Bol. Soc. Paran. Mat. (3s.) **v. 36** 2 (2018): 83–97.
©SPM –ISSN-2175-1188 on line ISSN-00378712 in press
SPM: www.spm.uem.br/bspm doi:10.5269/bspm.v36i2.30447

Hermite Interpolant Multiscaling Functions for Numerical Solution of the Convection Diffusion Equations

E. Ashpazzadeh and M. Lakestani*

ABSTRACT: A numerical technique based on the Hermite interpolant multiscaling functions is presented for the solution of Convection-diffusion equations. The operational matrices of derivative, integration and product are presented for multiscaling functions and are utilized to reduce the solution of linear Convection-diffusion equation to the solution of algebraic equations. Because of sparsity of these matrices, this method is computationally very attractive and reduces the **CPU** time and computer memory. Illustrative examples are included to demonstrate the validity and applicability of the new technique.

Key Words: Hermite interpolant multiscaling functions, Convection-diffusion equation, operational matrix of derivative, Operational matrix of integration, operational matrix of product.

Contents

1. Introduction

Consider the convection-diffusion equation with variable coefficients

$$\frac{\partial u}{\partial t} + \alpha(x)\frac{\partial u}{\partial x} = \beta(x)\frac{\partial^2 u}{\partial x^2} + q(x,t), \tag{1.1}$$

with initial condition

$$u(x,0) = f(x), \quad 0 \le x \le \ell, \tag{1.2}$$

* Corresponding Author.
2010 *Mathematics Subject Classification:* 65T60, 65N35.
Submitted December 29, 2015. Published May 14, 2016

and boundary conditions

$$u(0,t) = g_0(t), \quad u(\mathsf{T},t) = g_1(t), \quad 0 \le t \le \mathsf{T}, \tag{1.3}$$

where $\alpha(x)$ and $\beta(x)(\neq 0)$ are continuous functions and $f(x)$, $g_0(t)$, $g_1(t)$ are known functions while u is unknown function. Convection diffusion equation appears in many branches of engineering, such as fluid mechanics, heat transfer and problems in structural mechanics posed over thin domains [4]. Since it is impossible to solve convection diffusion equations analytically for most application problems, efficient numerical algorithms are becoming increasingly important to numerical simulations involving convection diffusion equations. For this model, some authors have studied the numerical techniques such as finite difference method [1,4,17], the Bessel collocation method [19], the finite element method [6,7], the wavelet-Galerkin method [2,8], the Crank-Nicolson method [18], the piecewise-analytical method [16], ADI method [11] and Second kind Chebyshev wavelet method [20]. Among these methods, the wavelet method is more attractive.

Within the last years, wavelet bases have been successfully used for the numerical solution of PDEs. The main characteristic of this method is that it reduces these problems to those of solving systems of algebraic equations, thus it greatly simplifies the problems.

It is worth pointing out that in this work, we propose an alternative approach. We shall apply collocation method to solve convection-diffusion equations numerically, in general case. In the current investigation, we reduce the problem to a set of algebraic equations by expanding the unknown function as Hermite scaling functions, with unknown coefficients. The operational matrices of derivative, integration and product are given. The idea of using operational matrices was used in the literature by several authors [5, 12-15]. These matrices together with the Hermite scaling functions are then utilized to evaluate the unknown coefficients. Our results have good agreement with exact results and demonstrate the viability of the new technique. In this sense,these methods have the potential to provide a wider applicability. On the other hand, the comparison of the results obtained by these methods and others shows that the new method provide more accurate solutions than those obtained by other authors.

This article is organized as follows: In Section 2, we describe the formulation of the Hermite scaling functions on $[0,\ell]$ and derive the operational matrices of derivative, integration and product required for our subsequent development. In Section 3, the proposed method are used to approximate the solution of the problem. As a result, a set of algebraic equations is formed and a solution of the considered problem is introduced. In Section 4, we report our computational results and demonstrate the accuracy of the proposed numerical scheme by presenting several numerical examples. Section 5, completes this article with a brief conclusion.

2. Cardinal Hermite interpolant multiscaling functions

The cardinal Hermite interpolant scaling functions $\phi = (\phi_1(x), \phi_2(x), \phi_3(x), \phi_4(x))^T$ is defined as [9]

$$\begin{aligned}
\phi_1(x) &= (x+1)^4(1-4x+10x^2-20x^3)\chi_{[-1,0]}(x) \\
&\quad +(x-1)^4(1+4x+10x^2+20x^3)\chi_{[0,1]}(x), \\
\phi_2(x) &= (x+1)^4(x-4x+10x^3)\chi_{[-1,0]}(t) \\
&\quad +(x-1)^4(x+4x+10x^3)\chi_{[0,1]}(x), \qquad (2.1)\\
\phi_3(x) &= (x+1)^4(x^2/2-2x^3)\chi_{[-1,0]}(x) + (x-1)^4(x^2/2+2x^3)\chi_{[0,1]}(x), \\
\phi_4(x) &= (x+1)^4x^3/6\chi_{[-1,0]}(x) + (x-1)^4x^3/6\chi_{[0,1]}(x),
\end{aligned}$$

where

$$\chi_{[x_0,x_1]}(x) = \begin{cases} 1, & x \in [x_0, x_1], \\ 0, & o.w. \end{cases}$$

It is known that ϕ is symmetric, supported on $[0,1]$, has accuracy of order 8 and belongs to $W^{4.5}$. Therefore [9,10], $\phi \in C^{4-\epsilon}$ for any $\epsilon > 0$. Moreover the vector ϕ satisfies the following properties

$$\begin{aligned}
&\phi(k) = \delta_k[1,0,0,0]^T, && \phi'(k) = \delta_k[0,1,0,0]^T, \\
&\phi''(k) = \delta_k[0,0,1,0]^T, && \phi'''(k) = \delta_k[0,0,0,1]^T, \qquad \forall k \in \mathbb{Z}. \qquad (2.2)
\end{aligned}$$

where $\delta_0 = 1$ and $\delta_k = 0$ for all $k \in \mathbb{Z}\backslash\{0\}$.
Suppose

$$\phi_i^{j,k}(x) = \phi_i(2^jx-k), \quad i = 1,2,3,4, \quad j,k \in \mathbb{Z},$$

and

$$ß_{i,j,k} = \text{supp}\left[\phi_i^{j,k}(x)\right] = \text{clos}\{x : \phi_i^{j,k}(x) \neq 0\}, \quad i = 1,2,3,4,$$

It is easy to see that

$$ß_{i,j,k} = \left[2^{-j}(k-1), 2^{-j}(k+1)\right], \quad j,k \in \mathbb{Z}.$$

Define the set of indices

$$S_{i,j} = \{k : ß_{i,j,k} \cap (0,\ell) \neq \emptyset\}, \quad j \in \mathbb{Z},$$

it is easy to see that $S_{i,j} = \left\{0, \ldots, \ell \times 2^j\right\}$, $j \in \mathbb{Z}$.

We need the biorthogonal Hermite functions intrinsically defined on $[0,\ell]$, so we put

$$\phi_i^{j,k}(x) = \phi_i^{j,k}(x)\chi_{[0,\ell]}(x), \qquad j \in \mathbb{Z}, \qquad k \in S_{i,j}, \quad i = 1,2,3,4.$$

2.1. Function Approximation

For a fixed $j = J$, a function $f(x)$ defined over $[0, \ell]$ may be approximated by biorthogonal multiscaling function as [5,12,13]

$$f(x) \approx \sum_{k=0}^{\ell\times 2^J} \left\{ c_1^{J,k}\phi_1^{J,k}(x) + c_2^{J,k}\phi_2^{J,k}(x) + c_3^{J,k}\phi_3^{J,k}(x) + c_4^{J,k}\phi_4^{J,k}(x) \right\} = C^T\Phi_J(x), \quad (2.3)$$

where Φ_J and C are $N \times 1$ vectors with $N = 4(\ell \times 2^J + 1)$ as

$$\Phi_J(x) = \Big[\phi_1^{J,0}(x), \phi_2^{J,0}(x), \phi_3^{J,0}(x), \phi_4^{J,0}(x)| \dots \\ \dots|\phi_1^{J,\ell\times 2^J}(x), \phi_2^{J,\ell\times 2^J}(x), \phi_3^{J,\ell\times 2^J}(x), \phi_4^{J,\ell\times 2^J}(x)\Big]^T \quad (2.4)$$

$$C = \Big[c_1^{J,0}, c_2^{J,0}, c_3^{J,0}, c_4^{J,0}|\dots|c_1^{J,\ell\times 2^J}, c_2^{J,\ell\times 2^J}, c_3^{J,\ell\times 2^J}, c_4^{J,\ell\times 2^J}\Big]^T \quad (2.5)$$

Because of the interpolatory nature of multiscaling function defined in (2.2), the coefficients $c_i^{J,k}$ are computed by

$$c_1^{J,k} = f(k/2^J), \quad c_2^{J,k} = 2^{-J}f'(k/2^J), \\ c_3^{J,k} = 2^{-2J}f''(k/2^J), \quad c_4^{J,k} = 2^{-3J}f'''(k/2^J), \quad k = 0, 1..., \ell \times 2^J.$$

The approximation of the two-dimensional function $u(x, t)$ for $0 \le x \le \ell$ and $0 \le t \le \mathsf{T}$ is [5,12]

$$u(x, t) \approx \Phi_J^T(t)U\Phi_J(x), \quad (2.6)$$

where U is a $M \times N$ block matrix with $M = 4(\mathsf{T} \times 2^J + 1)$ and is given by

$$U = \begin{bmatrix} U_0^0 & \cdots & U_{\mathsf{T}\times 2^J}^0 \\ \vdots & & \vdots \\ U_0^{\ell\times 2^J} & \cdots & U_{\mathsf{T}\times 2^J}^{\ell\times 2^J} \end{bmatrix}. \quad (2.7)$$

In Eq. (2.7), U_s^l, $s = 0, \dots, \mathsf{T} \times 2^J$, $l = 0, \dots, \ell \times 2^J$, are 4×4 matrices. Using Eq. (2.3) we have

$$U_s^l(i, j) = 2^{(i+j-2)} \frac{\partial^{j-1}x}{\partial x^{j-1}} \left(\frac{\partial^{i-1}u(x, t)}{\partial t^{i-1}} \big|_{t=s/2^J} \right) \big|_{x=1/2^J}, \qquad i, j = 1, \dots, 4. \quad (2.8)$$

3. Some formulas for cardinal Hermite functions

In this subsection, we give some formulas for our future computations. Using Eq. (2.3), we can approximate the function $\phi_i'(2^Jx - l)$ as

$$\phi_i'(2^Jx - l) = \sum_{k=l-1}^{k=l+1} \Big\{ (\phi_i'(k - l)\phi_1^{J,k}(x) + \phi_i''(k - l)\phi_2^{J,k}(x) \\ + \phi_i'''(k - l)\phi_3^{J,k}(x) + \phi_i''''(k - l)\phi_4^{J,k}(x) \Big\},$$

in which $i = 1, 2, 3, 4$ and $l = 0, \dots, \ell \times 2^J$.

Then, the differentiation of vector Φ_J in Eq. (2.4) can be expressed as [5,12,13,14]

$$\Phi'_J(x) = D_\phi \Phi_J(x). \tag{3.1}$$

where the D_ϕ is $N \times N$ matrix as defined as

$$D_\phi = 2^J \begin{bmatrix} A_0 & B_0 & & & & \\ C_0 & A_1 & B_1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & C_{\ell\times 2^J-2} & A_{\ell\times 2^J-1} & B_{\ell\times 2^J-1} \\ & & & & C_{\ell\times 2^J-1} & A_{\ell\times 2^J} \end{bmatrix}.$$

in which

$$A_k = \begin{bmatrix} 0 & 0 & 0 & -840 \\ 1 & 0 & 0 & -480\gamma_k \\ 0 & 1 & 0 & -120 \\ 0 & 0 & 1 & -16\omega_k \end{bmatrix}, \quad B_k = \begin{bmatrix} 0 & 0 & 0 & 840 \\ 0 & 0 & 0 & 360\lambda_k \\ 0 & 0 & 0 & 60 \\ 0 & 0 & 0 & 4\lambda_k \end{bmatrix},$$

$$C_k = \begin{bmatrix} 0 & 0 & 0 & 840 \\ 0 & 0 & 0 & -360\tau_k \\ 0 & 0 & 0 & 60 \\ 0 & 0 & 0 & -4\tau_k \end{bmatrix},$$

with

$$\begin{aligned} &\gamma_k := \operatorname{sgn}(\phi_2(2^J x - k)), \qquad \omega_k := \operatorname{sgn}(\phi_4(2^J x - k)), \\ &\lambda_k := \kappa_1(2^J x - k - 1), \qquad \tau_k := \kappa_2(2^J x - k), \qquad k = 1, \dots, \ell \times 2^J. \end{aligned}$$

where

$$\kappa_1(x) = \begin{cases} 1 & x \leq 0 \\ 0 & \text{otherwise} \end{cases},$$

$$\kappa_2(x) = \begin{cases} 1 & x \geq 0 \\ 0 & \text{otherwise} \end{cases},$$

and

$$\operatorname{sgn}(x) = \begin{cases} 1 & x \geq 0 \\ -1 & \text{otherwise} \end{cases},$$

Using a similar method, we can find the operational matrix of integration for multiscaling function. The integration of vector Φ_J, defined in (2.4), can be achieved as [5,12]

$$\int_0^x \Phi_J(s)ds \approx I_\Phi \Phi_J(x), \tag{3.2}$$

where I_Φ is $N \times N$ operational matrix of integration for multiscaling functions and can be obtained by the following process. The function $\int_0^x \phi_i(2^J s - l)ds$ using Eq. (2.3) can be approximated as

$$\int_0^x \phi_i(2^J s - l)ds \approx \sum_{k=0}^{\ell\times 2^J} (\{\int_0^{2^J} \phi_i(2^J x - l)dt\}\phi_1^{J,k}(x) + \frac{1}{2^J}\phi_i(k-l)\phi_2^{J,k}(x) + \frac{1}{2^{2J}}\phi_i'(k-l)\phi_3^{J,k}(x) + \frac{1}{2^{3J}}\phi_i''(k-l)\phi_4^{J,k}(x)),$$

for $i = 1,2,3,4$ and $l = 1,\dots,\ell\times 2^J$. Then, It can be shown that

$$I_\Phi = \frac{1}{2^J}\begin{bmatrix} R_1 & R_2 & R_2 & \cdots & \cdots & R_2 \\ & R_3 & R_4 & \cdots & \cdots & R_4 \\ & & R_3 & \ddots & \cdots & R_4 \\ & & & \ddots & \ddots & \vdots \\ & & & & R_3 & R_4 \\ & & & & & R_3 \end{bmatrix},$$

where

$$R_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; R_2 = \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 \\ \frac{3}{28} & 0 & 0 & 0 \\ \frac{1}{84} & 0 & 0 & 0 \\ \frac{1}{1680} & 0 & 0 & 0 \end{bmatrix},$$

$$R_3 = \begin{bmatrix} \frac{1}{2} & 1 & 0 & 0 \\ -\frac{3}{28} & 0 & 1 & 0 \\ \frac{1}{84} & 0 & 0 & 1 \\ -\frac{1}{1680} & 0 & 0 & 0 \end{bmatrix}, R_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{84} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

The following property of the product of two multiscaling function vectors [12] will also be used. Let

$$\Phi_J(x)\Phi_J^T(x)Z \approx \tilde{Z}\Phi_J(x), \tag{3.3}$$

where

$$Z = \left[z_1^{J,0}, z_2^{J,0}, z_3^{J,0}, z_4^{J,0}|\dots|z_1^{J,\ell\times 2^J}, z_2^{J,\ell\times 2^J}, z_3^{J,\ell\times 2^J}, z_4^{J,\ell\times 2^J}\right]^T$$

is an $N \times 1$ vector, and $\tilde{Z}$ is a $N \times N$ operational matrix of product given by

$$\tilde{Z} = \begin{bmatrix} \tilde{Z}_0 & & & \\ & \tilde{Z}_1 & & \\ & & \ddots & \\ & & & \tilde{Z}_{\ell\times 2^J} \end{bmatrix},$$

where $\tilde{Z}_k$ is 4×4 matrix given by

$$\tilde{Z}_k = \begin{bmatrix} z_1^{J,k} & z_2^{J,k} & z_3^{J,k} & z_4^{J,k} \\ & z_1^{J,k} & 2z_2^{J,k} & 3z_3^{J,k} \\ & & z_1^{J,k} & 3z_2^{J,k} \\ & & & z_1^{J,k} \end{bmatrix}, \qquad k = 0, \dots, \ell \times 2^J. \tag{3.4}$$

Theorem. Suppose that the function $f : [0, \ell] \to \mathbb{R}$ is eight times continuously differentiable, $f \in C^8[0, \times]$, and the interpolation operator P_J mapping function f into space V_J be as Eq. (2.3), then the error bound is given by

$$||f - P_J||_\infty = \max_{x\in[0,\ell]} |f(x) - P_J(x)| \leq \frac{2^{-8J-8}}{8!} ||f^{(8)}||_\infty.$$

where $P_J(x) = C^T \Phi_J(x)$.

Proof. See [3]. □

4. Description of proposed method

In this section, we solve the convection-diffusion equation in Eq. (1.1) with initial and boundary conditions in Eqs. (1.2) and (1.3). At the first we integrate of both side of (1.1) in the interval $[0, t]$ and using Eq. (1.2) we have

$$u(x,t) - f(x) + \alpha(x) \int_0^t u_x(x,s)ds = \beta(x) \int_0^t u_{xx}(x,s)ds + \int_0^t q(x,s)ds \tag{4.1}$$

Similarly to Eq. (2.6), we expand $f(x)$ and $q(x,t)$ as

$$f(x) \approx \Phi_J^T(t) F \Phi_J(x), \tag{4.2}$$
$$q(x,t) \approx \Phi_J^T(t) Q \Phi_J(x), \tag{4.3}$$

Now using (2.6), (3.1), (3.2) and (4.3) we can write

$$\int_0^t u_{xx}(x,s)ds = \left(\int_0^t \Phi_J^T(s)ds\right) U \left(\frac{d^2\Phi_J(x)}{dx^2}\right) \approx \Phi_J^T(t) I_\Phi^T U D_\phi^2 \Phi_J(x), \tag{4.4}$$
$$\int_0^t u_x(x,s)ds = \left(\int_0^t \Phi_J^T(s)ds\right) U \left(\frac{d\Phi_J(x)}{dx}\right) \approx \Phi_J^T(t) I_\Phi^T U D_\phi \Phi_J(x), \tag{4.5}$$
$$\int_0^t q(x,s)ds \approx \int_0^t \Phi_J(s) Q \Phi_J(x) ds \approx \Phi_J^T(t) I_\phi^T Q \Phi_J(x), \tag{4.6}$$

The functions $\alpha(x)$ and $\beta(x)$ can be approximated as

$$\alpha(x) \approx \Phi_J^T(x)\alpha, \qquad \beta(x) \approx \Phi_J^T(x)\beta. \tag{4.7}$$

Applying Eqs. (4.2), (4.4)- (4.7) in Eq. (4.1), we get

$$\begin{aligned}\Phi_J^T(t)U\Phi_J(x) - \Phi_J^T(t)F\Phi_J(x) + \Phi_J^T(t)I_\Phi^T UD_\phi\Phi_J(x)\Phi_J^T(x)\alpha \\ = \Phi_J^T(t)I_\Phi^T UD_\phi^2\Phi_J(x)\Phi_J^T(x)\beta + \Phi_J^T(t)I_\phi^T Q\Phi_J(x),\end{aligned} \tag{4.8}$$

By using operational product matrix, the above relation can be written

$$\begin{aligned}\Phi_J^T(t)U\Phi_J(x) - \Phi_J^T(t)F\Phi_J(x) + \Phi_J^T(t)I_\Phi^T UD_\phi\tilde{\alpha}\Phi_J(x) \\ = \Phi_J^T(t)I_\Phi^T UD_\phi^2\tilde{\beta}\Phi_J(x) + \Phi_J^T(t)I_\phi^T Q\Phi_J(x),\end{aligned} \tag{4.9}$$

where $\tilde{\alpha}$ and $\tilde{\beta}$ are $N \times N$ known matrices. Now by rearranging the Eq. (4.9) and collocating it in $N(N-2)$ points (x_j, t_i), $j = 1, \cdots, N-2$, $i = 1, \cdots, N$, where $x_j = \frac{j}{N-1}$, $j = 1, \cdots, N-2$ and $t_i = \frac{i}{N}$, $i = 1, \cdots, N$, we obtain

$$\Gamma(x_j, t_i) = \Phi_J^T(t_i)(U - F + I_\Phi^T UD_\phi\tilde{\alpha} - I_\Phi^T UD_\phi^2\tilde{\beta} - I_\phi^T Q)\Phi_J(x_j) = 0, \tag{4.10}$$

Using Eq. (2.6) in the boundary conditions (1.3) we get

$$\Phi_J^T(0)U\Phi_J(x) \approx V_1^T\Phi_J(x), \tag{4.11}$$

$$\Phi_J^T(\mathsf{T})U\Phi_J(x) \approx V_2^T\Phi_J(x), \tag{4.12}$$

where V_1 and V_2 are $N \times 1$ vectors and we know their entries. The entries of $\Phi_J(t)$ and $\Phi_J(x)$ are independent, so Eqs. (4.11) and (4.12) give

$$\Lambda_1 = \Phi_J^T(0)U - V_1^T = 0, \tag{4.13}$$

$$\Lambda_2 = \Phi_J^T(\mathsf{T})U - V_2^T = 0. \tag{4.14}$$

By choosing N equations of $\Lambda_1 = 0$ and $\Lambda_2 = 0$, we get $2N$ equations. These equations together with Eqs. (4.10) give a system of algebraic equations with N^2 unknowns and equations, which can be solved for $U_{i,j}$, i, $j = 1, \ldots, N$. So the unknown function $u(x,t)$ can be found using Eq. (2.6).

5. Numerical experiment

In this section, we will use the proposed method to solve the initial boundary value problem of convection-diffusion equation with variable or constant coefficients. The following numerical examples are given to show the effectiveness and practicality of the method and the results have been compared with the exact solution.

Example 1. Consider the convection-diffusion equation

$$\frac{\partial u}{\partial t} + \alpha(x)\frac{\partial u}{\partial x} = \beta(x)\frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1, \ 0 \le t \le 1,$$

with initial condition

$$u(x,0) = e^{-x},$$

and boundary conditions

$$u(0,t) = e^{-0.09t}, \quad u(1,t) = e^{-1-0.09t}, \quad 0 \le t \le 1.$$

where $\alpha(x) = -0.1$ and $\beta(x) = 0.01$. The exact solution of this example is $u(x,t) = e^{-x-0.09t}$.

The results for example 1 are displayed in Table I. For the purpose of comparison in Table I, we compare the absolute error of our method with different values of J with the method in [20]. The space-time diagram of the exact and approximate solutions are shown in Fig. 1. Comparison of the approximate solution with the exact solution for $t = 1$ is presented in Fig. 2.

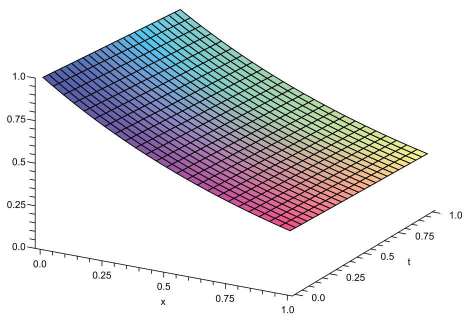

Figure 1: Approximate and exact solutions of example 1 with $J = 2$.

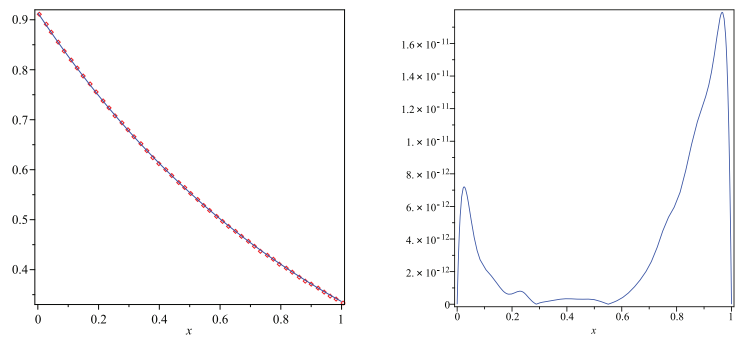

Figure 2: Comparison of the approximate solution with the exact solution (left) and absolute error (right) at $t = 1$ with $J = 2$ for $x \in [0, 1]$, example 1.

x_i	Method of [20]		
	$k=2,\ M=4$	$k=2,\ M=5$	$k=2,\ M=6$
0.1	2.54×10^{-7}	5.55×10^{-9}	4.65×10^{-12}
0.2	7.19×10^{-8}	4.28×10^{-9}	8.34×10^{-12}
0.3	6.48×10^{-8}	3.97×10^{-9}	1.36×10^{-11}
0.4	2.09×10^{-7}	4.54×10^{-9}	3.85×10^{-11}
0.5	9.31×10^{-7}	2.37×10^{-8}	7.53×10^{-10}
0.6	1.73×10^{-7}	3.47×10^{-9}	9.90×10^{-12}
0.7	3.90×10^{-8}	2.79×10^{-9}	5.54×10^{-12}
0.8	5.96×10^{-8}	2.41×10^{-9}	1.18×10^{-11}
0.9	2.92×10^{-7}	6.61×10^{-9}	7.69×10^{-11}
x_i	Present method		
	$J=0$	$J=1$	$J=2$
0.1	5.31×10^{-9}	1.21×10^{-10}	1.03×10^{-13}
0.2	1.79×10^{-8}	2.14×10^{-10}	2.11×10^{-13}
0.3	3.60×10^{-8}	1.69×10^{-10}	4.13×10^{-13}
0.4	5.09×10^{-8}	5.58×10^{-11}	4.36×10^{-13}
0.5	5.35×10^{-8}	3.04×10^{-11}	3.70×10^{-13}
0.6	4.20×10^{-8}	7.36×10^{-11}	3.85×10^{-13}
0.7	2.38×10^{-8}	1.31×10^{-10}	7.44×10^{-14}
0.8	8.91×10^{-9}	1.01×10^{-10}	1.00×10^{-13}
0.9	1.50×10^{-9}	1.27×10^{-11}	3.20×10^{-13}

Table I. Comparison between the approximate and exact solutions with $t = 0.3$ for Example 1

Example 2. Consider the convection-diffusion equation

$$\frac{\partial u}{\partial t} + \alpha(x)\frac{\partial u}{\partial x} = \beta(x)\frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1,\ 0 \le t \le 1,$$

with initial condition

$$u(x,0) = x^3, \quad 0 \le x \le 1,$$

and boundary conditions

$$u(0,t) = 0, \qquad u(1,t) = e^t, \quad 0 \le t \le 1,$$

where $\alpha(x) = -\frac{x}{6}$ and $\beta(x) = \frac{x^2}{12}$. The exact solution of the above example is $u(x,t) = x^3 e^t$.

The results for example 2 are displayed in Table II, Fig. 3 and Fig. 4. In Table II, the approximate solution obtained by the present method have been compared with the exact solution. The approximate solutions with exact solutions for different values of t are presented in Fig. 3. Also in Fig. 4, the absolute errors at various times of t have been reported.

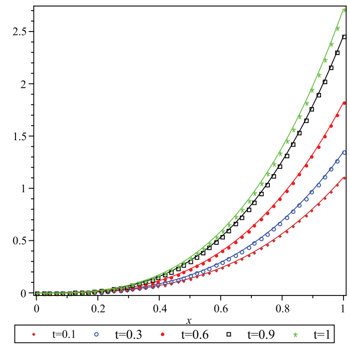

Figure 3: Comparison of the Approximate solution with the exact solution at various times with $J = 2$, example 2.

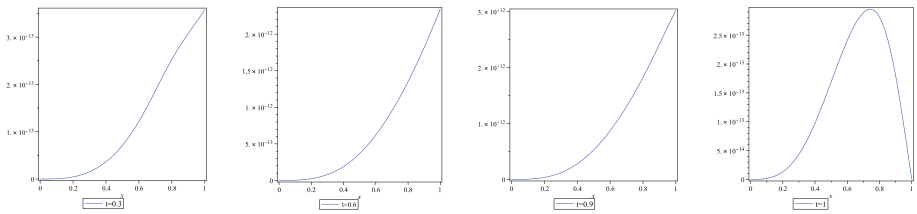

Figure 4: Absolute errors at various times of t, example 2.

Table II. Comparison between the approximate and exact solutions with $t = 0.3$ for example 2.

x_i	$J = 0$	$J = 1$	$J = 2$	Exact solution
0.1	0.0013498587153649	0.0013498588070820	0.0013498588075754	0.0013498588075760
0.2	0.0107988697594050	0.0107988704566576	0.0107988704606035	0.0107988704606080
0.3	0.0364461854312445	0.0364461877912192	0.0364461878045367	0.0364461878045521
0.4	0.0863909580582440	0.0863909636532605	0.0863909636848278	0.0863909636848642
0.5	0.1687323399533219	0.1687323508852751	0.1687323509469293	0.1687323509470004
0.6	0.2915694834559291	0.2915695023298699	0.2915695024362948	0.2915695024364167
0.7	0.4630015410427662	0.4630015708304760	0.4630015709983824	0.4630015709985691
0.8	0.6911276654943338	0.6911277092332921	0.6911277094786607	0.6911277094789136
0.9	0.9840470101034084	0.9840470703901410	0.9840470707225983	0.9840470707229063
1.0	1.3498587289115325	1.3498588071598232	1.3498588075756450	1.3498588075760031

Example 3. Consider the convection-diffusion equation

$$\frac{\partial u}{\partial t} + \alpha(x)\frac{\partial u}{\partial x} = \beta(x)\frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1,\ 0 \le t \le 1,$$

where the initial and boundary conditions are defined such that the exact solution is

$$u(x,t) = e^{t-x}.$$

We applied the new method presented in this paper with $J = 2$ and solve the above equation. The approximate solution with the exact solution have been displayed in Fig. 5. Fig. 6, shows the plot of error for $u(x,t)$ with $J = 2$.

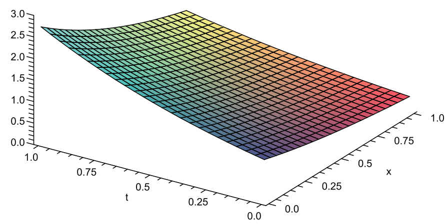

Figure 5: Approximate and exact solutions for $x \in [0,1]$ and $t \in [0,1]$ with $J = 2$, example 3.

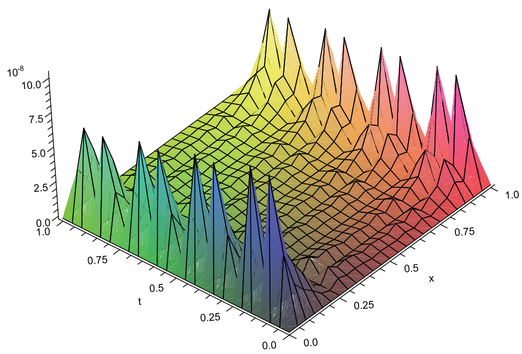

Figure 6: Absolute error with $J = 2$ for example 3.

Example 4. Consider the convection-diffusion equation

$$\frac{\partial u}{\partial t} + 0.1\frac{\partial u}{\partial x} = 0.02\frac{\partial^2 u}{\partial x^2}, \quad 0 \le x \le 1, \quad 0 \le t \le 1,$$

where the initial and boundary conditions are defined such that the exact solutionis

$$u(x,t) = e^{1.7712434446770x - 0.09t}.$$

The logarithm of the absolute error of the present method at the time level $t = 0.1$ have been displayed in Fig. 7. As the Fig. 7. shows the results of the new method is more effective than the method in [17] (see Fig. 1. in [17]). For the purpose of comparison, the absolute errors between exact and approximate solutions at various times are presented in Fig. 8, Also the approximate and exact solutions are plotted in Fig. 9.

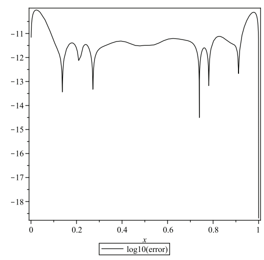

Figure 7: Accuracy of method with $J = 2$ for example 4.

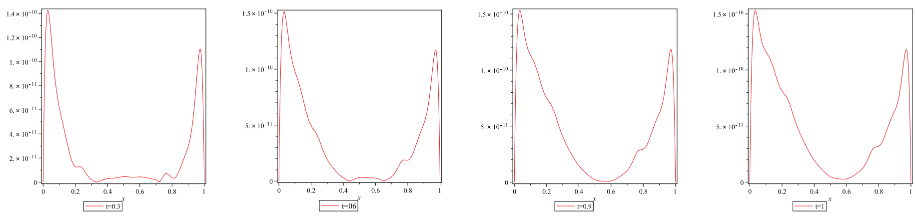

Figure 8: Absolute error between exact and approximate solutions at various times t with $J = 2$, example 4.

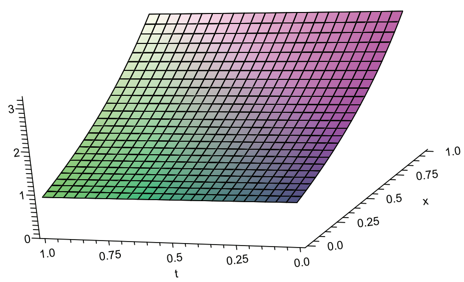

Figure 9: Approximate solution with exact solution for $J = 2$, example 4.

6. Conclusion

This paper focused on solving the Convection diffusion equations. Hermite interpolant biorthogonal multiscaling functions on $[0, 1]$ were employed and the operational matrices of derivative, integration and product for them are constructed. The obtained results showed that this approach can be solve the problem effectively.

References

1. H. H. Cao, L. B. Liu and Y. Zhang, S. M. Fu, A fourth-order method of the convection-diffusion equations with Neumann boundary conditions, Appl. Math. Comput. **217** (2011) 9133–9141.

2. X. Chen and J. Xiang, Solving diffusion equation using wavelet method, Appl. Math. Comput. **217** (2011) 6426–6432.

3. P. G. Ciarlet, M. H. Schultz and R. S. Varga, Numerical methods of highorder accuracy for nonlinear boundary value problems I. One dimensional problems, Numer. Math. **9** (1967) 294–430.

4. C. Clavero, J.C. Jorge and F. Lisbona, A uniformly convergent scheme on a nonuniform mesh for convection-diffusion parabolic problems, J. Comput. Appl. Math. **154** (2003) 415–429.

5. M. Dehghan and M. Lakestani, The Use of Cubic B-Spline Scaling Functions for Solving the One-dimensional Hyperbolic Equation with a Nonlocal Conservation Condition, Numerical Methods for Partial Differential Equations. **23** (2007) 1277–1289.

6. S. Dhawan, S. Kapoor and S. Kumar, Numerical method for advection diffusion equation using FEM and B-splines, J. Comput. Sci. **3** (2012) 429–437.

7. H. Ding and Y. Zhang, A new difference scheme with high accuracy and absolute stability for solving convection-diffusion equations, J. Comput. Appl. Math. **230** (2009) 600–606.

8. M. El-Gamel, A Wavelet-Galerkin method for a singularly perturbed convection-dominated diffusion equation, Appl. Math. Comput. **181** (2006) 1635–1644.

9. B. Han and Q. Jiang, Multiwavelets on the interval, Appl. Comput. Harmon. Anal. **12** (2002) 100–127.

10. Q. Jiang, On the regularity of matrix refinable functions, SIAM. J. Math. Anal. **29** (1998) 1157–1176.

11. S. Karaa and J. Zhang, High order ADI method for solving unsteady convection-diffusion problems, J. Comput. Phys. **198** (2004) 1–9.

12. M. Lakestani and M. Dehghan, A new technique for solution of a parabolic inverse problem, Kybernets, **37** (2008) 352–364.

13. M. Lakestani and M. Dehghan, The solution of a second-order nonlinear differential equation with Neumann boundary conditions using semiorthogonal B-spline wavelets. International J. Comput. Math. **83** (2006) 685–694.

14. M. Lakestani and M. Dehghan, Collocation and finite difference-collocation methods for the solution of nonlinear Klein-Gordon equation, Computer Physics Communications. **181** (2010) 1392–1401.

15. M. Lakestani and M. Dehghan, The use of Chebyshev cardinal functions for the solution of apartial differential equation with an unknown time-dependent coefficient subject to an extra measurement, Comput. Appl. Math. **235** (2010) 669–678.

16. J. I. Ramos, A piecewise-analytical method for singularly perturbed parabolic problems, Appl. Math. Comput. **161** (2005) 501–512.

17. D. K. Salkuyeh, On the finite difference approximation to the convection-diffusion equation, Appl. Math. Comput. **179** (2006) 79–86.

18. W. Q. Wang, The alternating segment Crank-Nicolson method for solving convection-diffusion equation with variable coefficient, Appl. Math. Mech. **24** (2003) 32–42.

19. S. Yüzbasi and N. Sahin, Numerical solutions of singularly perturbed one-dimensional parabolic convection-diffusion problems by the bessel collocation method, Appl. Math. Comput. **220** (2013) 305–315.

20. F. Zhou and X. Xu, Numerical solution of the convection diffusion equations by the second kind Chebyshev wavelets, Appl. Math. Comput. **247** (2014) 353–367.

E. Ashpazzadeh, Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

and

M. Lakestani, Department of Applied Mathematics, Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran
E-mail address: lakestani@tabrizu.ac.ir