$$

$$\Phi^T_{,ij} = \gamma^T_{,ij}B + \gamma^T_{,i}B_{,j} + \gamma^T B_{,ij} \tag{13}$$

2.2. *Adding stability term*

The finite point method is used to solve the convection diffusion equation. Due to the existence of the convective term, the stability is dealt with when the finite point method is used for numerical calculation [3].

Stability term is defined as follows - which refers to the residual term added to the control equation in order to prevent the numerical solution from oscillating due to the existence of the convection term in the convection-diffusion equation. This residual term is called the stability term.The stability term is obtained in finite region according to the principle of flow balance in fluid mechanics.

Consider the following convection diffusion equation

$$\begin{aligned} &L(v) = c\frac{\partial v}{\partial t} + u^T.\nabla v - \nabla^T(D\nabla v) - q = 0 && x\epsilon\Omega \\ &L_t(v) = n^T D\nabla v + Q_q = 0 && x\epsilon\Gamma_t \\ &v - v_p = 0 && x\epsilon\Gamma_u \end{aligned} \tag{14}$$

where ω is the domain, Γ_t and Γ_u are the boundaries.
First we consider the one-dimensional convection diffusion problem, as shown in the following figure
Explanation: Q represents Fluid flow rate,uv is advection, q(x) is source term, h is the length of the domain [3]. According to the above chart and Taylor's formula, the following results are obtained:

$$\begin{aligned} &Q(x_B - h) = Q(x_B) - h\frac{dQ}{dx}|_B + \frac{h^2}{2}\frac{d^2Q}{dx^2}|_B + O(h^3) \\ &[uv](x_B - h) = [uv](x_B) - h\frac{d[uv]}{dx}|_B + \frac{h^2}{2}\frac{d^2[uv]}{dx^2}|_B + O(h^3) \end{aligned} \tag{15}$$

According to the principle of balance of flow rate, can be obtained

$$\sum_{fluxes} = Qx_B + [uv](x_B) - Q(x_B - h) - [uv](x_B - h) - \frac{1}{2}[q(x_B) + q(x_B - h)]h = 0 \tag{16}$$

Among them, for the source term, the assumption is actually linearly distributed. It is also used in Taylor:

$$q(x_B - h) = q(x_B) - h\frac{dq}{dx}|_B + O(h^2) \tag{17}$$

Substituting (2.17) and (2.15) into (1.16), may wish to suppose B position is arbitrary. If $x_B = x$, the following results are attained.

$$-(\frac{d(uv)}{dx}) - \frac{dQ}{dx} + q + \frac{h}{2}\frac{d}{dx}[-(\frac{d(uv)}{dx}) - \frac{dQ}{dx} + q] = 0 \tag{18}$$

In this equation, Q and v is related and it can be written as:

$$Q = -D\frac{dv}{dx} \tag{19}$$

where, D is diffusion terms.
Substituting it into (2.18) gives the following formula

$$r - \frac{h}{2}\frac{dr}{dx} = 0 \tag{20}$$

of which

$$r = -\frac{d(uv)}{dx} + \frac{d}{dx}(D\frac{dv}{dx}) + q \tag{21}$$

The first form of (2.14) becomes the following form

$$L(v) = c\frac{\partial v}{\partial t} - r + \frac{h}{2}\frac{\partial r}{\partial x} = 0 \tag{22}$$

Where r is

$$r = -u\frac{\partial v}{\partial x} + \frac{\partial}{\partial x}(D\frac{\partial v}{\partial x}) + q \tag{23}$$

In two-dimensional situation :

$$L(v) = c\frac{\partial v}{\partial t} - r + \frac{h}{2\mid u\mid}u^T\nabla r = 0 \tag{24}$$

Where r is

$$r = -u^T\nabla v + \nabla^T(D\nabla v) + q \tag{25}$$

Based on the derivation of the process of applying the stable term, the finite element method of the meshless finite point algorithm is obtained. In the following, the finite point method for the convection diffusion equation is established and the corresponding numerical examples are validated.

3. Construction of the finite point algorithm for nonlinear convection diffusion equation

3.1. The algorithm of one-dimensional nonlinear convection diffusion equation

$$\begin{aligned} &\frac{\partial u}{\partial t} + b(x,u)\frac{\partial u}{\partial x} - \frac{\partial}{\partial x}(a(x,u)\frac{\partial u}{\partial x}) = f(x,u,t), && x\epsilon(0,L), t\epsilon(0,T),\\ &u(0,t) = f_1(t), \quad u(L,t) = f_2(t), && t\epsilon[0,T],\\ &u(x,0) = u_0(x), && x\epsilon[0,1]. \end{aligned} \tag{26}$$

Where $a(x,u)$, $f(x,u,t)$ are independent variable functions, As a result of the application of the stabilization term, the following consideration is given to the case where the convective term $b(x,u)$ is constant and is abbreviated as b.

The solution theory of (3.1) is not the object of the this paper. But the existence and unique of the exact solution and also about the existence and unique solution of FEM can be found in Alfio Q. and Alberto V. [11].

Transforming the control equation of (3.1) into:

$$\frac{\partial u}{\partial t} + b\frac{\partial u}{\partial x} - \frac{\partial a(x,u)}{\partial u}\Big(\frac{\partial u}{\partial x}\Big)^2 - a(x,u)\frac{\partial^2 u}{\partial x^2} - f(x,u,t) = 0 \tag{27}$$

The equation applies a stable term

$$L(u) = \frac{\partial u}{\partial t} - r + \frac{h}{2}\frac{\partial r}{\partial x} = 0 \tag{28}$$

Of which

$$r = -b\frac{\partial u}{\partial x} + \frac{\partial a(x,u)}{\partial u}\Big(\frac{\partial u}{\partial x}\Big)^2 + a(x,u)\frac{\partial^2 u}{\partial x^2} + f(x,u,t) \tag{29}$$

$$\begin{aligned}\frac{\partial r}{\partial x} &= -b\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 a(x,u)}{\partial u^2}\Big(\frac{\partial u}{\partial x}\Big)^3 + 2\frac{\partial a(x,u)}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + \frac{\partial a(x,u)}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + a(x,u)\frac{\partial^3 u}{\partial x^3} + \frac{\partial f(x,u,t)}{\partial x}\\ &= -b\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 a(x,u)}{\partial u^2}\Big(\frac{\partial u}{\partial x}\Big)^3 + 3\frac{\partial a(x,u)}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + a(x,u)\frac{\partial^3 u}{\partial x^3} + \frac{\partial f(x,u,t)}{\partial x}\end{aligned} \tag{30}$$

Constructing approximate function by moving least squares

$$u(x) \cong \hat{u}(x_m) = \sum_{i=1}^{N} \varphi_{mi}(x_m)u(x_i) \tag{31}$$

Where $\varphi_{im}(x)$ is the shape function of the i node obtained by the node x_m as the support domain center, $\varphi_{im}(x_m)$is the value of x_m node, of which, N is the number of nodes in the local support field used by the x_m shaped function
Substituting (3.4), (3.5) into (3.3)

$$\begin{aligned}&\frac{\partial u}{\partial t} + b\frac{\partial u}{\partial x} - \frac{\partial a(x,u)}{\partial u}\Big(\frac{\partial u}{\partial x}\Big)^2 - a(x,u)\frac{\partial^2 u}{\partial x^2} - f(x,u,t)\\ &+ \frac{h}{2}[-b\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 a(x,u)}{\partial u^2}\Big(\frac{\partial u}{\partial x}\Big)^3 + 3\frac{\partial a(x,u)}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2} + a(x,u)\frac{\partial^3 u}{\partial x^3} + \frac{\partial f(x,u,t)}{\partial x}] = 0\end{aligned} \tag{32}$$

Firstly, the spatial variables of the equations are discretized. From the construction of the approximation function of (3.6), we can get the approximate expression of the first, second and third derivative of the approximate function at node x_m , And then the time variable for forward differential dispersion, then the results are:

$$\begin{aligned}u_m^{k+1} &= u_m^k - \Delta t\sum_{i=1}^{N}[b\frac{\partial\varphi_{mi}(x)}{\partial x} - (a(x,u_m^k) + \frac{bh}{2})\frac{\partial^2\varphi_{mi}(x)}{\partial x^2} + \frac{h}{2}a(x,u_m^k)\frac{\partial^3\varphi_{mi}(x)}{\partial x^3}]u_i^k\\ &+ \triangle t[\frac{\partial a(x,u_m^k)}{\partial u}(\sum_{i=1}^{N}\frac{\partial\varphi_{mi}(x)}{\partial x}u_i^k)^2 - \frac{3h}{2}\frac{\partial a(x,u_m^k)}{\partial u}(\sum_{i=1}^{N}\frac{\partial\varphi_{mi}(x)}{\partial x}u_i^k)(\sum_{i=1}^{N}\frac{\partial^2\varphi_{mi}(x)}{\partial x^2}u_i^k)\\ &- \frac{h}{2}\frac{\partial^2 a(x,u_m^k)}{\partial u^2}(\sum_{i=1}^{N}\frac{\partial\varphi_{mi}(x)}{\partial x}u_i^k)^3 + f(x,u_m^k,t) - \frac{h}{2}\frac{\partial f(x,u_m^k,t)}{\partial x}]\,|_{x=x_m}\end{aligned} \tag{33}$$

The result after arranging:

$$\begin{aligned}u_m^{k+1} &= u_m^k - \Delta t\sum_{i=1}^{N}\Big[b\frac{\partial\varphi_{mi}(x)}{\partial x} - \Big(a(x,u_m^k) + \frac{bh}{2}\Big)\frac{\partial^2\varphi_{mi}(x)}{\partial x^2} + \frac{h}{2}a(x,u_m^k)\frac{\partial^3\varphi_{mi}(x)}{\partial x^3}\Big]u_i^k\\ &\quad + \triangle t\Big[\frac{\partial a(x,u_m^k)}{\partial u}\Big(\sum_{i=1}^{N}\frac{\partial\varphi_{mi}(x)}{\partial x}u_i^k\Big)^2 - \frac{3h}{2}\frac{\partial a(x,u_m^k)}{\partial u}\Big(\sum_{i=1}^{N}\frac{\partial\varphi_{mi}(x)}{\partial x}u_i^k\Big)\Big(\sum_{i=1}^{N}\frac{\partial^2\varphi_{mi}(x)}{\partial x^2}u_i^k\Big)\\ &\quad - \frac{h}{2}\frac{\partial^2 a(x,u_m^k)}{\partial u^2}\Big(\sum_{i=1}^{N}\frac{\partial\varphi_{mi}(x)}{\partial x}u_i^k\Big)^3 + f(x,u_m^k,t) - \frac{h}{2}\frac{\partial f(x,u_m^k,t)}{\partial x}\Big]\end{aligned} \tag{34}$$

Assuming that the boundary condition is the first kind of boundary condition, the function value of the boundary point is known, just calculate the interior point, If

$$\beta_{m,i} = -\Delta t[b\frac{\partial \varphi_{mi}(x)}{\partial x} - (a(x,u_m^k) + \frac{bh}{2})\frac{\partial^2 \varphi_{mi}(x)}{\partial x^2} + \frac{h}{2}a(x,u_m^k)\frac{\partial^3 \varphi_{mi}(x)}{\partial x^3}] \tag{35}$$

can be written as

$$\begin{aligned} u_m^{k+1} &= u_m^k + \sum_{i=2}^{N-1} \beta_{mi} u_i^k + \beta_{m1} u_1^k + \beta_{mN} u_N^k \\ &+ \Delta t \Big[\frac{\partial a(x,u_m^k)}{\partial u} \Big(\sum_{i=1}^{N} \frac{\partial \varphi_{mi}(x)}{\partial x} u_i^k \Big)^2 - \frac{3h}{2} \frac{\partial a(x,u_m^k)}{\partial u} \Big(\sum_{i=1}^{N} \frac{\partial \varphi_{mi}(x)}{\partial x} u_i^k \Big) \Big(\sum_{i=1}^{N} \frac{\partial^2 \varphi_{mi}(x)}{\partial x^2} u_i^k \Big) \\ &- \frac{h}{2} \frac{\partial^2 a(x,u_m^k)}{\partial u^2} \Big(\sum_{i=1}^{N} \frac{\partial \varphi_{mi}(x)}{\partial x} u_i^k \Big)^3 + f(x,u_m^k,t) - \frac{h}{2} \frac{\partial f(x,u_m^k,t)}{\partial x} \Big] \end{aligned} \tag{36}$$

The above equation is a nonlinear equation, the application for collocation method approach is used in discrete process, so a simple linear iteration is carried out in the process of solving the initial value problem. Then the equation linearization, the following specific examples are discussed.

3.2. The algorithm of two-dimensional nonlinear convection diffusion equation

$$\begin{aligned} &\frac{\partial u}{\partial t} + b_1 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} - \frac{\partial}{\partial x}\Big(a(x,y,u)\frac{\partial u}{\partial x}\Big) - \frac{\partial}{\partial y}\Big(a(x,y,u)\frac{\partial u}{\partial y}\Big) = f(x,y,u,t), \\ &u(0,y,t) = g_1(y,t), \quad u(L,y,t) = g_2(y,t), \\ &(x,0,t) = h_1(x,t), u(x,L,t) = h_2(x,t) \\ &u(x,y,0) = u_0(x,y) \end{aligned} \tag{37}$$

Then we derive the finite point algorithm. Because the steady term is applied, then just considering that the nonlinear term is only included in the source term and the diffusion term, and the convective terms b_1, b_2 are constant coefficients. Of which $b = [b_1, b_2]$, f(s,y,u,t) is abbreviated as f , and a(x,y,u) as a . Transforming (3.12) into:

$$\frac{\partial u}{\partial t} + b_1 \frac{\partial u}{\partial x} + b_2 \frac{\partial u}{\partial y} - \frac{\partial a}{\partial u}\Big(\frac{\partial u}{\partial x}\Big)^2 - a\frac{\partial^2 u}{\partial x^2} - \frac{\partial a}{\partial u}\Big(\frac{\partial u}{\partial y}\Big)^2 - a\frac{\partial^2 u}{\partial y^2} - f = 0 \tag{38}$$

$$r = -b_1 \frac{\partial u}{\partial x} - b_2 \frac{\partial u}{\partial y} + \frac{\partial a}{\partial u}\Big(\frac{\partial u}{\partial x}\Big)^2 + a\frac{\partial^2 u}{\partial x^2} + \frac{\partial a}{\partial u}\Big(\frac{\partial u}{\partial y}\Big)^2 + a\frac{\partial^2 u}{\partial y^2} + f \tag{39}$$

Then the gradient ∇r of r is

$$\nabla r = \begin{pmatrix} \frac{\partial r}{\partial x} & \frac{\partial r}{\partial y} \end{pmatrix}^T \tag{40}$$

After applying a stable term

$$\frac{\partial u}{\partial r} - r + \frac{h}{2\,|\,b\,|} \cdot b \cdot \nabla r = 0 \tag{41}$$

After Substituting (3.14) and (3.15) into (3.16)

$$\begin{aligned}
&\frac{\partial u}{\partial t}+b_1\frac{\partial u}{\partial x}+b_2\frac{\partial u}{\partial y}-\frac{\partial a}{\partial u}\Big(\frac{\partial u}{\partial x}\Big)^2-a\frac{\partial^2 u}{\partial x^2}-\frac{\partial a}{\partial u}\Big(\frac{\partial u}{\partial y}\Big)^2-a\frac{\partial^2 u}{\partial y^2}-f+\frac{h}{2\mid b\mid}\Big[b_1\Big(-b_1\frac{\partial^2 u}{\partial x^2}-b_2\frac{\partial^2 u}{\partial x\partial y}\\
&\quad+\frac{\partial^2 a}{\partial u^2}\Big(\frac{\partial u}{\partial x}\Big)^3+2\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+a\frac{\partial^3 u}{\partial x^3}+\frac{\partial^2 a}{\partial u^2}\frac{\partial u}{\partial x}\Big(\frac{\partial u}{\partial y}\Big)^2+2\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial y^2}\\
&\quad+a\frac{\partial^3 u}{\partial y^2\partial x}+\frac{\partial f}{\partial x}\Big)+b_2\Big(-b_2\frac{\partial^2 u}{\partial x^2}-b_1\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial^2 a}{\partial u^2}\Big(\frac{\partial u}{\partial x}\Big)^3+2\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x^2}+a\frac{\partial^3 u}{\partial x^3}\\
&\quad+\frac{\partial^2 a}{\partial u^2}\frac{\partial u}{\partial x}\Big(\frac{\partial u}{\partial x}\Big)^2+2\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial x\partial y}+\frac{\partial a}{\partial u}\frac{\partial u}{\partial x}\frac{\partial^2 u}{\partial y^2}+a\frac{\partial^3 u}{\partial y^3}+\frac{\partial f}{\partial x}\Big)\Big]=0
\end{aligned} \tag{42}$$

Construct an approximate function

$$u(x_p,y_q)=\sum_{i,j}\varphi_{pq,ij}(x_p,y_q)\cdot u(x_i,y_j) \tag{43}$$

Where $\varphi_{pq,ij}$ denotes the shape function formed by (x_p,x_q) as the support domain center and point (x_i,x_j).

First, the (3.17) space variables are semi-discrete, and then the time variables are fully discrete, and the approximate function expressions are substituted into the discrete equations to form the discrete partial differential equation.

$$
\begin{aligned}
u_{pq}^{k+1} &= u_{pq}^{k} - \Delta t \cdot \sum_{i,j}\Big[b_1 \frac{\partial \varphi_{pq,ij}(x,y)}{\partial x} + b_1 \frac{\partial \varphi_{pq,ij}(x,y)}{\partial y} - \Big(a + \frac{hb_1^2}{2\mid b\mid}\Big)\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial x^2} \\
&- \Big(a + \frac{hb_1^2}{2\mid b\mid}\Big)\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial y^2} - \frac{hb_1 b_2}{\mid b\mid}\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial y \partial x} + \frac{hab_1}{2\mid b\mid}\frac{\partial^3 \varphi_{pq,ij}(x,y)}{\partial y^3} \\
&+ \frac{hab_2}{2\mid b\mid}\frac{\partial^3 \varphi_{pq,ij}(x,y)}{\partial x^3} + \frac{hab_1}{2\mid b\mid}\frac{\partial^3 \varphi_{pq,ij}(x,y)}{\partial y^2 \partial x} + \frac{hab_2}{2\mid b\mid}\frac{\partial^3 \varphi_{pq,ij}(x,y)}{\partial x^2 \partial y}\Big]\,|_{x=x_p, y=y_q} \cdot u_{ij}^k \\
&+ \Delta t\Big[\frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)^2 + \frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big)^2 \\
&- \frac{hb_1}{2\mid b\mid}\Big[\frac{\partial^2 a}{\partial u^2}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)^3 + 3\frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)\Big(\sum_{i,j}\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial x^2}u_{ij}^k\Big) \\
&+ \frac{\partial^2 a}{\partial u^2}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big)^2 + 2\frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big) \\
&\Big(\sum_{i,j}\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial x \partial y}u_{ij}^k\Big) + \frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y \partial x}u_{ij}^k\Big)^2\Big] \\
&- \frac{hb_2}{2\mid b\mid}\Big[\frac{\partial^2 a}{\partial u^2}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big)^3 + 3\frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big) \\
&\Big(\sum_{i,j}\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial y^2}u_{ij}^k\Big) + \frac{\partial^2 a}{\partial u^2}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big)\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)^2 \\
&+ 2\frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial x}u_{ij}^k\Big)\Big(\sum_{i,j}\frac{\partial^2 \varphi_{pq,ij}(x,y)}{\partial x \partial y}u_{ij}^k\Big) + \frac{\partial a}{\partial u}\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y}u_{ij}^k\Big) \\
&\Big(\sum_{i,j}\frac{\partial \varphi_{pq,ij}(x,y)}{\partial y \partial x}u_{ij}^k\Big)^2\Big] + f^k - \frac{hb_1}{2\mid b\mid}\frac{\partial f^k}{\partial x} - \frac{hb_2}{2\mid b\mid}\frac{\partial f^k}{\partial y}\Big]\,|_{x=x_p, y=y_q}
\end{aligned} \tag{44}
$$

The two-dimensional boundary problem is similar to the one-dimensional problem. The algebraic equation of the partial differential equation can be obtained by the collocation method. The Nonlinear equations are linearized by simple linear iterations, and the construction process of the finite algorithm is similar to one-dimensional nonlinearity.

4. Numerical Simulation

In the numerical simulation, if we know the relationship between time step and space step, We can choose the appropriate time step and space step, and maybe can get good numerical results.This relationship is generally obtained through error analysis, such as finite element method and finite difference method. However, due to the difference between grid-free methods and traditional methods, it is difficult to analyze the error of grid-free method. At present, such research results are rarely seen, and more of them are used to verify the effectiveness of the algorithm through numerical experiments. Based on the current research status, we can only analyze the effectiveness of the methods presented in this paper through different numerical experiments. So we use two liner examples and two nonliner examples and fully demonstrate the convergence under different parameters,such as space step,time step, number of nodes, etc . And The error comparison of different algorithms is also given. It can reflect the effectiveness and feasibility of this methods to some extent.

4.1. One-dimensional numerical examples

Example 1 Consider the initial boundary value model of the convection-dominated convection diffusion equation

$$\begin{aligned} &\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}-a(x,u)\frac{\partial^2 u}{\partial x^2}=f(x,u,t), \quad x\epsilon(0,L), t\epsilon(0,T),\\ &u(0,t)=f_1(t), \quad u(L,t)=f_2(t), \qquad t\epsilon[0,T],\\ &u(x,0)=x(1-x), \qquad x\epsilon[0,1]. \end{aligned} \tag{45}$$

where $a(x,u)=0.001$, the exact solution is $u(x,t)=x(1-x)e^{xt}, f(x,u,t)=-u^2+G(x,t)$,$G(x,t)$ can be substituted into the equation by $u(x,t), f(x,u,t)$. The basis function is quadratic basis function $p^T=(1,x,x^2)$, scale is 3, and the weight function selects the Gaussian weight function. The convergence orders of the algorithm are as follows.

Table 1: △t=0.001, The convergence order about the algorithm of this paper when T=0.5

Space step	$max \mid u_{exa}-u_{FPM} \mid$	Convergence order
0.4	2.8369e-001	
0.2	1.5584e-002	3.9370
0.1	6.2717e-003	1.3131
0.05	3.3539e-003	0.9030

Table 2: △x=0.1, △t=0.002, △t=0.01, The different kinds of algorithm compare about the error

Space step	Time Step	Time	FPM error	FEM error	FDM error
0.1	0.002	T=0.1	2.8320e-003	1.6171e-002	3.1122e-002
		T=0.5	1.2482e-002	4.4704e-002	5.8201e-002
		T=1.0	2.3778e-002	5.5131e-001	8.0023e-001
	0.01	T=0.1	2.8680e-003	1.6278e-002	5.3822e-002
		T=0.5	1.2535e-002	4.4589e-002	1.9681e-001
		T=1.0	2.3514e-002	5.4908e-001	8.9681e-001

Table 1 shows the convergence order of the algorithm at T = 0.5. It can be seen from the table that when the time step is determined and the solution time is determined.The finite point algorithm in this paper is convergent when the space step becomes smaller.

From Table 2 and 3, it can be seen that take the same space step length and different time steps, the result is different from the exact solution at different time, the smaller the time step is, the smaller the error is, and the method is more stable when the numerical of iterations reach hundreds of layers. Compared with the finite difference and finite element method, the advantages of the proposed method are very obvious. Figure 1 shows the comparison of the exact and numerical solutions of the algorithm at different moments.

It can be seen from Fig. 1 that the numerical solution and the exact solution fit well. It is feasible to solve the convection diffusion equation with the nonlinear term. It also validates the validity of the stable finite point method of nonlinear convection diffusion equation.

Example 2 Consider the initial boundary value model of convection-dominated convection diffusion equation

$$\begin{aligned} &\frac{\partial u}{\partial t}+b(x,u)\frac{\partial u}{\partial x}-\frac{\partial}{\partial x}\Big(a(x,u)\frac{\partial u}{\partial x}\Big)=f(x,u,t),\\ &u(0,t)=0, \quad u(1,t)=0,\\ &u(x,0)=x(1-x), x\epsilon[0,1]. \end{aligned} \tag{46}$$

Table 3: $\triangle$x=0.05, $\triangle$t=0.002, $\triangle$t=0.02, The different kinds of algorithm compare about the error

Space step	Time Step	Time	FPM error	FEM error	FDM error
0.05	0.002	T=0.1	3.3990e-003	2.8477e-002	3.1122e-002
		T=0.2	5.7140e-003	4.2743e-002	4.9947e-002
		T=1.0	5.7760e-003	7.7798e-001	3.8832e-001
	0.02	T=0.1	4.7240e-003	2.9501e-002	2.7839e-002
		T=0.2	9.1570e-003	4.3485e-002	9.4371e-002
		T=1.0	2.5313e-002	7.5986e-001	6.7201e-001

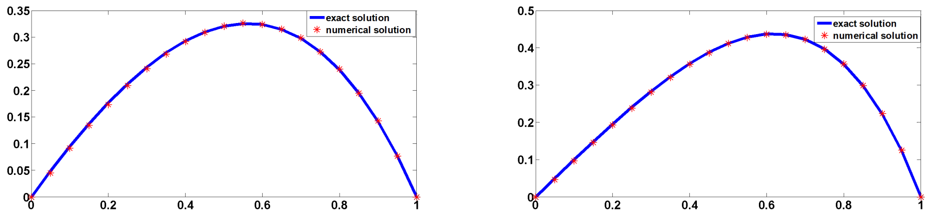

Figure 1: T=0.5 (left), T=1.0 (right), $\triangle$t=0.002,The algorithm of this paper compared with the exact solution when $\triangle$x=0.05

In this case $b(x,u) = 1$,$a(x,u) = \frac{1}{\pi^3}u$, the exact solution of this problem is $u(x,t) = e^t(x - x^2)$, the basis function is a quadratic basis function $p^T = (1, x, x^2)$,$f(x,u,t) = e^t(1 - x - x^2) - \frac{1}{\pi^3}ue^{2t}(6x^2 - 6x + 1)$ can be obtained by substituting the equation u(x,t) and a(x,u) into the equation, Gaussian weight function is taken as the weight function.

Table 4: $\triangle$t=0.002, The convergence order about the algorithm of this paper when T=0.5

Space step	$max \mid u_{exa} - u_{FPM} \mid$	Convergence order
0.10	0.0310	
0.05	0.0032	3.2761
0.02	0.0031	0.0346
0.01	0.0029	0.0474

Table 4 shows the convergence order of the algorithm in T = 0.5. It can be seen from the table that when the time step is determined and the solution time is determined. As we can see, the finite point algorithm in this paper is convergent when the space step becomes smaller..

The time step is 0.0002, the space step is 0.02, and the algorithm errors are compared at different time in different support domain size scale. As can be seen from Table 5, the result is better than 2 when Scale takes 3 and 4, noticing that different scale values have a certain effect on the solution.

The following study takes the effect of different time steps, the same space step size on the simulation results and the same time step, and the effect of different space steps on the simulation results. Table 6 shows the comparison of errors among the proposed method, finite difference method and the finite element method at different time. Table 7 shows the error and calculation of the algorithm when T = 1 time with different space steps.

From Table 6 and 7, it can be seen that the results are in good agreement with the exact solution, and compared with the finite difference, the finite element, it is obvious that the accuracy of the proposed method is very high. At the same time, the time step is fixed and the space step is taken. When the time is T = 1, the algorithm is more accurate and basically stable in the order of e-3. As the space step decreases, the calculation time increases. Figure 2 shows the comparison of the exact solution and the numerical solution of the algorithm at different moments.

Table 5: $\triangle t$=0.0002, $\triangle x$=0.02, Effect of Errors on support domain at different time

Time Step	Space step	Time	Scale=2	Scale=3	Scale=4
0.0002	0.02	0.2	5.2800e-004	5.2200e-004	5.2210e-004
		0.5	7.5300e-004	7.4500e-004	7.4100e-004
		1	6.7500e-004	6.5600e-004	6.4900e-004
		2	7.4400e-004	6.7100e-004	6.3100e-004
		3	9.6300e-004	6.7500e-004	5.4100e-004
		4	1.2261e-003	7.3200e-004	5.9100e-004

Table 6: $\triangle x$=0.05, $\triangle t$=0.0002, $\triangle t$=0.002, The different kinds of algorithm compare about the error

Space step	Time Step	Time	FPM error	FEM error	FDM error
0.05	0.002	T=0.2	4.9900e-004	9.8443e-003	5.2454e-002
		T=0.5	7.2100e-004	6.0572e-002	8.9346e-002
		T=1.0	6.3100e-004	9.3996e-002	7.8614e-001
	0.002	T=0.2	4.9060e-003	8.0059e-003	5.2564e-002
		T=0.5	7.0040e-003	6.4227e-002	9.9371e-002
		T=1.0	6.0480e-003	1.4586e-001	7.8613e-001

Table 7: $\triangle t$=0.002, Effect of Errors on space step at T=1

Time Step	Time Space	step	FPM error	Calculating time
0.002	T=1.0	0.10	5.6170e-003	0.6719
		0.05	6.0489e-003	0.5313
		0.02	6.2580e-003	0.7344
		0.01	5.9610e-003	2.1563

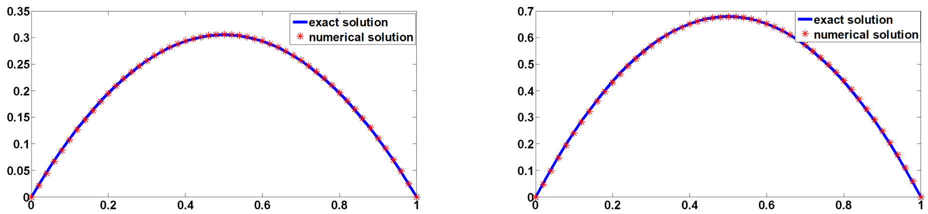

Figure 2: T=0.2 (left), T=1.0 (right), △t=0.002,The algorithm of this paper compared with the exact solution when △x=0.02

It can be seen from Figure 2 that the numerical solution and the true solution are better fitted. It is shown that the finite point method can be used to solve nonlinear one-dimensional convection diffusion equation.

4.2. *Two-dimensional numerical examples*

Example 1 Consider the initial boundary value model of the convection-dominated convection-diffusion equation

$$\begin{aligned}
&\frac{\partial u}{\partial t}+\frac{\partial u}{\partial x}+\frac{\partial u}{\partial y}-0.001\Big(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\Big)+f(x,y,u,t)\\
&u(0,y,t)=0,u(L,y,t)=0\\
&u(x,0,t)=0,u(x,L,t)=0\\
&u(x,y,0)=0
\end{aligned}\tag{47}$$

where$(x,y)\epsilon[0,1]\times[0,1]$ $t\geq 0$, The exact solution is $u(x,y,t) = txy(1-x)(1-y)e^{x+y}$,$f(x,y,u,t) = -u^2+G(x,y,t)$,$G(x,y,t)$ can be substituted into the Equation by $u(x,y,t)f(x,y,u,t)$. The basis function is quadratic basis function and the weight function selects the Gaussian weight function.

The solution domain is $[0,1]\times[0,1]$ separated by 11×11 uniform nodes, taking time step $\Delta t = 0.002$, comparison table of scale is 3, and the difference between finite error and finite element error

Table 8: 11×11 nodes, The algorithm of this paper compared with the FEM

Number of nodes	Number of iterations	FPM error	FEM error
11×11	10	3.9558e-003	2.9393e-002
	20	4.7260e-003	5.2418e-002
	30	7.9601e-003	7.2776e-002
	40	1.1814e-002	9.2139e-002
	50	1.5814e-002	1.1143e-001

The solution domain is $[0,1]\times[0,1]$, separated by 41×41 uniform nodes, taking time step $\Delta t = 0.002$, scale is 3, and the comparison of errors between finite point method and finite element method is listed.

It can be seen from the above table and graph that the finite point method is suitable for solving the nonlinear two-dimensional convection diffusion equation, which is more accurate than the finite element method. And the numerical solution fits well with exact solution. In this paper, we study the influence of different nodes on the algorithm error, taking the time step as 0.001, using the simple linear iteration to the 10th layer. The error is shown in Table 10.

As can be seen from Table 10, the number of nodes increases and the error decreases. The finite point method can be used to solve the nonlinear two-dimensional convection diffusion equation.

Table 11 shows the convergence order of the algorithm in T = 0.5. It can be seen from the table that the finite point algorithm in this paper is convergent when the space step becomes smaller.

Table 9: 41×41 nodes, The algorithm of this paper compared with the FEM

Number of nodes	Number of iterations	FPM error	FEM error
41×41	10	4.1808e-003	1.4074e-002
	20	8.7934e-003	2.1519e-002
	30	1.5019e-002	2.8161e-002
	40	2.1438e-002	3.5746e-002
	50	2.7730e-002	4.9367e-002
	100	6.4720e-002	1.2160e-001

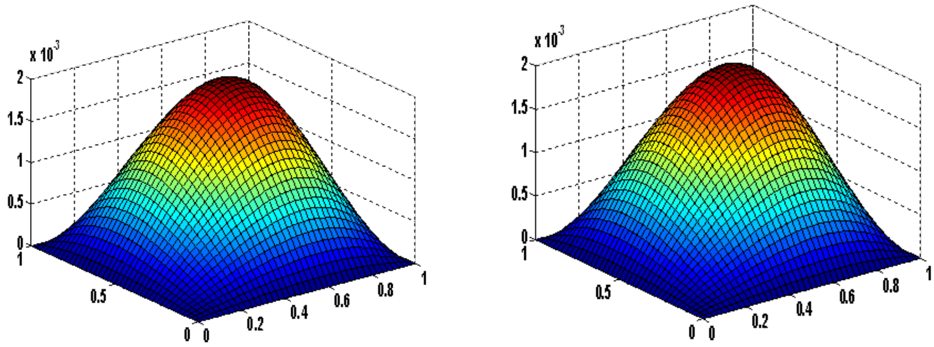

Figure 3: The surfaces of the exact solution(left) and numerical solution(right) for 41×41 nodes(T=0.1)

Example 2 Consider the initial boundary value model of convection-dominated convection diffusion equation

$$\begin{aligned} &\frac{\partial u}{\partial t} + b_1\frac{\partial u}{\partial x} + b_2\frac{\partial u}{\partial y} - \frac{\partial}{\partial x}\Big(a(x,y,u)\frac{\partial u}{\partial x}\Big) - \frac{\partial}{\partial y}\Big(a(x,y,u)\frac{\partial u}{\partial y}\Big) + f(x,y,u,t)\\ &u(0,y,t) = g_1(y,t), \quad u(L,y,t) = g_2(y,t)\\ &u(x,0,t) = h_1(x,t), \quad u(x,L,t) = h_2(x,t)\\ &u(x,y,0) = u_0(x,y) \end{aligned} \tag{48}$$

where $(x,y,t)\epsilon[0,L]\times[0,L]\times[0,T], b_1 = b_2 = 1, a(a,y,u) = 0.001u + 0.001$ the exact solution $u(x,y,t) = e^t(x^2 - x)(y^2 - y)$. The support domain impact factor scale is 3, select the Gaussian weight function as the Gaussian weight function. Table 4-12 and Table 4-13 below show the algorithm error of 11×11 nodes and 21×21 nodes at different times.

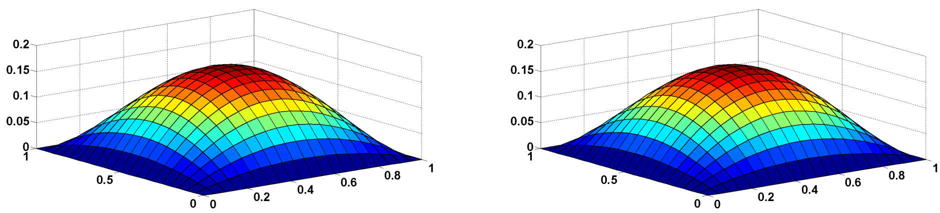

Figure 4: The surfaces of the exact solution(left) and numerical solution(right) for 21×21 nodes(T=1)

From the above figure,we can see that taking the number of different nodes, the numerical solution coincide well with the exact solution at different times. The number of nodes is 21×21, and the time step

Table 10: The calculation accuracy of numerical method at different grid node when T=0.01

Number of nodes	Number of iterations	Number of nodes	FEM error
0.001	10	6 ×6	6.3070e-003
		11×11	3.2510e-003
		21×21	1.1680e-003
		31×31	6.1900e-004
		41×41	4.2300e-004

Table 11: △t=0.002, The convergence order about the algorithm of this paper when T=0.5

Space step	$max \mid u_{exa} - u_{FPM} \mid$	Convergence order
0.2	2.8094e-002	
0.1	1.8407e-002	0.6100
0.05	7.1227e-003	1.3698
0.025	1.6959e-003	2.0704

Table 12: The calculation accuracy of numerical method at different time

Time Step	Number of nodes	Time	FPM error
0.0002	11× 11	T=0.2	7.1250e-003
		T=1	1.4010e-002
		T=2	1.4173 e-002
0.002	11× 11	T=0.2	1.0701e-002
		T=1	1.7396e-002
		T=2	1.7873e-002

Table 13: The calculation accuracy of numerical method at different time

Time Step	Number of nodes	Time	FPM error
0.0002	21× 21	T=0.2	1.5420e-003
		T=0.5	2.0530e-003
		T=1	1.9720e-003
		T=2	2.0250e-003
0.002	21× 21	T=0.2	7.4770e-003
		T=0.5	8.3260e-003
		T=1	6.9120e-003
		T=2	7.5670e-003

is taken as 0.001. The errors of algorithm in this paper and finite element method are compared at different times.

Table 14: 21×21 nodes ,The algorithm of this paper compared with the FEM

Number of nodes	Time	FPM error	FEM error
21×21	0.01	9.3570e-004	3.5750e-003
	0.05	8.3260e-003	1.5528e-002
	0.10	6.4770e-003	9.9026e-002
	0.50	8.3260e-003	1.5178e-001

It can be seen from Table 14 that when the number of nodes is 21×21, the error of the method in this paper is smaller and can be stabilized in the order of e-3, when comparing the finite point algorithm with finite element method at different times.Table 15 shows the convergence order of the algorithm in T = 0.5. It can be seen from the table that the finite point algorithm in this paper is convergent when the space step becomes smaller. Thus, the finite point algorithm can effectively solve the two-dimensional convection diffusion equation with the nonlinear term.

Table 15: $\triangle$t=0.002, The convergence order about the algorithm of this paper when T=0.5

Space step	$max \mid u_{exa} - u_{FPM} \mid$	Convergence order
0.4	6.3311e-002	
0.2	6.4930e-003	3.2855
0.1	1.3397e-003	2.2770
0.05	4.2088e-004	1.6904

5. Conclusion

The numerical results show that the finite point method is feasible to solve the nonlinear convection diffusion equation by the finite point algorithm, which can eliminate the numerical oscillation caused by convection dominance. From the calculation results, the support domain size, step size and time directly affect the calculation accuracy, and the solution is simple and has high precision. Compared with the traditional difference scheme and finite element method, the calculation error is more advantageous. In addition, the method does not need to be meshed, and does not need grid integration, which makes the method have the advantages of low computational cost, high calculation precision and stability.

Acknowledgments
This work is supported by the National Natural Science Foundation of China (No.51875454, No.51979220). Thank the reviewers for their suggestions and comments.

References

[1] Lu Jinshu, Wu Wanqing. Gas liquid interface moves oil tank gas phase zone of hydrocarbon gas convection diffusion model[J]. Oil and gas storage and transportation, 2008, 27(4): 18-20.
[2] Shen Laihong, Zhang Mingyao. In the bubbling fluidized bed particle mixed convection-diffusion model[J].Proceedings of the csee, 1995, 15(1): 45-53.
[3] Onate E, Idelsohn S, Zienkiewicz O C, et al. A finite point method in computational mechanics. Applications to convective transport and fluid flow[J]. International journal for numerical methods in engineering, 1996, 39(22): 3839-3866.
[4] Onate E, Idelsohn S, Zienkiewicz O C, et al. A stabilized finite point method for analysis of fluid mechanics problems[J]. Computer Methods in Applied Mechanics and Engineering, 1996, 139(1-4): 315-346.
[5] Ortega E, Oñate E, Idelsohn S, et al. Comparative accuracy and performance assessment of the finite point method in compressible flow problems[J]. Computers Fluids, 2014, 89: 53-65.

[6] Bajko J, Čermák L, Jacha M. High order finite point method for the solution to the sound propagation problems[J]. Computer Methods in Applied Mechanics and Engineering, 2014, 280: 157-175.
[7] Liu Xin. Meshless method [M]. Beijing: Science press, 2011.
[8] Lancaster P, Salkauskas K. Surfaces generated by moving least squares methods[J]. Mathematics of computation, 1981, 37(155): 141-158.
[9] Yu-Min Cheng, Li Jiugong. Elastic force of the complex variable meshless method [J]. Journal of physics, 2005, (10): 4663-4471.
[10] Yu-Min Cheng, Peng Miaojuan, Li Jiugong. Complex variable moving least-square method and its application [J]. Journal of mechanics, 2005, 37 (6): 719-723.
[11] Alfio Q, Alberto V. Numerical approximation of partial differential equations. Springer-Verlag,1996.